DOI:10.3819/CCBR.2019.140011 Volume 15, 2022

COMPARATIVE COGNITION
& BEHAVIOR REVIEWS

PROBABILITY LEARNING BY PERCEPTRONS AND PEOPLE

Michael R.W. Dawson
University of Alberta, Department of Psychology

ISSN: 1911-4745

Contents

Preface

In 2008, *Comparative Cognition & Behavior Reviews* was kind enough to publish a monograph that explored the relationship between simple artificial neural networks called perceptrons and models of associative learning (Dawson, 2008). That work attempted to use perceptrons as a medium in which associative learning could be examined from the perspective of cognitive science. To do so, it presented a number of formal analyses as well as the results of a number of computer simulations of associative learning. It made the interesting discovery that two systems (i.e., perceptrons and the Rescorla-Wagner model) could be formally equivalent and at the same time produce different behavioral results. This "perceptron paradox" was dealt with by arguing that the equivalence between the two systems was at what cognitive scientists call the computational level of analysis, but differences between formally equivalent systems could still exist when their formal theories were brought to life at a different level, the algorithmic level of analysis.

One consequence of that monograph was my involvement in a research project with Marcia Spetch, Debbie Kelly, and my student Brian Dupuis that attempted to use perceptrons to model the behavior of biological agents in the reorientation task (Dawson, Kelly, Spetch, & Dupuis, 2008, 2010; Dupuis & Dawson, 2013a, 2013b). During this work, I (too slowly) realized that what the perceptrons were really doing was learning about the probability of reward associated with signals carried by cues. This led to some early explorations of the behavior of perceptrons in simple contingency experiments (Dawson & Dupuis, 2012; Dawson, Dupuis, Spetch, & Kelly, 2009). Eventually I started to explore the behavior of perceptrons when they learned about uncertain environments—environments in which an input stimulus does not signal an outcome with certainty but only signals the outcome with a certain degree of probability. The current monograph describes the results of this exploration.

The current monograph is a sequel to Dawson (2008). It presents computational and algorithmic treatments of how perceptrons adapt to uncertainty. It reports formal analyses that relate perceptron structures to Bayesian probability and logistic regression. It describes the results of experiments that investigate what perceptrons learn when there is not a one-to-one relationship between stimuli and responses. It also details the results of a study that explores human probability learning in a variety of conditions and relates human performance to that of perceptrons. All of these results suggest that both perceptrons and people behave as if they are naive Bayesians, at least in the basic kind of task studied here.

The current monograph also serves as a case study in synthetic psychology, an approach that involves building simple systems and then studying their behavior in a variety of interesting environments. I have long viewed artificial neural networks as a medium in which this synthetic approach can be pursued (Dawson, 2004). The various approaches described in the chapters that follow are attempts to demonstrate the utility of the synthetic approach for the study of probability learning.

Finally, the current monograph relates a specific topic (how associative systems adapt to uncertainty) to a variety of other literature related in one way or another to the evolution of cognitivism in psychology. These include cybernetics, information theory, probability theory, systems theory, statistical inference, decision theory, and the cognitive psychology of category learning. Some of the core ideas in these theories appear repeatedly as one studies the cognitive science of associative learning. I hope that the current monograph illustrates the rich interrelationships between the psychology of associative learning and these other fields.

Chapter 1: Uncertainty and Adaptation

Chapter 1 introduces the main topics of this monograph and the basic components that are used to explore these topics. Our relationship with the world is uncertain: We can never predict the future with perfect accuracy. We will see that this is due to three sources of uncertainty: epistemic, somatic, and ontic. However, we thrive in this uncertain environment, which suggests that we somehow can adapt to it. How do we adapt to an uncertain world? This chapter describes a probabilistic discrimination task, called the card-choice task, which focuses this question by exploring a smaller, but interesting, component of probability learning. This task is straightforward but can be related to diverse topics, including experiments on animal reorientation and studies of multiarmed bandit strategies. The card-choice task is also closely related to an old and well-established literature on probability learning. One of the core findings in this psychological literature is that humans exhibit probability matching: The number of times an action is performed is correlated with the probability that the action is rewarded. We hypothesize that probability matching is the result of associative learning of the sort described by a simple artificial neural network, the perceptron. We describe the basic properties of this type of network and train it on a probability discrimination task. The results of this study indicate that perceptrons are, at face value, plausible models of probability learning. A more detailed examination of probability estimation by perceptrons is required and is provided in later chapters.

1.1 Sources of Uncertainty

1.1.1 The End of Time's Arrow

The scientific revolution that began in the 16th century developed the metaphor of the clockwork universe (Shapin, 1996). According to this metaphor, the universe is a perfect machine, a clock wound by God; the laws of physics govern the movements of its gears. The seeds of this metaphor were planted in astronomy's attempts to improve its predictions about the future locations of heavenly bodies. Johannes Kepler revolutionized astronomy and its predictive power with his insight that the orbits of the planets are elliptical, not circular. Kepler's three laws of planetary motion may be the greatest scientific contribution ever (Ekeland, 1988).

The idea of the clockwork universe was firmly established by the mathematical treatment of Kepler's theory by another giant of the scientific revolution, Sir Isaac Newton. Newton's theory of gravity, formalized in the calculus that he invented, could be used to derive Kepler's laws. Newton's mathematical treatment of gravity provided the foundation of celestial mechanics, which attracted the attention of the greatest mathematical minds; the successes of celestial mechanics became the envy of all of the other sciences (Ekeland, 1988).

One intriguing property of celestial mechanics is that it eliminates the unidirectional arrow of time. That is, differential equations themselves pay no heed to the direction of time. If one knows the current positions and velocities of celestial bodies of interest, then one can use the differential equations of celestial mechanics to predict their positions at any desired time in the future or in the past. "[The human mind's] discoveries in mechanics and geometry, added to that of universal gravity, have enabled it to comprehend in the same analytical expressions the past and future states of the system of the world" (Laplace, 1796/1952, p. 4).

Celestial mechanics' ability to predict both the infinite future and the infinite past demonstrates the certainty provided by the clockwork universe. Given the current states of its gears, and understanding the differential equations that govern their turning, the clockwork model offered complete, deterministic, and certain predictions.

1.1.2 Epistemic Uncertainty

It is important to note that the clockwork universe offers predictive certainty only in principle. In practice, there are limits to its predictive power. For instance, consider Newton's derivation of Kepler's elliptical orbits. This derivation requires considering the gravitational forces between only two celestial bodies (e.g., the sun and the Earth). However, the result is in fact only a highly accurate approximation of reality, because other celestial bodies—the Earth's moon, the other planets, their moons, and so on—also contribute forces that influence the positions of the sun and the Earth. Perfect and complete prediction requires considering the effects of these other bodies on the particular orbits of interest.

Unfortunately, the clockwork model of the universe is compromised as soon as one uses celestial mechanics to predict the positions of three bodies instead of two (e.g., the sun, the Earth and the Earth's moon). By the end of the 19th century, Heinrich Bruns and Henri Poincaré proved that this situation, known as the three-body problem, has no general mathematical solution. Making predictions involving more than three bodies is similarly intractable.

The difference between the in principle and the in practice predictive power of the clockwork universe illustrates the limits of human abilities to understand and predict the world (Laplace, 1796/1952). In 1796, French mathematician Pierre Simon, Marquis de

Laplace, considered the properties of an agent, now known as Laplace's demon, capable of completely understanding and predicting the world:

> Given for one instant an intelligence which could comprehend all the forces by which nature is animated and the respective situation of the beings who compose it – an intelligence sufficiently vast to submit these data to analysis – it would embrace in the same formula the movements of the greatest bodies of the universe and those of the lightest atom; for it, nothing would be uncertain and the future, as the past, would be present to its eyes. (Laplace, 1796/1952, p. 4)

The predictive power of Laplace's demon arises from its infinite knowledge as well as from its unlimited power to process this information. Limitations to knowledge or to analytic power must in turn reduce the ability to predict the future (or the past) from the present. Laplace recognized that the human mind was subject to such limitations. As a result, the human mind is "feeble" in comparison to the demon's vast intelligence, "from which it will always remain infinitely removed" (Laplace, 1796/1952, p. 4). The practical implication of this is that the human mind is not capable of making perfect predictions about the future using current information. Human predictions can lead to surprises (e.g., when they prove to be incorrect); at best they are only probabilistic. The inability to perfectly predict the future due to limited knowledge of the world is called *epistemic uncertainty* (Gigerenzer & Murray, 1987).

Epistemic uncertainty exists when perfect predictions cannot be made from the current situation because of the lack of knowledge. Such uncertainty might simply occur because an agent is ignorant about how to use current evidence to make accurate predictions. In this case, acquiring knowledge about what evidence to use, or how to use it, can mediate or eliminate epistemic uncertainty. In other words, epistemic uncertainty can be reduced via learning.

1.1.3 Somatic Uncertainty

However, although epistemic uncertainty can be reduced, another kind of uncertainty is more difficult to counteract. For example, limitations in an agent's perceptual mechanisms may prevent it from sensing the appropriate information to be used to make predictions. Similarly, an agent's brain mechanisms may impose limitations that prevent the proper processing of information. Although epistemic uncertainty occurs when there is simply limited knowledge of the world, this second type of uncertainty occurs because limits in knowledge are due to constraints inherent in the bodily structure of the agent. Let us call this *somatic uncertainty*.

Somatic uncertainty has played an important role in the study of visual perception. It has long been known that the relationship between the distal visual world and our interpretation of it is ambiguous. This ambiguity is due to somatic limitations of perceptual mechanisms (Marr, 1982). For instance, the visual world (called the *distal stimulus*) can be considered to be a three-dimensional array of light sources and surfaces that reflect light. Light from this array stimulates retinal receptors in the eye, producing a pattern of activity called the *proximal stimulus*. However, when the distal stimulus produces the proximal stimulus, information is necessarily lost because of the somatic properties of the retina.

There are two reasons for this loss of information. First, the proximal stimulus is only two-dimensional, meaning that a whole dimension of the distal stimulus is lost when it is projected as a proximal stimulus. This is because the retina itself is two-dimensional. Second, certain properties of receptors (such as their limited field of view) also produce information loss. For example, some components of the movement of an edge will not be detected when motion detectors have small fields of view, leading to what is known as the aperture problem (Hildreth, 1983). Both of these reasons cause the proximal stimulus to underdetermine its distal cause: In principle, a single proximal stimulus is consistent with an infinite number of different distal stimuli (Marr, 1982). The proximal stimulus does not uniquely specify the distal array that caused it.

The somatic uncertainty of visual perception leads to theories about how the missing information can be replaced. These theories range from using unconscious inference (Gregory, 1970, 1978; Helmholtz & Southall, 1856/1962; Rock, 1983) to applying wired-in natural constraints (Dawson, 1991; Grimson, 1981; Hildreth, 1983; Marr, 1982; Richards, 1988; Ullman, 1979). These various theories are all motivated by the problem that visual experience is subject to somatic uncertainty.

Somatic uncertainty is also central to Egon Brunswik's *probabilistic functionalism* (Brunswik, 1943, 1952, 1955). This is illustrated in Brunswik's lens model, a version of which is provided in Figure 1-1. The lens model describes a relatively stable connection between a class of events in the world and a corresponding achievement, which might be an action on the world or a judgment about it. The connection between these two is provided by the functional arc, the dashed line in the figure.

In the lens model, the stability of the functional arc must be achieved by passing the effects of the world through the agent, who evaluates these filtered effects in order to render a judgment. The agent is the lens that maps

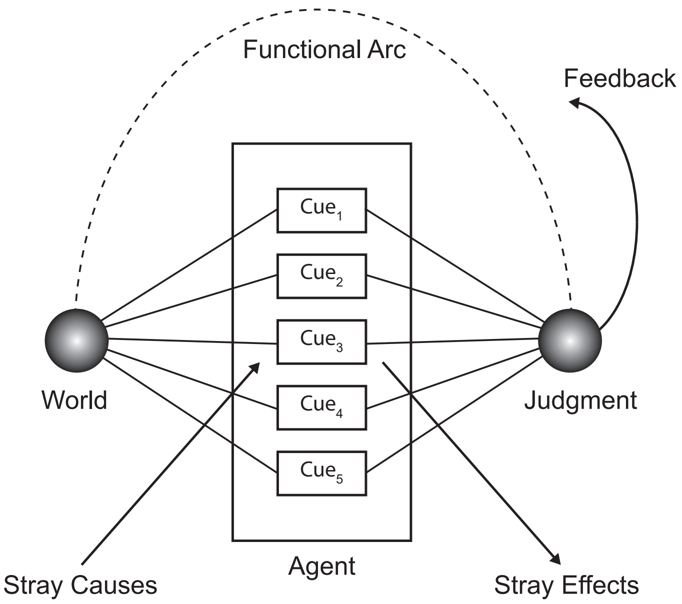


Figure 1-1. Brunswik's lens model of the mapping between the world and actions upon it. This figure is a variation of Brunswik (1952) Figure 1.

properties of the world onto judgments or actions. This mapping is mediated through a set of internal cues that represent perceived properties of the world. Important to note, the association between the world and the cues it produces is not perfect; Brunswik viewed the relationship between the world and a cue as being a correlation. This is consistent with the problem of visual underdetermination as just discussed. In addition, cues may be provided or modified by additional stray causes. A similarly imperfect or correlational mapping exists between the cues that are represented within the agent and the judgment or action that is made using these cues. Formalizations of the lens model use regression equations to describe the relationships between the cues and the world, and between the cues and the judgment (Hursch, Hammond, & Hursch, 1964; Tucker, 1964). Because neither of these relationships is perfect (i.e., because both of these relationships are correlational), "gross organismic coming-to-terms with the environment can thus never become foolproof" (Brunswik, 1952, p. 23).

The lens model is of interest for two reasons. First, it offers a context in which the differing influences of somatic and epistemic uncertainty can be considered. For instance, the problem of underdetermination for visual perception suggests that in the lens model the uncertain relationship between the world and the cues reflects somatic uncertainty. That is, our perceptual mechanisms may lead to a degree of uncertainty about which cues are present in the environment. If such uncertainty is somatic, then it cannot be reduced by an organism.

In contrast, the uncertain relationship between the cues and the judgment made by the agent reflects epistemic uncertainty. In this situation, if the agent's cues do not lead to a certain judgment, then this is likely because the agent does not properly understand how to use the cues to inform the judgment.

The second reason for interest in the lens model is related to this point about epistemic uncertainty. The purpose of the lens model is to illustrate a stable relationship between the world and an agent's judgments about it. This relationship is mediated by the lens of the agent, but it can also be changed. This is why Brunswik includes feedback in the lens model: As the agent experiences instances of an event class in the world, the agent can learn to improve its judgment of this class. Presumably, this can be accomplished only by reducing epistemic uncertainty—that is, by altering the mapping from cues to judgment. This is because somatic uncertainty (reflected in the mapping from the world to the cues) is related to limitations imposed by the agent's body and is therefore much more difficult to change.

As we proceed through this monograph, we will return to the lens model. One purpose of this monograph is to provide an account of mechanisms that might reduce epistemic uncertainty in a fashion that is consistent with the role of feedback in Figure 1-1. We use the perceptron as a model of how epistemic uncertainty can be reduced via associative learning.

Important to note, although perceptron learning can reduce epistemic uncertainty, perceptrons are still subject to somatic uncertainty. In particular, because perceptrons are simple—because they do not include hidden units—they are limited in terms of the kind of relationships that they can detect (Minsky & Papert, 1969). In later chapters, we show that in learning probabilistic relationships, this limitation manifests itself in the inability to detect probability of reward that is signaled by interactions between cues. This means that perceptrons cannot learn probabilities of reward perfectly in all situations. As the perceptron's simple structure is the source of this limitation, it cannot be overcome, and it reflects somatic uncertainty. We will see that with training, the perceptron does its best in these circumstances (by adjusting network parameters to reduce its epistemic uncertainty), but the somatic uncertainty cannot be eliminated. Of greater interest is that we will see that human participants exhibit a similar pattern of behavior and are much poorer learners of reward probability when it is signaled by cure interactions. This suggests that the somatic uncertainty of the perceptron, due to its simple structure, can provide a plausible model of human probability learning.

1.1.4 Ontic Uncertainty

The clockwork universe reflects a deterministic view of nature. The successes of Newtonian physics strengthened the assumption of determinism. Even the advent of quantum physics, which includes stochastic concepts, did not diminish determinism. This is because the wave function of quantum mechanics is deterministic in nature (Prigogine, 1997).

As we saw earlier, one property of the deterministic equations in physics is that they eliminate time's arrow. That is, these equations are reversible: One can use them to predict the state of the future or to predict the state of the past. To physics, the universe is time-symmetric; there is no distinction between past, present, and future (Prigogine, 1997). However, the time-symmetric view of physics seems problematic, because our experience indicates that worldly events cannot be reversed. "Yet everywhere—in chemistry, geology, cosmology, biology, and the human sciences—past and future play different roles" (Prigogine, 1997, p. 2). This leads to what Prigogine called the time paradox: If physics takes the world to be time-symmetric, then how can the arrow of time observed in all of the other sciences emerge from physics?

Prigogine's (1997) solution to the time paradox is to extend physics by including using probabilities as the basic objects of physics instead of trajectories or wave functions. This produces physical theories that are time-asymmetric. This also leads to a physics of nonequilibrium processes that describes self-organizing systems. These systems violate traditional physical ideas because their ability to self-organize reflects negative entropy, or a violation of the second law of thermodynamics. This second law states that the entropy of a system can never decrease; over time, a system must become less organized.

Ideas similar to Prigogine's have also developed in other fields of science. For example, beginning in the 1920s, Ludwig von Bertalanffy began his reaction against mechanistic and deterministic views in biology. Bertalanffy argued that biological systems were not closed systems of the sort that were governed by the second law of thermodynamics. Instead, Bertalanffy argued that biological systems were open; because of this openness, they could achieve a state of dynamic equilibrium that is a characteristic of self-organization (Bertalanffy, 1933, 1952). Bertalanffy argued that biology should discover the laws of biological organization; physicist Erwin Schrödinger's (1945) publication *What Is Life?* popularized this type of position. Later, Bertalanffy extended his view by arguing that similar principles of organization can be discovered across diverse scientific disciplines. This resulted in Bertalanffy pioneering what is now known as systems theory (Bertalanffy, 1968a). In 1954 Bertalanffy and three other Fellows at Stanford University's Center for Advanced Study in the Behavioral Sciences (economist Kenneth Boulding, mathematician Anatol Rapoport, and neurophysiologist Ralph Gerard) famously laid the foundations for a society devoted to the promotion of general systems theory (Hammond, 2003).

The seeds of Bertalanffy's original general system theory have grown into a forest of advances in multiple disciplines, including biology, ecology, neuroscience, psychology, and economics (Arthur, 2015; Bertalanffy, 1968b; Holland, 1992, 1995, 2012; Kauffman, 1995; Meadows & Wright, 2008; Ramage & Shipp, 2009). Many of these advances have been due to the use of new kinds of computer simulations, including genetic algorithms (Mitchell, 1996), artificial life (Langton, 1995), and cellular automata (Farmer, Toffoli, & Wolfram, 1984; Toffoli & Margolus, 1987). The common thread uniting these methods involves local, nonlinear interactions between neighboring elements; these interactions propagate through the system over time. These systems are compelling because complex—and difficult to predict—global regularities emerge from these local interactions (Holland, 1998).

The regularities that emerge in such systems are difficult to predict analytically; they are instead typically studied by building working models (Ashby, 1960). One reason that these systems are very difficult to predict is because they are frequently instances of what are known as chaotic or complex systems (Devaney, 1989; Gleick, 1987; Waldrop, 1992). In a chaotic system, very tiny differences in initial starting states can lead to dramatic qualitative differences in system behavior over time. As well, a dynamic chaotic system may not settle into a distinct stable state but instead will move through a regular, though unpredictable, sequence of states called a *strange attractor*. The complexity of the behavior of these systems often leads to their being described using fractal mathematics (Mandelbrot, 1983).

Chaotic systems are of interest because they are intrinsically impossible to predict. This impossibility is not due to epistemic or somatic uncertainty, which are properties of an agent. Instead, this impossibility is a fundamental property of the system being predicted itself. A world that by its very nature is impossible to predict exhibits *ontic uncertainty* (Gigerenzer & Murray, 1987). Note that the difference between ontic uncertainty and the other two forms of uncertainty (epistemic and somatic) that we have considered is that ontic uncertainty is a property of the world, not of the agent. Epistemic uncertainty reflects an agent's lack of knowledge of how to use cues, which can be reduced via learning. Somatic uncertainty reflects an agent's physical inability to detect cues that could be used to predict the world and cannot be overcome with learning. However, an agent with perfect knowledge (zero epistemic uncertainty) and gifted with the ability to perfectly perceive any cue (zero somatic uncertainty) would still be unable to predict the world if the world itself is intrinsically uncertain (nonzero ontic uncertainty).

There is a growing sense in systems theory that natural systems exhibit ontic uncertainty. However, ontic uncertainty can also be true in very mundane situations. Consider fair games of chance such as the Monty Hall problem, where one must choose which of three identical doors hides a prize, or a game of poker in which one can see only the backs of the cards held by opponents. The fairness of these games depends on the fact that, in principle, the available evidence does not permit the player to know which door hides the prize or whether his opponent holds the ace of spades.

1.1.5 Summary and Implications

The clockwork universe envisioned by the natural philosophers was completely deterministic and perfectly predictable. Humans, and other biological agents, do not exist in such an ideal universe. Limits to our knowledge about the world produce epistemic uncertainty. Structural constraints imposed by our sensory and neural systems restrict the kind of information that we can sense or process, causing somatic uncertainty. The growing belief that the world is intrinsically dynamic, nonlinear, chaotic, and unpredictable means that we constantly face ontic uncertainty.

This implies that biological agents exist in an uncertain, unpredictable world. Given our current knowledge of the world, we can never predict what will happen next with complete certainty. Nevertheless, we are able to survive or thrive; we can cope with this uncertainty. This suggests that we are at least able to make predictions that have a high probability of being correct, even though their accuracy can never be certain. How do we adapt to the probabilistic nature of our world?

One approach to explaining how biological agents deal with the world's uncertainty involves developing new formalisms about unpredictability. Mathematical probability theory, pioneered during the 18th century, is the prototypical example of this approach. It has a long history of development not only within mathematics itself but also in the context of a diversity of disciplines (David, 1962; Gigerenzer, 1989; Hacking, 2006). Probability theory also forms the basis for related formalisms that are intended in one way or another to cope with the world's uncertainty. These include game theory (Neumann & Morgenstern, 1944), decision theory (Bock & Jones, 1968; Chernoff & Moses, 1959; Eells, 1982; Luce, 1959), information theory (Cherry, 1957; Shannon & Weaver, 1949), statistical inference (Cox, 1961; Pólya, 1954; Savage, 1962), signal detection theory (Green & Swets, 1974), and cybernetics (Ashby, 1956; Wiener, 1948, 1950). Probabilistic techniques can be used as tools to aid decision-making in uncertain conditions (Heyck, 2012). After such tools have been developed and established, they often evolve into psychological theories concerning decisions under uncertainty (Gigerenzer & Murray, 1987).

Another approach to explaining how humans cope with uncertainty is to develop psychological theories of information processing by using experimental findings. Cognitive psychology has developed a number of theories in this fashion, including bounded rationality (Simon, 1982; Simon, Egidi, & Marris, 1995), heuristics and biases (Tversky & Kahneman, 1974), fast and frugal heuristics (Gigerenzer, 2000, 2010), and Bayesian rationality (Oaksford & Chater, 1998, 2007). Clearly, accounts of how to deal with uncertainty have a very long history; exploit diverse methodologies that include formal proofs, computer simulations, and psychological experiments; take a variety of theoretical forms; and appear in a multitude of disciplines.

The goal of this monograph is consistent with this state of affairs. It provides the perspective of a particular field, connectionist cognitive science, on judgments under uncertainty. Because this perspective is rooted in cognitive science, which is itself highly interdisciplinary (Dawson, 1998, 2013), it too will make contributions that include formal analyses, computer simulations, and psychological experiments. However, it attempts to use these various approaches to shed insight into judgments under uncertainty by focusing on a specific (though important) task and by modeling this task with a very simple system (an artificial neural network called a perceptron) the properties of which can be directly translated into formal notions of probability. In addition, perceptron learning can be formally related to psychological theories of associative learning (Dawson, 2008). It is hoped that this approach—the detailed study of a basic mechanism that is capable of coping with the probabilistic nature of its environment—will provide a concrete example that can be used as a foundation for exploring the large and varied literature related to this topic and for relating this literature to basic characteristics of associative learning.

Chapter 1 introduces the basic components of this investigation. First, it describes the task of interest: using signals or cues as information to make a choice, where this choice will lead to a reward, but only with a certain probability. Second, it explores the aspects of this task that can be improved with experience. In some sense, this investigates the kinds of changes that feedback might make in the lens model of Figure 1-1. Third, it describes the basic properties of a simple neural network, the perceptron. Simulation results show that this network is well suited to adapt itself in such a way that its performance on the task of interest can improve. After establishing these basic elements in Chapter 1, we will be in a position to consider a more detailed account that relates the properties of this task and of this artificial mechanism to probability discrimination and associative learning.

1.2 Seeking Rewards in an Uncertain World

This section of Chapter 1 introduces the first main element in our exploration of adapting to uncertainty: a basic task. We require a task that is straightforward so that it can be easily used to train a simple artificial neural network and so that it can be used to collect observations from human participants. However, in spite of its simplicity, it must be broad enough in scope to be able to provide insight into a variety of topics. Furthermore, this task should be relevant to everyday problems that arise in an uncertain world.

We develop an account of this task as follows: First, we provide a real-world example of a problem that illustrates its core properties. Second, we provide a general description of what these properties are. Third, we present several examples of different kinds of paradigms that have been explored in a variety of domains and argue that each is a variation of the task that is a basic element of the current monograph.

1.2.1 Choosing Cherries

Consider a commonplace activity: buying a flat of cherries. The goal is to purchase some cherries that are rewarding because of their sweet taste. However, in most cases we cannot directly test their sweetness by tasting them. Instead, we have to consider a variety of other cues or signals and use this evidence to inform our choice of a particular flat of cherries from various possibilities. Many cues could predict cherry sweetness.

For instance, confronted with a store's display of many flats of cherries, the various flats could be compared in terms of differences in cherry sizes, or differences in cherry colors. It might be expected that a flat containing large deep red cherries will be more rewarding than a flat containing smaller lighter-colored cherries. Similarly, it may be likely that a flat with cherries that are bruised, misshapen, or apparently spoiled in any other way will be less rewarding than a flat that does not possess these characteristics.

Other evidence might come into play as well. A store might have cherries labeled in terms of their place of origin. Perhaps cherries from British Columbia will be sweeter than cherries from Chile, or from Mexico, or from California. Perhaps this depends on the time of year that the shopping occurs. Chilean cherries might be the most rewarding if I am shopping in Edmonton in January. However, if I am shopping in the summer, then I might expect Canadian cherries to be sweeter. In short, the time of year can also provide information relevant to choosing cherries. The location being shopped might itself be a source of evidence. Perhaps cherries purchased at a fruit stand or at a farmers' market will be sweeter than cherries purchased at a Superstore, Walmart, or Costco. In addition, the price of cherries possibly indicates their sweetness. It is possible that a vendor recognizes that sweeter cherries are more valuable, and therefore offers them at a higher price. The vendor might also provide information about when the cherries were picked, or the flats might be labeled with a "best before" date.

Clearly, many kinds of evidence might inform a choice of cherries. Assume that some or all of this evidence is used to select a particular flat to purchase. This choice results from the evidence-based *prediction* that the cherries in this flat will be sweet. The actual taste of the chosen cherries then provides a *test* of this prediction. The results of this test can motivate changes in how evidence is used to predict sweetness. If the cherries are as sweet as predicted, then perhaps no changes will be made at all. If the prediction is less accurate, then there may be a stronger motivation to alter how evidence is used to choose cherries. It is plausible to assume that larger errors in predictions could lead to larger changes in the use of evidence.

Pleasant surprises may also cause changes the use of evidence. For instance, perhaps circumstances force a purchaser to choose one of the few remaining flats with cherries that are more yellow than red. This atypical color might predict that the cherries will be sour. A pleasant surprise—that these different cherries are also sweet, because they are a different species—could alter how color is used as evidence of cherry quality.

This commonplace task of choosing cherries has some key attributes. First, varieties of cues are used to inform an action: choosing one flat of cherries from the many available. This choice is made under the assumption that the cues predict that the selected cherries will be sweet. This prediction is not certain; it is tested later by tasting the selected fruit. The results of this test can then motivate changes in how the different cues are used to select cherries.

This commonplace task also raises a number of interesting questions. How and when is the use of evidence modified? If more than one source of evidence has been used, how is it decided which sources need to be reevaluated, and by how much? How are cues converted into a particular choice in the first place? A central goal of this monograph is to explore associative processes that provide answers to such questions.

1.2.2 A Generic Task

The previous section provided a concrete example of the kind of task that is of particular interest in the current monograph. Let us now provide a generic description of this task before reviewing some notable versions of it that have been reported in the literature.

In the generic task, the world consists of a finite set of different objects. Each object is characterized by a finite set of characteristics, which we call *cues*. So, some object x (represented as O_x) can be characterized as a set of cues, where each cue takes on some value: $O_x = (Cue_1, Cue_2, \ldots, Cue_n)$. The task itself is to choose one of the objects with the goal of being rewarded. It is assumed that each object, after being selected, will have a certain probability of providing a reward. It is also assumed that each cue provides a signal that carries information about the probability of reward. For instance, each cue could independently signal reward probability, and the actual probability of reward is the result of combining these signals. It is also possible that some cues interact, so multiple cues must be considered together to determine the likelihood of reward.

In general, it is assumed that the task proceeds as follows: An agent considers an object from the set of those that are available. Knowledge about this object—the signals provided by the object's cues—is then used to predict reward probability. This predicted probability is then used to decide whether to choose the object. When selected, the object provides a reward (or not) with a particular probability. This offers a test of the prediction.

It is also assumed that when the task begins, an agent may have no knowledge about reward probability. That is, the agent may have no idea about which objects are more likely and which objects are less likely to provide rewards. Furthermore, the agent may have no knowledge about how cues are related to an object's probability of reward. The task is structured so that an agent can explore the environment (i.e., the set of objects) over time, first choosing one object, then another, and observing the rewards that may or may not be provided. It is assumed that the agent will acquire knowledge about the environment by means of this exploration. For instance, over time the agent will learn something about how the various cues signal reward probability and take advantage of this knowledge to increase the likelihood of being rewarded. As a result, this generic paradigm is an example of what is known as *probabilistic discrimination task* (Estes, Burke, Atkinson, & Frankmann, 1957). In a probabilistic discrimination task, participants explore an environment to learn its probability structure by translating different stimuli (or the cues that define them) into probabilities of reward.

It is assumed that, from the perspective of the agent, the purpose of the task is to seek reward. That is, receiving a reward is better for an agent than not receiving a reward. However, beyond this, no further assumptions are made. For instance, we are not concerned with characterizing an ideal agent that seeks the maximum possible number of rewards. We are instead concerned with a more modest goal, exploring the choice behavior of an agent that adapts to this uncertain environment, and whose learning permits the agent to receive more rewards later in the task than was the case earlier in the task. The nature of this learning is of primary interest: If an agent's behavior changes over time (i.e., if they receive rewards more frequently over time), then what sort of learning or adaptation is responsible?

This generic characterization of the task of interest indicates that it is related to the lens model that is presented in Figure 1-1. The common point of contact between the task and the lens model is the set of cues that are detected by the agent and the probabilistic relationship between these cues and the judgment or action that they support. Our characterization of the task places less emphasis on the probabilistic relationship between the focal point in the world and the detected cues that is made explicit in the lens model. As we explore how agents learn in the task, we place more emphasis on mechanisms that provide feedback than is typically found in treatments of the lens model. This is because a description of the nature of this feedback amounts to an account of how learning proceeds.

1.2.3 Example: The Card-Choice Task

The previous section provided an account of a generic situation in which agents explore objects in an environment, are rewarded (or not) when particular ob-

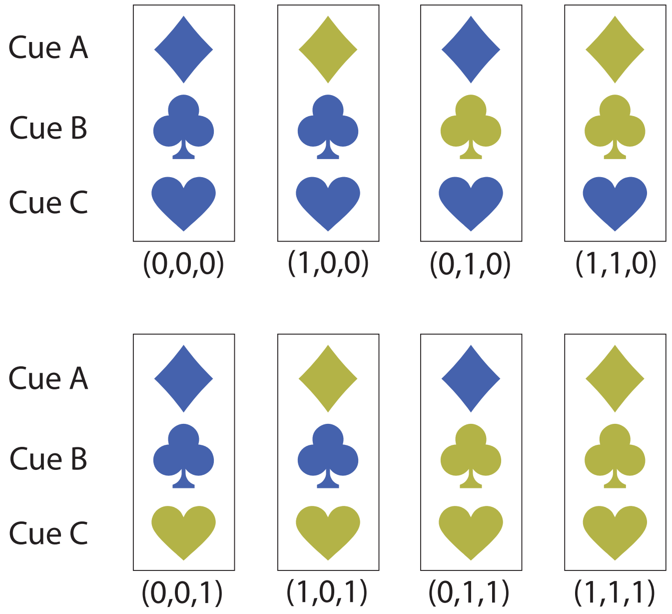


Figure 1-2. The stimulus properties for the card-choice task. Each stimulus is a “playing card” defined by three symbols that can be filled with one of two colors. One color indicates that the cue is in one state (e.g., present), and the other color indicates that the cue is in a different state (e.g., absent). The labels in the figure map card symbols to cue names. The vectors at the bottom of each card provide a numerical representation of cue states; this representation assumes that blue signals 0 and that green signals 1. In the card-choice task, participants see only a tableau of the eight cards; the labels are not presented.

jects are chosen, and use these rewards to learn about how environmental cues (e.g., the patterns of cues that identify objects) signal reward probability. Let us now turn to providing some concrete examples of this generic situation.

The first example is a particular paradigm, called the card-choice task, which is central to the current monograph. For instance, detailed results about human performance on this task are provided in Chapter 7. In the card-choice task, participants see a set of eight different playing cards on a computer screen. Each card consists of three symbols (diamond, club, heart) and each symbol has one of two colors (green, blue). Each card is uniquely identified by its symbols. The eight possible cards that can be constructed in this fashion are illustrated in Figure 1-2, which also indicates a mapping between card symbols and cue names, as well as a numeric vector that indicates the state of each cue for each stimulus.

In the card-choice task, participants are told that different cards may be more or less likely to provide a reward and that the symbols on each card might provide some information about what this likelihood might be. They are also told that because reward is probabilistic, a card that generates a reward when it is chosen at one point may not give a reward when it is chosen again. Participants are instructed that the environment needs repeated exploration so that they can learn about the different reward probabilities. While they are informed that the symbols that identify the card can provide hints about reward probability, they are also informed that a card’s location does not. To reinforce this instruction, after a participant makes 16 card choices, the cards are randomly shuffled. After shuffling, the cards are in different positions, but each card will still give a reward with the same probability as it did before. Participants are given the goal of obtaining as high a score as they can. They are also told that as they explore the environment by choosing cards, they should be able to learn which cards reward more and which cards reward less, and they can use what they learn to achieve a higher score.

A participant’s task in this environment is quite simple: The participant simply uses the computer’s mouse to choose a card. When a card is chosen, it will either give a reward or not. If a reward is not given, then nothing happens. If the participant is rewarded, then he or she hears a tone and the score on the screen increases. The computer program that conducts the experiment records the card selected, whether it provided a reward, and the card’s location for each of the participant’s choices. The experiment proceeds by the participant successively making card choices. After 320 choices, the participant is finished with this task.

The independent variable in this task involves manipulating how the various cues are used to determine the probability that a reward will be given when a card is selected. For example, in an independent probability condition, when a cue is present (e.g., when its color is green as in Figure 1-2) this is a signal of a particular probability of reward. Each of the three cues can independently signal a different reward probability. The actual probability of reward associated with a card can be computed by combining the signaled rewards of each cue using the additive probability rule (Rozanov, 1977). Thus, a particular card will never generate a reward every time it is presented but will only do so probabilistically, where this probability is signaled by the pattern of symbols on the card. This makes this task an example of a probabilistic discrimination task (Estes et al., 1957). Other rules that map cue states onto reward probabilities can be easily created.

This simple environment has several different advantages. First, despite its simplicity it can be used to collect a rich source of information about choice behavior. Second, by altering the rules that are used to map cue patterns to reward probabilities, it can easily be used to explore conditions that make choice behavior easy or difficult. For instance, how are participants affected when the probability of reward depends on combining the states of more than one cue? Third, the dynamics of choice behavior are completely under a participant's control: They decide which card to choose next and move through the task at their own speed. Fourth, examining the decisions that participants make by simply asking them to make choices is ecologically valid, in the sense that this represents everyday behavior. Some studies have instead explored decisions under uncertainty by asking participants to generate numbers that represent probabilities that participants use (Gigerenzer & Hoffrage, 1995; Kahneman & Tversky, 1972, 1973). This approach neglects the possibility that participants make choices under uncertainty that are guided by probabilities but that these probabilities are determined by unconscious mechanisms. In our view, our task for participants is more natural and is appropriate for the study of choices that are guided by probabilities that may be unconsciously computed.

Note that the card-choice task is a probabilistic discrimination task (Estes et al., 1957) analogous to the generic examples described earlier. For instance, consider the relationship between it and the cherry choice example from Section 1.2.1. In the card-choice task, each card is analogous to a different flat of cherries. The symbols on each card are analogous to the different properties of each flat of cherries that are used to predict sweetness. When participants choose a particular card and note the resulting reward, this is analogous to choosing a particular flat of cherries and then testing the prediction that these cherries (because of their properties) are sweet. The advantage of the card-choice task for studying probabilistic discrimination is that it limits the number of cues that are involved in learning and limits the states of these cues to two. However, the analogy between the card-choice task and the earlier generic description shows that the logical structures of these tasks are all equivalent.

Important to note, other tasks also have this structure. These tasks have received considerable discussion in the literature. The next two sections briefly describe two such tasks that have appeared in two very different research domains. Each of these tasks can be considered as a probabilistic discrimination task. The purpose of describing these other tasks is to indicate that models of how agents learn probabilities in the card-choice task are relevant to a wide range of research topics.

1.2.4 Example: The Reorientation Task

Our first example of a paradigm that is strongly related to those introduced earlier in this monograph is the *reorientation task* (K. Cheng, 1986). The reorientation task is used by experimental psychologists to study animal and human navigation. In a typical reorientation task, an agent is placed inside a rectangular arena. Covers or small barriers are located at different places in the arena; one hides a food reward, and the others do not. An agent's task is to discover the location of the reward in the arena. By repeatedly placing an agent in the arena, it learns to use available cues to identify the reward's location. These cues—the evidence supporting the agent's choice—can include the shape of the arena, colors of arena walls, or the appearance of landmarks at various locations in the arena. The task can be quite flexibly used to explore the extent to which agents navigate using different kinds of evidence (K. Cheng, 2005; K. Cheng & Newcombe, 2005) and has been used to study reorientation in an enormous range of animals, including insects (Wystrach & Beugnon, 2009), fish (Sovrano, Bisazza, & Vallortigara, 2007), birds (Gray, Bloomfield, Ferrey, Spetch, & Sturdy, 2005), and humans (Twyman, Friedman, & Spetch, 2007).

Although the reorientation task is firmly linked to the animal navigation literature, it is clearly an example of a probabilistic discrimination task like those just introduced. First, animals make a choice in the task (by selecting a location) that is either rewarded or not. Second, the choice is determined by the information provided by the cues that are available in the arena. Third, the relationship between cues and rewards can be probabilistic.

This last point is illustrated in Figure 1-3. This figure provides an example of a reorientation task arena. In this arena, the only cues available are its shape. This is because all the walls are indistinguishable in their appearance, and no identifying landmarks are present at any of the locations. An agent can only identify the location of the reward (labeled *Correct* in the figure) using wall properties (e.g., length). From the animal's perspective, the only evidence that indicates the correct location is a short wall on its left and a long wall on its right. The problem is that this evidence is not perfect. This is because the same evidence is provided at the incorrect location labeled *Rotational*, a location that is never rewarded. Even with perfect processing of the evidence, the agent will at best pick the correct location, and be rewarded, on only 50% of its choices. In other words, this evidence (called *geometric evidence*) provides only a probable identification of correct location. Any agent that is using such evidence is making a choice under uncertainty consistent with the earlier characterization of a probabilistic discrimination task.

One reason that the reorientation task is introduced at this point is to illustrate its relationship to probabilistic discrimination in general, which indicates that what we learn about models of the card-choice task later in this

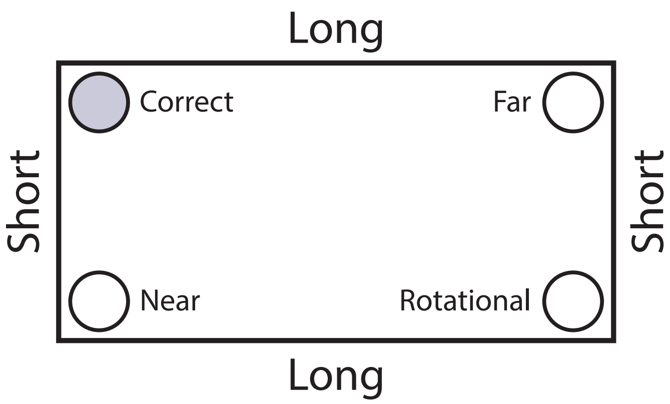


Figure 1-3. An overhead view of a reorientation task arena. The four possible locations are indicated by the circles; the filled-in gray circle indicates the only location that hides a food reward. The shape of the arena provides the only cues that can be used to identify the correct location. This evidence is probabilistic, though, because the same shape information is present at the locations labeled Correct and Rotational, but only one of these two locations is rewarded. In other words, this shape information only signals a probability of reward of 0.50.

monograph can have implications that are more general. A second reason, developed in more detail later in this chapter, is that a very simple kind of artificial neural network called a *perceptron* has been shown to model performance in the reorientation task (Dawson, Kelly, et al., 2010). This suggests that this simple type of network is a plausible model of probabilistic discrimination. For this reason, the early chapters of this monograph explore the relationship between this type of model and probabilistic discrimination. Later (Chapter 7), evidence is presented that perceptrons are appropriate models of human probability learning in the card-choice task.

1.2.5 Example: The Multiarmed Bandit

The reorientation task is an instance of a probabilistic discrimination task that has arisen in experimental psychology in an area that is not typically portrayed as concerned with probability learning. The probabilistic discrimination task also arises in another literature that is concerned with formal and computer models of statistical learning. This is because the probabilistic discrimination task is also analogous to the classic *multiarmed bandit problem* (Auer, Cesa-Bianchi, & Fischer, 2002; Gittins, 1979, 1989).

In the classic form of multiarmed bandit problem, an agent is in a room with a slot machine that has several arms that can be pulled, each of which rewards with a different probability. An equivalent version of this problem is to assume the presence of many “one-armed bandit” slot machines. When the arm of one of these machines is pulled, it pays a reward of either 1 or 0 units; each machine has a different (and usually fixed) probability of paying off, which is not initially known by the agent. The agent can explore this environment by choosing to pull the arm of one machine or another. Over time, the agent makes a number of these choices. Researchers who are interested in this problem attempt to discover strategies that an agent can use to maximize the total payoff over the game’s duration (Kaelbling, Littman, & Moore, 1996). These strategies seek an optimal combination of exploration and exploitation: An agent must explore different machines to determine their various payoff probabilities but must also exploit the results of this exploration to maximize reward. A “greedy strategy” pulls only the arm of the machine with the highest expected payoff probability. However, as the duration of the game increases, an alternative strategy would be to explore other machines as well, in case early probability estimates were inaccurate.

Theories of multiarmed bandits usually view each machine as a unique whole. However, it has been argued that each bandit can easily be viewed as feature collections, with the machine’s probability of reward being determined by this feature set and not upon the (whole) machine itself (Dawson et al., 2009). The utility of a feature for predicting reward can depend on the payoff of several machines because a feature may be shared by more than one bandit. When bandits are associated with sets of cues, the multiarmed bandit problem becomes formally equivalent to the card-choice task. To be more precise, each card in the card-choice task is analogous to a bandit that pays out with a set probability; this probability is signaled by the pattern of cues that differentiate this bandit from the others in the room.

Linking the multiarmed bandit problem to a probabilistic discrimination task is important in three respects. First, it makes it evident that an account of probabilistic discrimination must pay attention to key issues related to bandits, such as acknowledging the importance of exploration and exploitation in choice behavior. Second, the common framework provided by the probabilistic discrimination task provides a medium that shows the relationship between very different literatures, such as research on multiarmed bandit algorithms and research on animal navigation. Third, as was the case for the reorientation task, it has been shown that perceptrons can be used to model aspects of the multiarmed bandit problem (Dawson et al., 2009). This provides more evidence for exploring the properties of these simple networks when they adapt to uncertain environments.

1.2.6 Summary and Implications

The card-choice task is an example of a generic discrimination task (Estes et al., 1957). In such a task, various cues provide information about an item (e.g., an indication of whether it will lead to a reward), but this information is uncertain. This task was introduced as an example paradigm for exploring how agents learn to adapt to uncertain environments. We will see that one reason for introducing this task is that it can easily be defined in a way that can be presented to a simple artificial neural network.

Although this probabilistic discrimination task is quite basic, variations of it can easily be found in the literature. Sections 1.2.4 and 1.2.5 demonstrated, for example, that the reorientation task and the multiarmed bandit are strongly related to the card-choice task. This suggests that the card-choice task might provide further understanding that is applicable in a variety of fields.

As a probabilistic discrimination task, the card-choice task is also related to a broader literature on probability learning. This literature, and its implications for the aims of the current monograph, is introduced next in Section 1.3.

1.3 Probability Learning

The previous section introduced the card-choice task for studying choice behavior under conditions in which reward is uncertain. We now turn briefly to consider the relationship between such a task and an area that has received considerable interest in psychology since the 1950s: the study of probability learning. We briefly review some of the core findings in the probability learning literature and use them to make basic predictions about the sorts of results that we should expect from human participants when they perform the card-choice task. We also use the probability learning literature to raise some additional issues that we will need to consider when examining human data.

1.3.1 Early Research

The study of probability learning began in the 1930s (Brunswik, 1939; Humphreys, 1939), but it was in the 1950s that research on this topic exploded (Estes, 1964). In his review, Estes cited 80 studies that were published beginning in the early 1950s, all of which examined probability learning in human participants.

The typical paradigm used to study probability learning was first described by Humphreys in 1939. In this paradigm, an individual participates in a series of trials. On each trial, the participant chooses a response from a set of alternatives provided by the experimenter. This choice is the participant's prediction of what will happen next. For instance, a typical choice involves predicting whether a light will turn on. After the participant makes this choice, the experimenter indicates whether the choice or prediction was correct. In a typical noncontingent version of this paradigm, each response has a fixed probability of being reinforced. This probability is completely independent of the participant's present or past choices. In a different, contingent version of this paradigm, the probability of reinforcement is determined from the participant's response (Detambel, 1955; Neimark, 1956). That is, if the participant makes one response, then one probability of reward is used, but if the participant makes a different response, then a different probability of reward is used. In most studies that used variations of Humphreys's original task, participants could make one of two possible choices. On occasion, studies were conducted in which the number of possible choices was greater than two (R. A. Gardner, 1957; Neimark, 1956).

The early probability research reviewed by Estes (1964) produced three general findings. First, it appeared probability learning was also affected by variables which affected other forms of human learning. As a result, researchers were encouraged to model probability learning with simple learning models such as the pattern model (Estes, 1957b) or statistical sampling theory (Estes, 1950). For instance, the pattern model is a linear model that predicts the probability of a response on the next trial from the current response probability using the equation $P_{n+1} = (1 - \theta) \cdot P_n + \theta$. In this equation, P is the probability of the response, its subscript indicates the trial, and θ is a learning constant with a value between 0 and 1.

Second, one of the main regularities revealed by probability learning experiments is *probability matching*. When probability matching is observed, the probability that the participant makes a particular response closely corresponds to the probability that the response is rewarded (Estes, 1957a). This phenomenon demonstrated "a degree of replicability quite unusual for quantitative findings in the area of human learning" (Estes, 1964, p. 94). Probability matching has been observed in a variety of organisms, including insects (Fischer, Couvillon, & Bitterman, 1993; Keasar, Rashkovich, Cohen, & Shmida, 2002; Longo, 1964; Niv, Joel, Meilijson, & Ruppin, 2002), fish (Behrend & Bitterman, 1961), turtles (Kirk & Bitterman, 1965), pigeons (Graf, Bullock, & Bitterman, 1964), and humans (Estes & Straughan, 1954). Important to note, probability matching is typically seen when the data from a group of participants are combined and is an asymptotic effect that is reached after participants have experienced a number of probability learning trials.

Third, probability learning experiments raised interesting questions that required further study. For instance, it was typical to observe many individual differences between participants in their response behavior. In addition, although statistical learning models performed well when predicting performance on basic probability learning tasks, they were challenged when confronted with data from more complex paradigms, such as those that used contingent reinforcement. To Estes (1964), such challenges indicate "the probability learning situation may be too complex to be handled in all desirable detail by any manageable model" (p. 116).

Two of the complications facing learning theorists' ability to account for probability learning deserve to be singled out. This is because of their relevance to the cognitive revolution's impact on this literature and to their relation to mathematical information theory.

First, many of the individual differences in probability learning behavior suggested that higher level cognitive strategies might be influencing participants' performance. For example, certain studies indicated that some participants exhibited a "gambler's fallacy" in that they were more likely to predict an event the longer that it had not occurred or been reinforced (Jarvik, 1951). As another example, after a long series of trials, participants often tend to "overshoot" by choosing a higher-reward response more frequently than its reward probability would predict (Jarvik, 1951), or will suddenly depart from their well-established probability matching behavior. In both of these cases, participants may be recognizing that their success rate has stabilized and therefore change response strategy because they recognize that their performance is not improving. That higher-level cognition might be involved in probability learning led some researchers to explore this phenomenon in situations in which general rules (instead of probability distributions) needed to be determined (Goodnow & Postman, 1955) or to identify various strategies that participants might be employing (Gluck, Shohamy, & Myers, 2002).

Second, at the root of learning theorists' accounts of probability learning is the basic assumption that a reward satisfies a drive or motive. However, some studies of probability learning suggested that the function of reward was not to satisfy a need. Instead, when a reward occurred, it simply served as a signal that conveyed information (Bitterman, 1956; Hillix & Marx, 1960). The view of rewards as signals or sources of information heralded the rise of cognitive approaches to probability learning, not to mention cognitive theories of learning that assumed that learning was not so much a function of reward, but was instead driven by surprises that occurred when predictions failed (Rescorla & Wagner, 1972).

1.3.2 Cognition and Probability Learning

Estes's (1964) review of probability learning described reliable phenomena, successful models from learning theory, and a variety of issues for inspiring new research. About a decade later, however, the cognitive revolution had taken a firm hold of experimental psychology and had dramatically altered the study of probability learning. This had three implications that are of interest to this monograph.

First, research on how humans adapt to an uncertain environment moved away from the learning theory approach described by Estes (1964) to a cognitive approach that was concerned with how humans reason when faced with uncertainty (Tversky & Kahneman, 1974). In particular, this cognitive research used probability theory to define optimal performance (Gigerenzer, 1989; Gigerenzer & Murray, 1987) and then demonstrated that when humans are faced with uncertainty they often make judgments that are not optimal (Kahneman & Tversky, 1972, 1973; Lindley, Tversky, & Brown, 1979; Tversky & Kahneman, 1983). In general, the cognitive position was that limits in processing ability required humans to base their judgments on heuristics instead of probability theory and that these heuristics included biases that produced suboptimal performance.

Second, some researchers attempted to modify traditional accounts of probability learning to make this research relevant to cognitive psychology. Mathematical models of probability learning focus on behavior or performance instead of underlying mechanisms (Estes, 1964). For instance, in the pattern model that was briefly introduced in Section 1.3.1, the only variable is the probability of responding (P_{n+1} or P_n). The pattern model's equation uses a learning rate but does not appeal to specific mechanisms of learning or of representing information that is required for an agent to match probabilities. Estes proposed a cognitive theory of probability matching that addressed this issue by exploring how different forms of memory investigated by cognitive psychologists might be used to store the information required to compute response or reward probabilities (Estes, 1976).

Third, the cognitive revolution arose beginning in the late 1950s by exploring the new ideas that emerged from the invention of the digital computer. In particular, when cognitive psychologists viewed humans as information processors, they viewed information processing in a very particular way: as the rule-governed manipulation of symbols (Dawson, 2013; Pylyshyn, 1984). However, by the mid-1980s a different perspective on information processing—parallel distributed processing—challenged the established symbolic perspective (Hinton & Anderson, 1981; McClelland & Rumelhart, 1986; Rumelhart & McClelland, 1986). With the arrival of this new view of information processing, some researchers explored the possibility of using artificial neural networks to explore probability learning (Gluck & Bower, 1988). For instance, Gluck and Bower used a simple network that made counterintuitive predictions about choices in a probabilistic categorization task. Of interest is that Estes, a key figure in the probability learning literature, felt that artificial neural networks of the sort explored by Gluck and Bower were too simple to provide an adequate account of probability learning (Estes, 1991).

1.3.3 Implications

The three observations that ended Section 1.3.2 are of interest because they foreshadow some of the themes of the current monograph. Its main goal is to use artificial neural networks to explore probability learning. One purpose of these networks is to serve as a possible mechanism for probability learning, which was a concern of Estes (1976). The networks that we begin this exploration with are quite simple, but we show that they can provide a great deal of insight into probability learning, in contrast to the position of Estes (1991). However, the networks themselves as traditionally conceived will not suffice to explain the strategic aspects of probability learning (e.g., the trade-off between exploration and exploitation).

To set the stage for the more detailed exploration of how artificial neural networks can contribute to the literature on adapting to uncertainty, the remaining sections of Chapter 1 develop the elements that we explore in technical detail in later chapters. First, we briefly establish that the card-choice task reveals that human participants adapt to uncertainty and that this adaptation reveals probability matching. These results are to be expected if the card-choice task measures some of the basic processes related to probability learning as this section and the previous one have implied.

Later, we introduce the basic properties of the simple network that will be used to launch our investigation, the *modern perceptron*. We then quickly demonstrate that modern perceptrons can generate some of the key properties of probability learning, such as estimating probabilities using the available cues. Later chapters elaborate these elements, providing computational analyses and computer simulations that indicate that the modern perceptron is a far more interesting medium for the study of probability learning than might have been realized by Estes (1991).

1.4 An Example of Probability Matching

The previous section suggests that the card-choice task is related to the broader psychological literature on probability matching. Given this putative relationship, we can generate some basic hypotheses about human performance on the card-choice task. First, our participants should learn something about the environment that they are exploring by choosing cards, and as a result of this learning they should be more likely to receive rewards later in their exploration than was the case when they started the task. Second, the reason for this improvement performance should be because participants learn the probability structure of the task (i.e., the probability of reward associated with each card or signaled by each cue), and then use this information to choose cards that have a higher probability of reward more frequently than cards that have a lower probability of reward. In short, they should match their frequencies of choosing cards to the frequencies of rewards associated with these cards.

The current section provides some pilot data collected from 17 participants (all of whom were introductory psychology students) in one version of the card-choice task that was introduced earlier. It uses these data to show that the card-choice task provides evidence that supports both of these hypotheses.

1.4.1 Reward Conditions

In any particular instance of the card-choice task, the cues on a card signal its reward probability. In the instance of the task that we now consider, a card's reward probability was determined by the presence of one of the cues, and the logical AND of the other two cues. In this instance of the card-choice task, if the diamond was green, then this was a signal of a reward probability of 0.48. If the diamond was blue, then this signaled a reward probability of 0. In addition, if both the club and the heart were green, then this also signaled a reward probability of 0.48. If this state was not true (i.e., if both cues were blue, or if the two cues were of different colors), then this signaled a reward probability of 0.

The two sources of information about reward probability were assumed to be independent of each other. In other words, if both of the cue types on a card signaled a reward, then there was an even higher probability of reward. The actual reward probability for each card was computed with the additive rule for independent probabilities (Rozanov, 1977). For this particular version of the card-choice task, this means that the reward probability for any card is equal to that card's probability associated with its diamond plus its probability associated with its club and heart minus the product of these two probabilities. The ideal probability of reward computed using this equation for each card is presented in the second column of Table 1-1.

Table 1-1. The probability of reward associated with each stimulus in the example.

Stimulus Card	Ideal	Actual
~D~C~H	0.00	0.000
~D~CH	0.00	0.000
~DC~H	0.00	0.000
~DCH	0.48	0.462
D~C~H	0.48	0.482
D~CH	0.48	0.478
DC~H	0.48	0.493
DCH	0.7296	0.720

Note. D = diamond; C = club; H = heart; ~ = a symbol is not present (i.e., has the color blue instead of green, so DC~H is the card with a green diamond, a green club, and a blue heart).

In the actual running of the experiment, when a participant chooses one of the eight stimulus cards, they are either rewarded or not. The computer program decides whether to give a reward as follows: First, it generates a random number between 0 and 1. Second, it compares this number to the ideal probability associated with the chosen card. If the random number is equal to or less than the ideal probability, then a reward is given. Otherwise, no reward is given. This method generates actual reward probabilities that are very similar to, but not identical to, the ideal probabilities. The third column of Table 1-1 provides the actual probability of reward for each stimulus by summing the number of times each card was rewarded, and dividing by the number of times each was selected, for each of the 17 participants who made 320 different card choices when they participated in the study.

1.4.2 Participants Improve Reward Seeking

The card-choice task provides an uncertain environment that participants can explore with the goal of seeking rewards. We expect that participants will learn about the environment as they explore it. If participants learn about the reward probabilities associated with the stimulus cards, then they can use this knowledge to increase their chances of being rewarded with each choice that they make. One simple test of this hypothesis is to compare the number of rewards that participants receive early in the study to the number of rewards that they receive toward the end.

In this pilot study, when we examine the total number of rewards that each of the 17 participants received from their first 64 card choices (the first 20% of the study), we discover an average of 19.71 rewards. In contrast, when we compute the total number of rewards that each participant received for their last 64 card choices, the average is 32.94 rewards. In other words, participants learn enough by exploring the environment to receive about 13 more rewards from their final 64 choices than they received from their first 64 choices. A paired t test reveals that this difference between the number of early and the number of late rewards is statistically significant, $t(16) = -8.6483$, $p = 1.989\text{e-}07$. Clearly, participants are learning the probability structure of the environment and are using this knowledge to obtain more rewards.

1.4.3 Learning Probabilities

What does it mean to say that participants are learning the probability structure of the environment? One definition is that as the study proceeds, participants are more likely to choose cards that have a higher reward probability and are less likely to choose cards that have a lower reward probability.

We can quantify this definition by using the probabilities associated with each card that were provided in Table 1-1. By hypothesis, let us make a prediction from the probability learning literature and say that an ideal participant will always match probabilities. In other words, let us say that the number of times an ideal participant will choose one of the eight cards is equal to that card's actual probability of reward multiplied by the number of choices available to a participant. Let us then use these probabilities to compute the number of choices of each card, assuming the ideal participant has only 16 choices available. We do this by multiplying each value in the third column of Table 1-1 by 16.

We are now in a position to compare the choices made by an actual participant to these ideal predictions. For each successive set of 16 choices made by a participant in the pilot study, we can compute the sum of squared differences (SSD) between the number of choices of each card that were actually made and the ideal number of choices. The smaller this measure, the more similar the participant is to the ideal.

Figure 1-4 presents the SSD computed for each set of 16 card choices in the study, averaging this measure across the 17 participants. Each dot represents a pairing of SSD

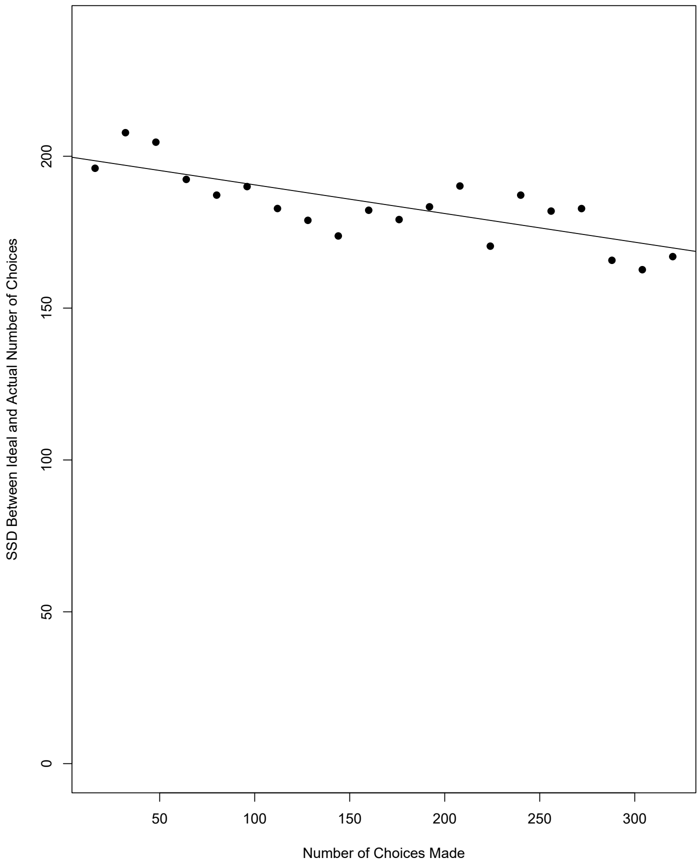


Figure 1-4. Change in the sum of squared deviation between responses of an ideal participant and those of the actual participants. A smaller number indicates that participant choices are becoming more similar to those predicted by the probability structure of the task.

with a set of 16 choices. The regression line fit to these dots is statistically significant (slope = −0.09442, intercept = 200.0325, $R^2 = 0.618$), $F(1, 18) = 29.17$, $p < .001$. It shows that as the study proceeds, on average participant choices better reflect the probability structure of the environment. The more they learn about this structure, the more likely they are to choose cards with a higher probability of delivering a reward. This accounts for why they receive significantly more rewards later in the study than they do earlier.

1.4.4 Evidence of Probability Matching

Is there any evidence of probability matching in our pilot study? To answer this question, we determined the number of times each of the eight stimulus cards was selected in the study, summing over all of the 320 card choices made by each of the 17 participants. We also determined the number of times each of the cards provided a reward, again summing over all of the choices made by all of the participants. Both of these sets of frequency data are provided in the histograms presented in Figure 1-5.

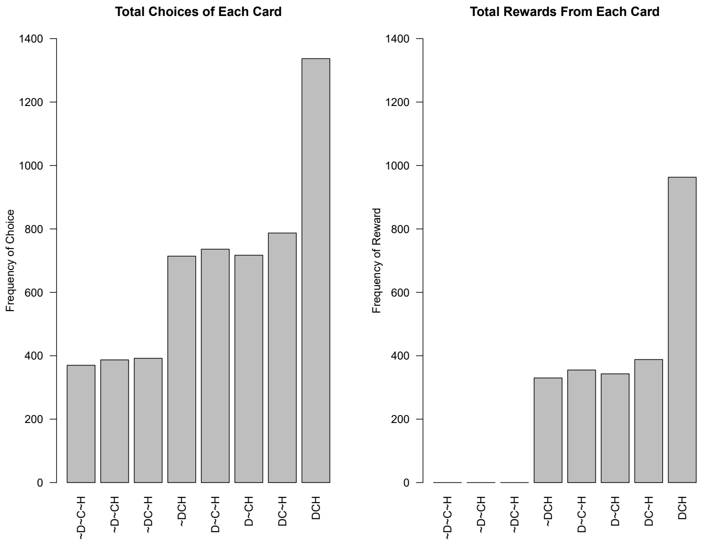


Figure 1-5. The histogram on the left presents the frequency that each of the eight stimulus cards was selected, summing over the 320 total choices of each of the 17 participants. The histogram on the right indicates the frequency of times that each of these cards was rewarded when selected. The similarity of shape between the two histograms provides evidence of probability matching, as is detailed in the text.

If probability matching is occurring, then there should be a strong relationship between the two graphs in Figure 1-5. An inspection of the two indicates their similarity of shape: The smallest numbers of choices are made of the three cards that never provide a reward, a moderate number of choices are made of the four cards that provide an intermediate reward, and the highest numbers of choices are made of the one card with the highest probability of reward.

If one correlates the eight frequencies of choice from the histogram on the left with the eight frequencies of reward from the histogram on the right, the resulting correlation coefficient equals 0.999. This indicates a striking quantitative relationship between choice and reward. One reason that this coefficient is so high is because slight variation in probability of reward (e.g., the four moderate reward bars on the right) are matched by similar variations in number of choices in the corresponding bars on the left.

In short, our pilot study provides clear evidence of probability matching: The number of choices of each card made by the participants is strongly related to the number of times the card is actually rewarded.

That human participants exhibit probability matching in the card-choice task is not to be taken as a particularly surprising or novel demonstration. There is, of course, a long history of experimental results that show that human participants, not to mention many other types of organisms, will probability match in a variety of different situations

(Estes, 1964). For the time being, the results presented in this section are merely another example to add to this literature. Later we see that variations of this task—created by altering the rules used to convert cue patterns into probability of reward—will provide novel results by demonstrating that under some rules, human participants are much poorer probability matchers than is the case for other rules.

Of greater interest than the results of Figure 1-5 are the nature of models that can be used to predict such results, which we later show can be used to predict situations in which humans are poorer probability matchers. In particular, we relate human performance on variations of the card-choice task to the performance of a particular type of model trained on an analogous task. Let us now consider the kind of model that we explore in an attempt to understand making choices under uncertainty.

1.5 The Modern Perceptron

Up to this point, we have been focusing on the uncertainty stated in Chapter 1's title. We have discussed sources of uncertainty, provided a concrete example of how such uncertainty is built into a probabilistic discrimination task, and reviewed some of the key findings of the psychological literature on probability learning, including the phenomenon of probability matching. We have also presented some evidence showing that agents can adapt to uncertainty. For instance, Section 1.4 provided results from an example card-choice study. These results indicated that as participants explored this uncertain environment, they learned which cards were more likely to provide rewards and they used this knowledge to receive more rewards.

We now turn to focusing on the other part of Chapter 1's title: adaptation. We have reviewed evidence that shows that agents adapt to uncertain environments. However, how is adaptation, like probability matching, achieved? The early chapters in this monograph explore one kind of model of adaptation. This model is a simple artificial neural network called a perceptron. Section 1.5 introduces this model and describes a modern variant that is of special interest to the current work. Section 1.6 briefly provides some evidence of why this simple model is of interest. Chapters 2 and 3 expand on these ideas in more detail by relating the properties of perceptrons to different ideas from probability theory.

1.5.1 The History of the Perceptron

As noted in Section 1.3.2, the cognitive revolution that began in the late 1950s exploited the metaphor that thinking is information processing and took information processing to be the rule-governed manipulation of symbols (Dawson, 2013; H. Gardner, 1984). This metaphor led cognitivists to explore the nature of human cognition's symbols and the types of rules that were used to manipulate them. These two topics define what is known as the *cognitive architecture* (J. R. Anderson, 1983; Pylyshyn, 1984; VanLehn, 1991). An architecture that is defined in terms of symbols and rules is called a *classical architecture* of cognition (Dawson, 1998, 2013).

Alternatives to the classical architecture have also been proposed and are of increasing importance in cognitive science. One that rose to prominence in the 1980s is *parallel distributed processing*, an architecture adopted by connectionist cognitive science (Bechtel & Abrahamsen, 2002; Hinton & Anderson, 1981; McClelland & Rumelhart, 1986; Rumelhart & McClelland, 1986). According to connectionist cognitive scientists, human information processing is not analogous to the operations of a digital computer. Instead, this information processing is much more brain-like. Parallel distributed processing involved numbers of simple processing units (analogous to neurons) that send signals to one another via weighted connections (analogous to synapses). These systems of interconnected processors are called *artificial neural networks*. The weights of an artificial neural network are determined via learning; a network is trained on a set of example training patterns, and its connection weights are adjusted to reduce the size of its errors. The connectionist revolution took hold with the discovery of a learning rule called backpropagation of error that permitted the training of networks with several layers of processing units (Amari, 1967; J. A. Anderson, 1995; Rumelhart, Hinton, & Williams, 1986; Werbos, 1994). Such networks are capable of learning to perform a number of tasks that are of interest to cognitive scientists.

Important to note, artificial neural networks flourished long before the cognitive revolution (and the connectionist response to it) occurred. They were first proposed as mathematical entities by pioneering cyberneticists Warren McCulloch and Allen Pitts (1943). McCulloch and Pitts were interested in translating the all-or-none responses of biological neurons into logical operations. From their perspective, when a neuron generates an action potential, this is functionally equivalent to asserting that a logical condition is true. Similarly, when the same neuron does not generate an action potential, this is functionally equivalent to asserting that the same logical condition is false. McCulloch and Pitts formalized these notions with what is now known as the McCulloch-Pitts neuron. This artificial neuron receives signals from external sources; these signals are multiplied by connection weights. The neuron sums the weighted signals and compares them to the neuron's threshold. If the total signal exceeds the threshold, then the McCulloch-Pitts neuron outputs a value of 1. Otherwise, it outputs a value of 0. McCulloch and Pitts used this formalism to build McCulloch-Pitts neurons that performed 14 of the 16 possible basic logical operations in Boolean algebra. They also proved that a network constructed by linking different neurons together so that the output of one McCulloch-Pitts neuron serves as an input to another was capable of emulating a universal Turing machine.

The mathematical formalism developed by McCulloch and Pitts (1943) demonstrated the potential logical power of artificial neural networks. However, their research goal was not to bring such a system into being. Indeed, one of the limitations of McCulloch-Pitts neurons was that they could not be trained. To construct an interesting network of McCulloch-Pitts neurons, one would have to set all of its connection weights and thresholds by hand.

This situation was changed by research that aimed to build machines that could learn (Rosenblatt, 1958, 1962; Widrow & Hoff, 1960). One of the most famous and historically important of these inventions was Frank Rosenblatt's (1962) *perceptron*. In general, a perceptron is strongly related to a McCulloch-Pitts neuron: It consists of one or more input units that are connected to one or more output units. The input units respond to an external stimulus and then send the output units a signal about the stimulus through weighted connections. The output units process the incoming signal (by comparing it to a threshold) and generate a binary response. As a result, the output units represent the perceptron's response to the stimulus that was presented to it. Figure 1-6 provides a concrete example of a perceptron and provides a general sense of how it learns from experience.

The perceptron in Figure 1-6 consists of a set of five input units, each of which represents the value of a cue provided by the environment. In this example, each of the five input units is connected to a single output unit (labeled output unit *j*) by a weighted connection. The weight associated with input unit 1 is labeled w_1. The input units send their activities (a_1, a_2, etc.) through these connections. The activities are scaled by the connection weights and are added together to form the total signal or net input for output unit *j* (net_j). Then an activation function is used to convert the net input into the output unit's activity (a_j). This activity is the response of the output unit to the signals from the input units. Learning proceeds by comparing the perceptron's actual response (a_j) to a desired response (T_j) that is provided by the environment. It is represented by T because it is a signal that is used to "teach" the perceptron.

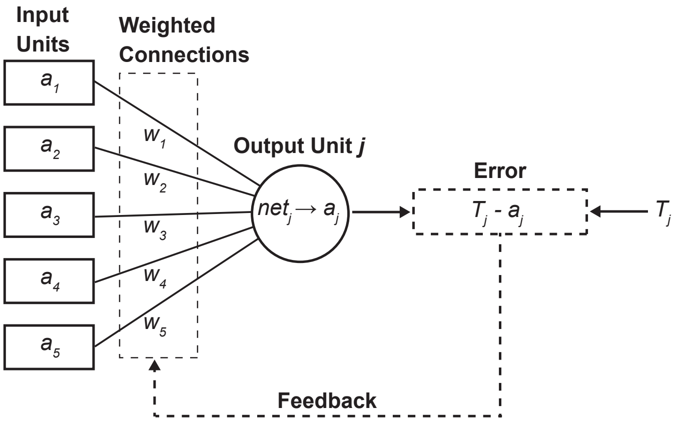


Figure 1-6. The basic components of a perceptron.

In classical conditioning, for instance, a response would be rewarded by setting T_j to 1; the absence of reward is achieved by setting T_j to 0 (Dawson, 2008). The output unit's error is simply the difference between the desired and actual responses ($T_j - a_j$). The learning rule uses this error as feedback to modify the weights; the goal of this modification is to decrease the output unit's error.

Figure 1-7 presents an alternative rendering of the perceptron by fitting it into Brunswik's (1952) lens model of probabilistic functionalism. This has been accomplished by replacing part of Figure 1-1 with a version of Figure 1-6 and is used to establish the relationship between the network-based accounts of probability learning that this monograph explores and more general accounts of adapting to uncertainty like Brunswik's (1952) probabilistic functionalism. Figure 1-7 shows that we, like Brunswik, assume that the world is a source of information or cues that are offered by the environment. The arrows that show this relationship are dashed; this reflects that we are less interested than Brunswik was in defining the probabilistic nature of cue detection, although this type of uncertainty could easily be included in the perceptron. The dashed arrows from the output unit that depict the functional arc and feedback indicate that we are also interested in stabilizing the relationship between the world and the agent's response to it. However, the additional solid arrows from the output unit indicate that the perceptron offers a much more specific notion of using feedback to establish the functional arc. In particular, the error signal generated for the output unit is used as feedback to change the perceptron's weights so that it can adapt to the uncertain relationship between cues and reward.

Note that although the role of perceptron learning in Figure 1-7 is to stabilize the functional arc, this is accomplished by using a particular type of feedback

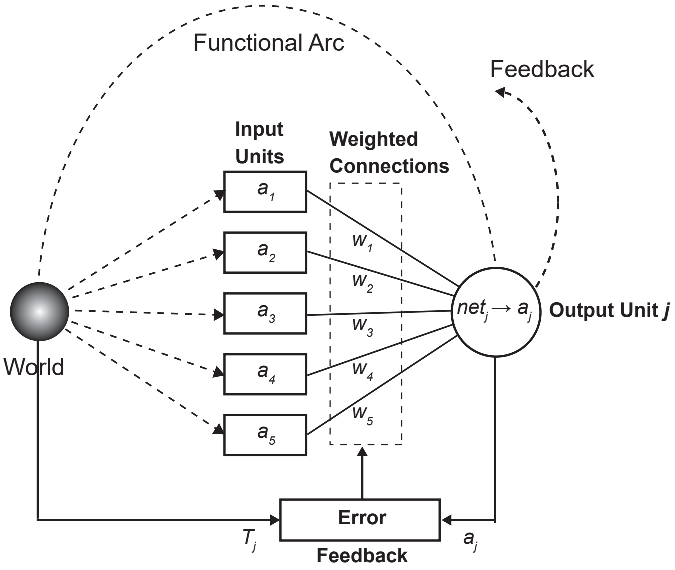


Figure 1-7. The basic components of a perceptron, placed in the context of Brunswik's lens model from Figure 1-1. See text for details.

to modify the relationships between the input units and the output unit response. This reflects the fact that the way that agents adapt to uncertain worlds is to alter their epistemic uncertainty, which is instantiated in the weights of the perceptron in Figure 1-7.

Rosenblatt's (1958) innovation was his invention of a procedure that could be used to train a perceptron from experience. In general, this procedure involves presenting the perceptron a series of training patterns. Each of these patterns defines how the perceptron's inputs are to be activated and defines the desired response to pattern. These patterns are presented one at a time to the perceptron. If the perceptron's responses are correct, then its weights are unchanged. However, if the perceptron produces an error, then this error is used to modify the weights. Rosenblatt (1962) demonstrated that these weight modifications always reduce error and used this to establish his famous perceptron convergence theorem. This theorem established that if a perceptron could represent a solution to a problem (its simplicity is such that this cannot be said for all problems; Minsky & Papert, 1969), then Rosenblatt's learning rule was guaranteed to find this solution. A solution is found when the perceptron generates the correct response to each pattern in the training set. The next section provides a more precise account of how a perceptron responds to a stimulus and of how it learns from this response.

1.5.2 Perceptron Responses

A perceptron can be described in purely mathematical terms. In this formalism, all of the signals that are being sent from the input units to an output unit are numbers; each signal is an input unit activity multiplied by a connection weight. As a result, the net effect of these signals can be determined by adding all of these numbers up into a single sum called the *net input*. Let the activity of input unit *i* be represented as a_i, and let the weight of the connection from input unit *i* to output unit *j* be represented as w_{ij}. Mathematically, the net input for output unit *j* (net_j) is determined Equation 1-1, which assumes that there are *n* different input units and which takes the sum over the *i* input units:

$$net_j = \sum a_i \cdot w_{ij} \tag{1-1}$$

Once an output unit has computed its net input, it then converts the net input into an internal level of activation, which is a number that represents the output unit's response to the signal received from the inputs. This is accomplished with an *activation function*, which has the general form *aj* = *f*(*netj*). In most artificial neural networks, the activation function is a nonlinear transformation of the net input. For both the McCulloch-Pitts neuron and Rosenblatt's (1958) perceptron, the activation function is the Heaviside function that is presented as Equation 1-2. The Heaviside function compares the net input to a threshold value represented as θ. If the net input is greater than the threshold, then the function returns a value of 1. Otherwise, it returns a value of 0. The Heaviside function is used because it defines the all-or-none response property of neurons.

$$a_j = H(net_j) = \begin{cases} 1: net_j > \theta \\ 0: net_j \leq \theta \end{cases} \tag{1-2}$$

1.5.3 Perceptron Training

Section 1.5.2 provided a formal description of how the perceptron in Figure 1-6 responds to a pattern of cues that activate its input units. We now turn to a mathematical account of how feedback is used to modify the perceptron's weights based on the response that it generates to a stimulus.

Rosenblatt's (1958, 1962) method for training a perceptron by providing it feedback about its mistakes is a learning rule called the *delta rule*. According to this rule, the observed response of a perceptron to a stimulus is compared to the desired response. The output unit's error is the difference between this desired response and its actual response, which is simply $T_j - a_j$. The logic of the delta rule is that the size of this error is used to modify the connection weights. If the error is large, then there will be a large change in the weights. If the error is zero, then there will be no change at all.

Equation 1-3 defines the change in a particular weight, represented as Δw_{ij}, according to the delta rule.

Note that this weight change is equal to the product of three values: a fractional learning rate η, the activity of the input unit at one end of the connection a_i, and the error at the other end of the connection $T_j - a_j$. The weight is changed by adding this computed weight change to the existing value of the weight.

$$\Delta w_{ij} = \eta \cdot a_i \cdot (T_j - a_j) \qquad (1\text{-}3)$$

With Equations 1-1, 1-2, and 1-3 in hand, we are in position to describe a general training procedure for the perceptron of Figure 1-6 (Dawson, 2005, 2008). Assume that we have a finite set of training patterns; each pattern provides the activation values for the n different input units, as well as the desired response for output unit j. Also, assume that the perceptron starts with all of its weights initialized to some small, randomly selected values. We select one of these patterns at random and use it to activate the inputs. This causes the output unit to compute its net input (Equation 1-1) and then to convert this into activity using Equation 1-2. Next, the output unit's activity is compared to the desired output in order to compute error, and then each of the perceptron's connection weights is updated by computing the respective weight change (Equation 1-3) and adding this change to the existing value of the weight. This procedure is then repeated for the next selected pattern. This continues until the perceptron generates the correct response for each pattern in the training set.

1.5.4 The Modern Perceptron

The Heaviside function (Equation 1-2) is but one of many activation functions to be found in the processors of artificial neural networks (Duch & Jankowski, 1999). In modern networks, it is quite common to see the Heaviside equation replaced with a continuous function that approximates the Heaviside's key features. One popular example is the sigmoid-shaped *logistic function* that is illustrated in Figure 1-8. Note that this function asymptotes to extreme values that can be described as "off" or "on." However, intermediate values of net input produce intermediate activities. Instead of a threshold that marks the boundary between being on or off, this function has a *bias* θ. When the net input equals this bias, the activation function produces activity equal to 0.5

The logistic function that produces sigmoid-shaped graph depicted in Figure 1-8 is provided by Equation 1-4. The logistic function was popularized in the neural network literature in the mid-1980s (Rumelhart et al., 1986). It is of particular interest in the current monograph because there is a long history using the logistic to model probabilistic mappings (Berkson, 1955; D. R. Cox, 1958a, 1958b; West, 1979). One can, in fact, use this relationship to translate network responses into probability theory (Dawson & Dupuis, 2012; Jordan, 1995; McClelland, 1998). We adopt this approach in Chapters 2 and 3 to translate perceptrons into versions of Bayes's theorem.

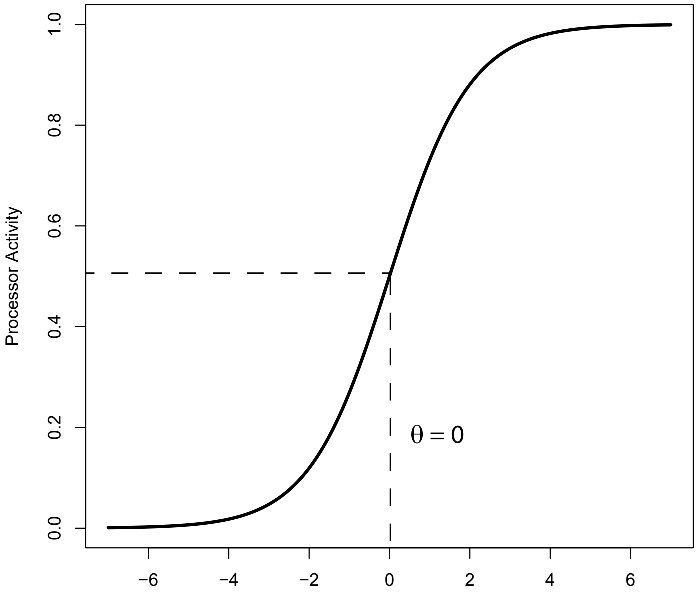


Figure 1-8. The logistic activation function is a continuous approximation of the Heaviside equation. In this example, when net input equals the bias (θ) of 0, then the function returns activity of 0.5.

$$a_j = f(net_j) = \frac{1}{1 + e^{(-net_j + \theta)}} \qquad (1\text{-}4)$$

We call a perceptron that uses a continuous activation function like the logistic equation a modern perceptron (Dawson, 2008). It can learn from a training set in a fashion that is very similar to that described in Section 1.5.3. However, with continuous activation functions one can modify the delta rule (Equation 1-3) by multiplying error by the first derivative of the activation function in an attempt to make learning more efficient (e.g., Dawson, 2004). Indeed, another reason for the popularity of the logistic equation is the simplicity of its first derivative, which is provided as *f'(netj)* in Equation 1-5:

$$f'(net_j) = a_j \cdot (1 - a_j) \qquad (1\text{-}5)$$

One can use the derivative from Equation 1-5 to convert the delta rule (Equation 1-3) into a different learning rule, the gradient descent rule for the modern perceptron (Dawson, 2008). This is accomplished by multiplying the error term in Equation 1-3 by the first derivative of the logistic. The gradient descent rule is provided in Equation 1-6:

$$\Delta w_{ij} = \eta \cdot a_i \cdot a_j \cdot (1 - a_j)(T_j - a_j) \qquad (1\text{-}6)$$

To complete our formal account of learning, one final detail needs to be included. The logistic equation includes the bias θ. The value of the bias can also be modified as the perceptron learns the training set. This is accomplished by treating the bias as if was the weight of a connection that was always receiving a signal of one. The desired change in the bias (Δθ*j*) is then computed using a version of Equation 1-6 that replaces *ai* with the value 1, as is shown in Equation 1-7. This change in bias is then added to the existing value of the bias. Equation 1-7 is used to update the bias using the gradient descent rule of Equation 1-6.

$$\Delta \theta_j = \eta \cdot a_j \cdot (1 - a_j)(T_j - a_j) \qquad (1\text{-}7)$$

1.5.5 Summary

Section 1.5 introduced the properties of a perceptron, related these properties to Brunswik's (1952) probabilistic functionalism and provided a formal account of the gradient descent learning rule for the modern perceptron. What is the relationship between training a modern perceptron and various topics in probability learning? The next section of Chapter 1 provides an empirical answer to this question by demonstrating that the modern perceptron learns to estimate probabilities when trained on a probabilistic discrimination task.

1.6 Modern Perceptrons and Probability Estimation

1.6.1 Certain and Uncertain Training

Earlier in this chapter it was pointed out that the history of artificial neural networks begins with research that aimed to provide a mathematical account of neuronal function (McCulloch & Pitts, 1943). The foundation of this research was the assumption that neurons could be described as asserting the truth or falsehood of logical operations. From this perspective, a neuronal response is digital, representing "on" or "off," "true" or "false." Similarly, this means that when a Rosenblatt (1962) perceptron was trained, the evaluation of its response could also only be digital: "right" or "wrong."

A similar perspective guides the training of most modern artificial neural networks. In particular, those that are trained with modern supervised learning rules are taught until they perform perfectly. That is, networks are typically trained until they generate the correct response to each stimulus in a training set (Dawson, 2004, 2005). This further implies that the training set is certain, in the sense that each of its patterns is associated with a unique and correct response.

However, the certainty of a training set disappears when we attempt to train a network on a probabilistic discrimination problem like the card-choice task. This is because, in some cases, a particular stimulus is rewarded, but in others, the identical stimulus is *not* rewarded. The mapping from stimulus to desired response is no longer unique, meaning that the training set is uncertain. How does a modern perceptron behave when it is trained on such an uncertain training set? It can no longer respond perfectly to every training pattern, because a single pattern can be associated with completely different responses (reward or no reward) at different times. To explore perceptron learning when the training set is uncertain, let us describe an example task and report the results of training on modern perceptron on this task.

1.6.2 The Weather Prediction Task

The weather prediction task is a modern probabilistic discrimination problem. Although it can be used to explore different strategies that might be employed during probability learning (Gluck et al., 2002), it is more typically used to explore implicit learning in a variety of clinical populations. It was originally developed to study probability learning in amnesic patients (Knowlton, Squire, & Gluck, 1994) and has been used to study other clinical populations, such as patients suffering from Huntington's disease (Knowlton et al., 1996), bulimia nervosa (Labouliere, Terranova, Steinglass, & Marsh, 2016), and schizophrenia (Keri et al., 2000).

The weather prediction task involves using different visual stimuli—cards—that can be used to depict combinations of different cues. When presented a card, the participant must decide whether the stimulus predicts rain. After making the prediction, the participant is provided feedback about whether he or she was correct. Each of the different cues is associated with a different probability of forecasting rain.

To provide a concrete example, let us consider one version of this task (Knowlton et al., 1994). In this example, four visual cues could be in a stimulus card. Each of the four cues is associated with a different likelihood of reward: Cue 1 signals a reward probability of 0.850, Cue 2 a probability of 0.620, Cue 3 a probability of 0.380, and Cue 4 a probability of 0.150. Each stimulus card includes one or more of these cues, and each is rewarded with the

Table 1-2. The properties of an example weather prediction task from Knowlton et al. (1994).

Cue 1	Cue 2	Cue 3	Cue4	*P*(Presented)	*P* (Rewarded)
0	0	0	0	0.000	0.000
0	0	0	1	0.140	0.150
0	0	1	0	0.084	0.380
0	0	1	1	0.087	0.100
0	1	0	0	0.084	0.620
0	1	0	1	0.064	0.180
0	1	1	0	0.047	0.500
0	1	1	1	0.041	0.210
1	0	0	0	0.140	0.850
1	0	0	1	0.058	0.500
1	0	1	0	0.064	0.820
1	0	1	1	0.032	0.430
1	1	0	0	0.087	0.900
1	1	0	1	0.032	0.570
1	1	1	0	0.041	0.790
1	1	1	1	0.000	0.500

Note. Each row in the table represents a possible stimulus. The first four columns indicate whether a cue is absent (0) or present (1) in the stimulus. The fifth column indicates the probability that a participant will see the stimulus. Note that this column indicates that participants never see a stimulus in which all four cues are either absent or present. The final column provides the likelihood that a particular stimulus will be rewarded when it is presented (i.e., will be said to predict rain if the participant makes this prediction).

Table 1-3. The characteristics of a perceptron's training set for the weather prediction task that was created from the information provided in Table 1-2.

Cue 1	Cue 2	Cue 3	Cue4	Number	Number
0	0	0	0	0	0
0	0	0	1	140	21
0	0	1	0	84	32
0	0	1	1	87	9
0	1	0	0	84	52
0	1	0	1	64	12
0	1	1	0	47	24
0	1	1	1	41	9
1	0	0	0	140	119
1	0	0	1	58	29
1	0	1	0	64	52
1	0	1	1	32	14
1	1	0	0	87	78
1	1	0	1	32	18

Note. The training set consisted of 1,000 stimuli. Each row in the table represents a possible stimulus. The first four columns indicate whether a cue is absent (0) or present (1) in the stimulus. The fifth column indicates the total number of instances of a particular stimulus to be found in the training set. The last column indicates the number of these patterns that were rewarded (i.e., were stimuli that the perceptron was trained to generate a response of 1). If a pattern was not rewarded, then when it was presented the perceptron was trained to generate a response of 0.

probability determined by combining the probabilities of reward signaled by its cues. The probability of reward for each card is presented in the last column of Table 1-2.

Knowlton et al. wanted the overall likelihood of reward across all stimuli to be 0.50. For this reason, not all of the cards are presented an equal number of times. First, a stimulus that possesses none of the cues, or one that possesses all of the cues, is never presented. Second, the remaining cards are presented with a different probability. The probability of each card being presented is also provided in Table 1-2.

The study proceeds by presenting a stimulus card to a participant, asking the participant to predict whether the card indicated sunshine or rain, and providing feedback about prediction accuracy. The major finding was that both control participants and amnesic patients had similar learning in the early trials but that later the control participants surpassed the amnesic patients in performance (Knowlton et al., 1994).

1.6.3 Weather Prediction by Perceptron

How does a modern perceptron behave when presented a problem derived from the weather prediction task described in Table 1-2? To answer this question, we designed a training set that was in close agreement with the properties of Table 1-2. The training set consisted of 1,001 different stimuli, where each stimulus was a configuration of four possible cues. If a cue was present, then the input unit that represented it was activated with a value of 1. Otherwise, the input unit as assigned an activity of 0.

As was the case in Table 1-2, different numbers of each stimulus were included in the training set. The number of presentations of each stimulus is provided in the fifth column of Table 1-3. For instance, this table indicates that the stimulus that included only Cue 4 was included 140 times in the training set. That is, the training set included 140 instances of this particular stimulus. In contrast, the stimulus that included only Cue 3 and Cue 4 was presented only 87 times in this training set.

The uncertainty of reward associated with each stimulus was accomplished by rewarding some instances of a stimulus and by failing to reward the other instances. For example, the stimulus that included only Cue 4 was presented 140 times but was rewarded on only 21 of these occasions. Thus its probability of reward was 21/140 = 0.15, which is identical to the probability of reward for this stimulus provided in Table 1-2. Similarly, the stimulus that included only Cues 3 and 4 was presented 87 times but was rewarded on only nine of these presentations. Thus its probability of reward was 9/87 = 0.1034, which is slightly higher than the value of 0.10 for this stimulus in Table 1-2.

The overall probability of reward for the training set was 0.5005, which is nearly identical to the desired value in the Knowlton et al. (1994) study. We accomplished this by rewarding 501 of the 1,001 stimuli in the training set. In other words, the reward structure of the training set that we created (Table 1-3) provides a close approximation of the reward probabilities associated with the various stimuli (Table 1-2).

With this training set in hand, we are in a position to use it to train a perceptron. This training, as well as all of the perceptron simulations reported later in this monograph, were conducted with the Rosenblatt program (Dawson, 2005). This program is available as freeware and can be downloaded from the following website: http://www.bcp.psych.ualberta.ca/~mike/Software/.

We proceeded to use the Rosenblatt program to train a perceptron on this training set as follows: First, the perceptron was a network with four input units (one to represent each of the four cues) and a single output unit that used the logistic activation function. Prior to training, the output unit's bias was initialized to 0, and each connection weight was randomly set to a value between −0.1 and 0.1. Second, the perceptron was trained with a learning rate (η) of 0.05, which is a standard value for this type of study (Dawson & Gupta, 2017). Training proceeded with the epoch-wise presentation of patterns. In a single epoch, each of the 1,001 patterns in the training set is presented once, but the order of presentation is randomized before the epoch begins. After each stimulus was presented, the weights and bias of the perceptron were modified using the gradient descent rule provided in Equations 1-6 and 1-7. Third, this training proceeded for 2,500 epochs. At the end of this training, the perceptron's response to each of the 14 types of stimuli was recorded.

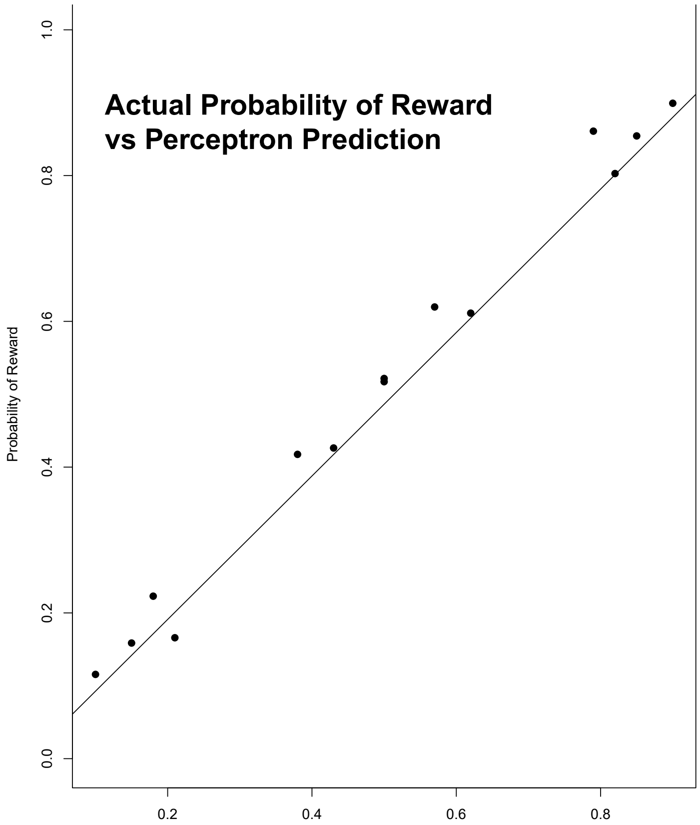


Figure 1-9. The performance of a perceptron on the weather prediction task. For each dot in the graph, the x-coordinate is the perceptron's response and the y-coordinate is the actual probability of reward.

Figure 1-9 provides the main result of this simulation study. Its *y*-axis plots the actual probability of reward of a stimulus card, where this probability is taken from Table 1-2. Its *x*-axis plots the response of the perceptron to the card. Each of the dots on the graph presents the data from one of the 14 stimuli in the experiment. As can be seen from Figure 1-9, the perceptron has learned to predict reward probabilities from cues because its output activity is nearly identical to the actual probability of reward for each stimulus. A regression equation that predicts the actual reward probability from the perceptron's response accounts for almost all of the variance in the data ($R^2 = 0.988$) and is statistically significant, $F(1, 13) = 1029.814$, $p < .0001$.

1.6.4 Perceptrons Predict Probability

Figure 1-9 provides some example evidence that demonstrates why we are interested in using modern perceptrons to explore probability learning. When confronted with an uncertain task, this type of perceptron learns the task's probability structure. That is, after learning it can convert a pattern of cues into an accurate prediction of reward probability. This fundamental property is required of a system that adapts to an uncertain environment via probability matching.

The results in Figure 1-9 provide additional evidence to be added to recent results that demonstrate probability estimation by perceptrons (Dawson et al., 2009; Dawson & Gupta, 2017). Perceptrons have also been shown to model many of the standard effects that have been discovered using the reorientation task (Dawson, Kelly, et al., 2010). In short, modern perceptrons appear to offer a viable medium for increasing our understanding of probability learning.

Furthermore, that perceptrons can estimate probabilities is of particular interest because these networks are very simple mechanisms. For example, the learning rules used to train perceptrons are formally identical to the Rescorla-Wagner model of animal learning (Dawson, 2008; Gluck & Bower, 1988; Sutton & Barto, 1981). Furthermore, some researchers have argued that artificial neural networks like the perceptron provide a plausible account of the brain mechanisms for associative learning (Shanks, 1995). In other words, if we can establish that perceptrons are plausible models of probability learning, they can provide accounts that link this type of learning to established principles of associative learning.

This last point is important, because one of the problems faced by a theory of probability learning concerns proposing possible mechanisms. Estes (1976) attempted to address this issue by exploring links between probability learning and cognitive theory. Similar issues have developed in the broader context of recent theories in cognitive science. For example, many modern theories that have arisen in the cognitive science of human reasoning and rationality explicitly appeal to Bayesian probability theory (Binmore, 2009; Chater & Oaksford, 1999; Oaksford & Chater, 1991, 1998, 2001, 2007). However, these theories do not provide mechanisms for performing Bayesian calculations, which critics claim is a fundamental flaw of Bayesian cognitive science (Bowers & Davis, 2012; Jones & Love, 2011). If simple mechanisms like perceptrons can respond with probabilities, then perhaps artificial neural networks can help solve this problem by serving as example Bayesian mechanisms.

Of course, the type of behavior illustrated in Figure 1-9 raises additional questions. How exactly does a modern perceptron generate probabilities, and how does it actually represent the kind of information that is required for probability estimation to occur? The final section of this chapter provides a brief review of the main points that it has covered, raises a set of interesting questions, and uses these questions to set the stage for the next few chapters that explore the relationship between perceptrons and probability theory. However, before setting the stage for these issues, let us first briefly consider why this monograph focuses on the perceptron and does not explore other possible architectures.

1.7 Why Study Perceptrons?

1.7.1 Perceptron Advantages

This monograph introduces a key theme as its focus: adapting to an uncertain environment. It also presents a basic architecture used to study this theme: the perceptron. However, the perceptron is a very old and simple type of artificial neural network (Rosenblatt, 1958, 1962) and has fallen out of favor in cognitive science, replaced by more modern and sophisticated architectures. Why do we explore the perceptron in this project and not some of these more popular and powerful architectures?

The first reason for studying the perceptron is that our primary concern is to understand how basic and well-understood psychological theories of associative learning can account for probability learning. To do so, we must use an architecture that can be related to such theories. The perceptron is one such architecture. Researchers have already established a formal relationship between learning rules for perceptrons and mathematical accounts of classical conditioning (Dawson, 2008; Gluck & Bower, 1988; Sutton & Barto, 1981). For example, Dawson demonstrated that one can literally translate the learning rule for a modern perceptron (Equation 1-6) into the equation for the Rescorla-Wagner model of associative learning (Rescorla & Wagner, 1972). This means that we can be confident that results that we obtain for the perceptron will be transferable to a particular psychological account of associative learning.

The second reason for studying the perceptron, and not more powerful architectures, is that perceptrons have inherent limitations. However, these limitations can explain some problems faced by humans when they adapt to uncertain environments.

Cognitive scientists have long been aware that there are limits to what perceptrons can learn because of their simple structure. Perceptrons fell out of favor in the late 1960s when it was established that they can solve only very basic categorization problems (Minsky & Papert, 1969), as discussed in more detail in Section 4. Cognitive scientists have long argued that the ability to model the complexities of human cognition requires much greater computational power, such as that provided by a universal Turing machine (Bever, Fodor, & Garrett, 1968; Chomsky, 1956, 1957). Perceptrons do not have such power; as a result, perceptrons became viewed as being inappropriate for cognitive modeling. In short, it is a standard belief that perceptrons are poor models of human cognition, because human cognitive abilities far exceed the limited power of these simple networks.

However, this standard belief needs to become more nuanced, because tasks that appear to be quite complex can often be simulated by simple models. For example, perceptrons have been shown to be fruitful models in diverse areas that include classical conditioning (Dawson, 2008), navigation (Dawson, Kelly, et al., 2010), and musical cognition (Dawson, 2018; Dawson & Zielinski, 2018). In other words, for some tasks the ability exhibited by a biological agent may be such that it can be plausibly modeled by a network that has no hidden units.

This observation is central to the current monograph. To foreshadow results that are presented in later chapters, formal results developed in Chapters 3 and 4 demonstrate that perceptrons can be described as naïve Bayesian mechanisms. This means that when they predict the probability of reward from presented cues, these predictions are blind to interactions between cues. Simulation results presented in Chapters 5 and 6 demonstrate that because of this limitation, perceptrons have difficulties estimating probabilities in variations of the card-choice task that involve interacting cues. Important to note, experimental results obtained from studying human participants in the card-choice task reveal performance that is very similar to the perceptrons. That is, humans and perceptrons both perform poorest in the conditions that cause the most difficulty for naïve Bayesian mechanisms. Thus, in this particular instance the limitations of perceptrons may be quite useful in accounting for problems encountered by human participants.

The third reason for studying perceptrons is that it is very straightforward to examine these networks mathematically, or to study the behavior of these systems in simulation experiments. This means that perceptrons provide an ideal medium for establishing a rich understanding of how basic associative mechanisms can be used to adapt to uncertain environments. Earlier studies of modern perceptrons have demonstrated that their continuous responses can be interpreted as estimates of reward probability signaled by cues and have related this property to other accounts of contingency learning (Dawson & Dupuis, 2012; Dawson et al., 2009; Dawson & Gupta, 2017; Dawson et al., 2008; Dawson et al., 2010; Dupuis & Dawson, 2013a). This monograph uses these results as a starting point for exploring conditions under which perceptrons can—and cannot—correctly estimate probabilities signaled by cues. It then presents the results of experiments of experiments that demonstrate similar capabilities—and limitations—in human participants. This creates a situation in which it is appropriate to use our formal and empirical understanding of perceptrons to account for aspects of human probability learning.

1.7.2 Why Not Study Multilayered Networks?

The proof that perceptrons had limited capabilities led to a dramatic decrease in using artificial neural networks as psychological models (Minsky & Papert, 1969; Papert, 1988). The renaissance of artificial neural networks in cognitive science began only after researchers discovered learning algorithms for training *multilayer perceptrons* (Rumelhart et al., 1986). These networks include at least one layer of intermediate processors, called *hidden units*, which transform the signals from input units before sending information on to the output units. Networks that include at least one layer of hidden units are far more powerful than perceptrons. They have the potential to solve any pattern classification problem (Lippmann, 1989) or to be universal function approximators (Hartman, Keeler, & Kowalski, 1989; Hornik, Stinchcombe, & White, 1989). In short, multilayered networks offer the same computational power as a universal Turing machine (Siegelmann, 1999; Siegelmann & Sontag, 1991). The ability to train multilayer perceptrons, combined with their potential computational power, resulted in a connectionist resurgence in cognitive science because these new networks seemed powerful enough to be plausible models of cognition (Dawson, 1998, 2013; Medler, 1998).

The computational power of multilayer perceptrons is an "in principle" property. In practice, however, learning rules like backpropagation of error (Rumelhart et al., 1986) have difficulty in training multilayer networks to perform extremely complex tasks. In recent years, this problem has been solved with the discovery of learning rules for *deep belief networks* (Bengio, 2009; Bengio, Courville, & Vincent, 2013; Hinton, 2007; Hinton, Osindero, & Teh, 2006; Larochelle, Mandel, Pascanu, & Bengio, 2012). These networks can include many different layers of hidden units. Deep learning has been the source of important new technologies in a variety of pattern recognition and data-mining domains; they too have the same computational power as a universal Turing machine (Sutskever & Hinton, 2008), but they offer the practical ability to harness such power. The many layers of hidden units in deep belief networks provide this power but also make the structure of these networks very complicated. This complexity currently limits their ability to contribute new theories to cognitive science because it makes their internal structure very difficult to understand (Dawson, 2018; Erhan, Courville, & Bengio, 2010; Liu et al., 2017; Montavon, Samek, & Muller, 2018).

Given the availability of modern architectures like multilayer perceptrons or deep belief networks, why do we not study them in the current monograph and instead focus on their much less powerful ancestor, the perceptron? One reason is their aforementioned complexity: If the internal structure of multilayered networks is difficult to comprehend, then they are not well suited to increase our understanding of probability learning. A second reason is that, although the psychological plausibility of the learning rule for perceptrons has been established (Sutton & Barto, 1981), this is not the case for the learning rules used to train multilayer perceptrons or deep belief networks.

A third reason, and perhaps the most important, is the computational power of these multilayered networks. The capabilities of these networks are such that if we trained one of them on a version of the card-choice task that used interacting cues to signal reward probabilities, then we would expect that the network would generate optimal responses. That is, the hidden units in the network would detect and exploit the information conveyed by interacting cues. However, as we pointed out in Section 1.7.1 and as we detail in Chapter 7, human performance on this task is not optimal, and is particularly poor in conditions in which cues interact. Thus, multilayer networks are too powerful to account for this performance. Fortunately, perceptrons are not. Thus, these simpler networks might provide better models of human performance in the card-choice task.

Of course, this is not to say that multilayered networks do not have a role in the study of probability learning. For instance, imagine a situation in which human probability learning exceeds the capability of perceptrons. In this case, the perceptron would not be an appropriate model, and the next obvious step would be to study this more advanced learning with multilayer networks. However, we see in Chapter 7 that this is not the case in the rich, uncertain environment provided by the card-choice task. Thus, it is appropriate to model this task with simpler networks, because performance in the card-choice task is such that the use of multilayer networks can be postponed.

1.7.3 What Is the Relationship to Bayesian Networks?

In Section 1.7.1, it was noted that one reason the current monograph focuses on perceptrons is that some of our previous research indicates that the outputs of these simple networks can be interpreted as representing conditional probabilities. That is, a perceptron's response represents its estimate of the probability of being rewarded given the set of cues that it is being presented. One of our current goals is to understand the kind of probability that the perceptron is computing and to understand how this probability is computed. Formal results to be presented in Chapters 3 and 4 indicate that the probability computed is defined by the naïve Bayesian equation (Equation 4-4 in Chapter 4). As a result, we later describe perceptrons as being naïve Bayesian mechanisms.

Given that perceptrons are networks, and that they can be described as Bayesian, what is their relationship to the large literature on Bayesian networks? In general, there is little relationship between the two. Bayesian networks are graphical structures that represent the causal relationships between variables, including causal relationships between chains of variables (Koski & Noble, 2009), and can also be used as statistical models of human causal reasoning (Glymour, 2001). In this graph, nodes represent variables, and connections between nodes (called edges) indicate a causal relationship from one variable to another. The structure of a Bayesian network can be used to infer a conditional probability of an event occurring given the presence of other events that are related (either directly or indirectly) to the event of interest. This inference is simplified by the absence of edges; such absences indicate that some predictors are independent of others. For a larger causal domain, determining the appropriate sets of edges between variables can be a challenge, and various techniques from statistical theory or machine learning can be used to solve this problem (Heckerman, Geiger, & Chickering, 1995; Koski & Noble, 2009). However, these techniques are quite different from the simple associative learning rule that was introduced in this chapter. In short, Bayesian networks are statistical models used to conduct probabilistic inference, using methods and assumptions that are quite different from those underlying the perceptron.

There is also a related literature on using Bayesian statistical techniques to train artificial neural networks that are much more strongly related to the networks explored in this monograph (MacKay, 1992, 1995). These systems are called Bayesian neural networks (Neal, 2012). The "Bayesian" in the name for these neural networks refers not to their behavior but rather to statistical techniques that provide alternatives to traditional learning rules for identifying a set of parameters (e.g., network weights) for solving a particular problem. Bayesian neural networks are not typically used as psychological models; for instance, Neal (2012) presented Bayesian methods for training multilayer networks to be used in engineering applications. In contrast, the perceptrons explored in the current monograph are much simpler, and they are of particular interest because of the known relationship between their learning rule and psychological models of associative learning.

1.7.4 Relationship to Reinforcement Learning

Section 1.7.3 argued that both Bayesian networks and Bayesian neural networks are not examined in the current monograph because the techniques for creating or adapting these systems are derived from statistics and not from psychological theory. Another approach in the literature, reinforcement learning, is much more strongly inspired by psychological considerations (Kaelbling et al., 1996; Sutton & Barto, 1998). How do the simple networks explored in later chapters relate to the reinforcement learning literature?

Reinforcement learning is concerned with modeling how an agent's actions on an environment can maximize cumulative reward. Typically, this involves balancing exploitation of current knowledge (choosing actions that are already known to accumulate reward) and exploration (choosing new actions to learn the degree to which they are rewarded). A reinforcement learning system includes a policy that maps perceived states of the environment onto actions that can be taken. It also includes a reward function that maps environmental states, or these states combined with actions, onto a number that reflects desirability or reward. It also includes a value function that specifies rewards to the agent in the long term; the reward function only specifies immediate returns. Finally, a reinforcement learning system includes a model of the environment that is used to predict the next environmental state and the next reward. Depending upon the sophistication of these various components, reinforcement learning systems span "the spectrum from low-level, trial-and-error learning to high-level, deliberative planning" (Sutton & Barto, 1998, p. 9). Neural networks can play a role in a reinforcement learning system, although this is not a necessary characteristic. For example, a multilayer perceptron could be used to provide a policy function or a value function.

Reinforcement learning is typically presented as an alternative to the supervised learning techniques that are used to train typical multilayer networks, including the perceptron. This is because reinforcement learning does not match each environmental state with a detailed representation of ideal responses. Instead, reinforcement learners receive only a single signal: the degree to which their action on the environment is rewarded.

As the perceptrons studied in the current monograph use a single output unit, and the training of this unit involves signaling whether the network receives a reward, they can be viewed as very simple reinforcement learning systems. In Chapter 6, which presents operant learning rules for perceptrons, the relationship between the current work and reinforcement learning becomes slightly stronger. This is because the operant learning rules involve perceptrons deciding upon a basic action: choosing to learn about a particular presented pattern of cues, where this decision is made using the network's estimation of the probability of reward signaled by these cues. However, the relationship between perceptron learning in the current monograph and reinforcement learning is weak. To the extent that we are comfortable in viewing our perceptrons as reinforcement learners, we are really investigating the kind of adaptation that is possible in an extremely simple reinforcement learning systems. If we felt that more sophisticated models than perceptrons were required to explore probability learning, then reinforcement learning, such as multilayer networks briefly discussed in Section 1.7.2, would offer a plausible and more powerful alternative.

1.7.5 Why Not Use Rescorla–Wagner?

Why does the current monograph not use the Rescorla–Wagner rule if its goal is to explore the relationship between associative learning and adapting to uncertain environments? The primary answer to this question is that to model probability learning, we need to generate responses (e.g., estimates of the probability of reward) from the associative strengths that are updated by learning. Surprisingly, the Rescorla–Wagner model does not describe how to convert associative strengths into responses; it is not a model of behavior (R. R. Miller, 2006). Rescorla and Wagner (1972) were quite explicit about this aspect of their model: "Independent assumptions will necessarily have to be made about the mapping of associative strengths into responding in any particular situation" (p. 75). Rescorla and Wagner resisted the temptation to make any such assumptions because of their belief that these assumptions would be related to a number of different "performance variables" that would vary from one experimental situation to another.

In contrast, although the training of perceptrons can be formally related to the Rescorla–Wagner rule, the activation function of the perceptron also provides an explicit account of how associative strengths or connection weights are converted into behavior. This means that the perceptron offers certain advantages over using the Rescorla–Wagner rule itself. The behavioral differences between perceptron learning and the Rescorla–Wagner rule have been examined in detail (in particular, see Dawson, 2008, Chapter 8).

1.8 Summary and Implications

1.8.1 Summary

Chapter 1 has introduced both the main topics of this monograph and the basic components used to explore these topics. It began by noting that our relationship with the world is confronted with epistemic, somatic, and ontic uncertainty. As a result, we can never predict the future with complete accuracy. However, we thrive in this uncertain environment, which suggests that we somehow can adapt to it. The core issue explored in this work is how agents adapt to uncertainty.

This is a very large topic, so one purpose of the chapter was to introduce a task that focused on a smaller, but still interesting, component: probabilistic discrimination. In probabilistic discrimination, cues or signals from the environment provide information about future events, but this information is not perfectly accurate. The chapter introduced an example task that is important throughout the monograph to illustrate this type of environment. In this card-choice task, agents explore an environment that contains different playing cards. When a card is chosen, it provides a reward with a set probability, and the symbols on the card serve as signals about the likelihood of a reward. This task is straightforward but can be related to diverse topics, including experiments on animal reorientation and studies of multiarmed bandit strategies.

The card-choice task is also closely related to an old and well-established literature on probability learning. One of the core findings in this psychological literature is that humans exhibit probability matching: The number of times an action is performed is correlated with the probability that the action will be rewarded. We linked performance in an example card-choice task to this phenomenon. We discovered that as participants explore the different cards, they are more likely to receive rewards; one reason for this is that their card choices exhibit probability matching.

What causes probability matching? We saw that older theories of probability matching provide descriptions of behavior but failed to explain this behavior by proposing putative mechanisms. To solve this problem, we hypothesized that probability matching related to associative learning of the sort described by a simple artificial neural network, the perceptron. We described the basic properties of a modern perceptron that uses the logistic equation as its activation function and provided a gradient descent learning rule for its training. A simple simulation study that trained this type of network on the weather prediction task revealed a key property. At the end of training, the response of a perceptron was a highly accurate estimate of the reward probability associated with a particular pattern of cues.

1.8.2 Implications

Chapter 1 frames the study of adapting to uncertainty quite specifically as the study of how artificial neural networks learn probabilistic discrimination tasks. The material captured in this frame is broad enough to raise many interesting questions. Fortunately, framing the topic in this way provides many tools that permit us to answer these questions.

To begin, we have already seen that the modern perceptron can learn the probabilities of reward signaled by cues in a particular probabilistic discrimination task; this is consistent with other results in the literature (Dawson et al., 2009; Dawson & Gupta, 2017). Given that the modern perceptron can be easily used to conduct simulation experiments, and can be analyzed mathematically, we are in a position to answer very specific technical questions about how this type of network adapts to uncertainty. In other words, we can use the methods of synthetic psychology (Braitenberg, 1984; Dawson, 2004, 2013; Dawson, Dupuis, & Wilson, 2010) to study probability learning in perceptrons. What kinds of probability learning do perceptrons exhibit? What limitations do perceptrons have when they learn probabilities? One way to explore these questions is to train perceptrons on probability learning problems, such as variations of the card-choice task, and then to observe their strengths and weaknesses.

The weaknesses of perceptrons are of particular interest. There is a long history of perceptrons being abandoned or ignored because their simple structure places limits on what they can learn (Minsky & Papert, 1969; Papert, 1988). We see in Chapter 2 that these limitations are typically described logically in terms of a property called *linear separability*. However, this sort of limitation may not be the most appropriate to consider when we are concerned with perceptrons' probabilistic behavior, and not their logical judgments (Dawson & Gupta, 2017). Furthermore, in some cases the limitations of perceptrons are appropriate, because similar limitations also govern the behavior of biological agents (Dawson, 2008; Dawson, Kelly, et al., 2010). Simulation studies are required to determine the limits of perceptrons, and experiments with human participants are required to relate perceptron responses to human probability learning.

A second limitation of the perceptron introduced in this chapter is more important. The simulation results show that perceptrons can estimate probabilities in some situations. However, the perceptron discussed in Sections 1.5 and 1.6 cannot use this estimated probability to make a choice, because this action is not part of the perceptron's repertoire. A different kind of network, such as the operant perceptron (Dawson et al., 2009), is required to convert probabilities into choices.

The next few chapters ignore this important limitation to use the modern perceptron to obtain a more rigorous understanding of probability learning. Chapter 2 begins this process by exploring the role of cues as signals that provide information. This leads to a discussion of three interrelated topics: information theory, probability theory, and cybernetics. These topics are brought into focus with the modern perceptron. First, it is argued that the role of information is to reduce uncertainty, but in the context of the perceptron, this uncertainty must be of a particular type, called subjective probability. Second, it is argued that the learning rule that was described in Section 1.5 is a prototypical example of negative feedback. Together these points link probability learning in perceptrons to cybernetics and Bayesian probability theory.

Chapter 3 uses computational analyses to explain exactly how modern perceptrons generate probabilities and details how the connection weights and bias of a network are related to probability theory. This is accomplished by demonstrating the Bayesian behavior of a very simple perceptron and then by proving that this behavior is a consequence of using the logistic activation function. This analysis shows how one can think of perceptrons in terms of other concepts in probability theory: odds, odds ratios, and contingency.

Chapter 4 builds upon the results of Chapter 3 to explore perceptrons in more complex situations. It begins by translating the traditional logical limits of perceptrons into different kinds of limits that are more appropriate when the probabilistic responses of perceptrons are of interest. It proves that perceptrons that consider signals from multiple simultaneous cues are naïve Bayesian mechanisms and that a key factor that limits their abilities is the notion of conditional dependence in probability theory. It uses this limitation to inspire five variations of the card-choice task. Probability estimations in these different simulations clearly demonstrate that some versions of this task are more difficult for perceptrons than are others. Later we ask whether this pattern of results is also evident in human probability learning.

By the end of Chapter 4, we will have acquired a very sophisticated understanding of the probabilistic behavior of the modern perceptron. We will then be in a position to extend this model into a network that explores an environment by making choices and by comparing the performance of networks to the performance of human participants. However, these aims are introduced later in the monograph. Let us start by exploring perceptrons in the context of information theory, probability theory, and cybernetics.

Chapter 2: Information, Probability, and Negative Feedback

Chapter 1 introduced the theme of this monograph—adapting to an uncertain world— as well as a basic system for exploring this theme, the modern perceptron. Chapter 2 explores how perceptrons learn in more detail and relate this learning to three interrelated topics: mathematical information theory, physical versus subjective probability, and negative feedback. The core message of Chapter 2 is that perceptrons generate subjective probabilities that predict physical probabilities of events in the world. Perceptrons then receive information—in the form of negative feedback—that is used to change network structure to improve the accuracy of these predictions. This process accounts for the probability estimating behavior of perceptrons and is illustrated with an example simulation. However, this process also means that perceptrons alter their structure to achieve a dynamic equilibrium with the environment. When this is achieved, the perceptron is constantly altering its internal structure, but its overall behavior is unchanged—it matches probabilities. A second simulation is used to demonstrate that these processes are active; in this second example the perceptron's environment suddenly changes—but it quickly alters its structure to achieve a different dynamic equilibrium.

2.1 From Association to Information

2.1.1 Association by Contiguity

Classical conditioning, as originally conceived (Pavlov, 1927), is a form of learning that is based on the associationist law of contiguity (Warren, 1921). According to the law of contiguity, when two ideas (e.g., A and B) occur together, the association between them is strengthened. As their association becomes stronger with repeated contiguity, the occurrence of A on its own will cause the occurrence of B. The law of contiguity was used by the associationists to account for the sequential production of thoughts. Its success in predicting the results of a variety of learning laid the foundation for psychological behaviorism, which dominated North American psychology in the first half of the 20th century.

However, as psychology's study of learning became more sophisticated in the second half of the 20th century, its findings begin to challenge the explanatory power of the law of contiguity. For instance, landmark studies discovered the phenomenon of blocking, which demonstrated that in certain conditions contiguity fails to produce learning (Kamin, 1968, 1969).

Blocking involves manipulating the contiguity of a conditioned stimulus (CS) with an unconditioned stimulus (UCS) that elicits a response. In classical conditioning, the repeated pairing of the CS and the UCS produces an association between the CS and the response so that presenting the CS elicits the response. To demonstrate blocking, two conditioned stimuli (CS_A and CS_B) are presented as a compound stimulus in a classical conditioning paradigm. A control group of animals is trained using the compound stimulus only; at the end of this training, both CS_A and CS_B elicit the response. In contrast, an experimental is first trained on CS_A alone. After this training, it is then trained with the compound stimulus in the same manner as the control group. The central finding is that experimental group shows less of a response to CS_B than does the control group. The pretraining with CS_A blocked later learning that could have occurred with CS_B.

The blocking phenomenon presented problems for traditional theories of learning that relied on the law of contiguity. This is because in the second phase of training for the experimental group, CS_A and CS_B were contiguous with the unconditioned stimulus that produced the response. Theories based on contiguity predicted that an association between CS_B and the response should be created. That this association was blocked by prior learning clearly indicated that something more than contiguity was involved in classical conditioning.

2.1.2 Cognitive Conditioning

Mathematical information theory was making an impact on psychology at roughly the same time that learning theorists were confronting phenomena like blocking (Attneave, 1959; Berlyne, 1960; G. A. Miller & Frick, 1949; Quastler, 1955). Thus, it is no surprise that learning theorists began to explore these ideas at this time. As we saw in Chapter 1, one issue that arose in the probability learning literature was whether the role of cues or rewards was to provide information to a learning agent (Bitterman, 1956; Hillix & Marx, 1960). This issue was beginning to be considered across all of learning theory at this time (Egger & Miller, 1962, 1963; Mowrer, 1960). For instance, Mowrer considered that the time was ripe for learning theory to integrate ideas from information theory and cybernetics. He raised the possibility that, with learning, the presence of a conditioned stimulus was a signal that produced "hope," that is, hope that a reward would soon appear. Egger and Miller argued that learning occurred only if the conditioned stimulus provided new information that was not redundant with the information provided by other cues.

The ideas that learning occurs when conditioned stimuli are not redundant, and that such learning makes conditioned stimuli signals of hope, are both tied up with the notion that cues produce expectations in the learner. Kamin believed that a similar idea explained blocking (Siegel & Allan, 1996). However, expectation did not have a strong influence on mathematical learning theory until the early 1970s (Rescorla & Wagner, 1972). The Rescorla–Wagner model of learning formalized the notion that learning requires surprise. That is, cues provide information that establishes expectations in learning agents. If these expectations are confirmed, then no learning occurs. However, if these expectations are not confirmed—if the learner is surprised—then associations are updated.

To be more precise, Rescorla and Wagner (1972) proposed that the amount of change in an association at any given time was a function of the difference between the current associative strength—V(t)—and the maximum possible associative strength (l), scaled by two constants that reflected learning rate and stimulus salience (α, β). Informally, this difference reflected the predictive power of the CS. If this difference was large, then the predictive power of the CS was poor, and the strength of the association needed to be modified. If this difference was small, then the predictive power of the CS was excellent, and little modification of the association was required. Formally, Rescorla and Wagner (1972) defined the change in associative strength that was required at any time t as

$$\Delta V_{(t)} = \alpha\beta\left(\lambda - V_{(t)}\right) \qquad (2\text{-}1)$$

With the change in associative strength defined using Equation 2-1, Rescorla and Wagner defined an iterative equation for the change in associative strength changed over a sequence of trials:

$$V_{(t+1)} = V_{(t)} + \Delta V_{(t)} \qquad (2\text{-}2)$$

Equations 2-1 and 2-2 describe a condition in which only a single CS can be present. Rescorla and Wagner (1972) extended these equations to easily deal with more complicated situations. For example, imagine that three conditioned stimuli, CS_A, CS_B, and $CS_{C,}$ are involved. The associative strength for each of these stimuli can be represented as V_A, V_B, and V_C, respectively. The overall associative strength (ΣV) for this example is then defined as

$$\Sigma V = V_A + V_B + V_C \qquad (2\text{-}3)$$

After defining ΣV, Rescorla and Wagner (1972) used it to define the change in individual associative strengths that would occur at some time *t* during learning. This extends Equation 2-1. For instance, the change in associative strength for CS_A at time *t* is defined as

$$\Delta V_{A(t)} = \alpha_A\beta\left(\lambda - \Sigma V_{(t)}\right) \qquad (2\text{-}4)$$

Note that this equation is sensitive to the fact that each CS might have a different salience; α_A is the salience for CS_A alone. Similar equations are used to calculate the change in associative strengths for the other conditioned stimuli that were present, with each equation using a different salience constant.

The Rescorla–Wagner model has been extremely influential (Miller, Barnet, & Grahame, 1995; Siegel & Allan, 1996; Walkenbach & Haddad, 1980). One reason for this is that this model provided a formal framework that predicted phenomena like blocking. In the experimental condition in a blocking study, there is no new learning involving CS_B because it does not provide any new information (or any surprises) beyond that which has already been provided by CS_A. The model also made many other counterintuitive predictions that have been confirmed experimentally.

We have two other important reasons to be interested in the Rescorla–Wagner model in this monograph. The first is the tacit assumption upon which it is based: that cues set up expectations, and that the amount of learning that occurs is proportional to the degree that these expectations are not met. This tacit assumption suggests that there should be a link between Rescorla–Wagner learning and information theory. This is particularly relevant because mathematical information theory is defined in terms of uncertainty and probability, as is detailed shortly in this chapter.

The second reason for our interest in the Rescorla–Wagner model is its strong relationship to learning rules used to train artificial neural networks. It has long been known that there is a formal relationship between these two types of learning rules when it is assumed that the artificial neural network employs a simple, linear activation function (Gluck & Bower, 1988; Sutton & Barto, 1981). More recently, Dawson (2008) proved that the gradient descent learning rule for the modern perceptron (Equation 1-6) can be translated into the Rescorla–Wagner rule. In other words, the manner in which we train modern perceptrons is formally equivalent to a hugely influential mathematical model of animal learning.

This relationship between animal learning and perceptron learning is important because Chapter 1 already demonstrated that perceptron responses are interpretable as probability estimates. Probabilities also provide the foundation for mathematical information theory, as we see in Section 2.2. We will soon be in a position to use the perceptron to provide a link between information theory and animal learning, particular in the context of learning that involves expectations of reward in an uncertain world.

Furthermore, exploring learning that involves expectations of reward in an uncertain world will lead us directly into two additional topics. First, we see that the idea of probability is itself complex. As we develop an account of learning that affects probability judgments or expectations, we are forced to accept that we are dealing with a controversial topic known as *subjective probability* (Jeffrey, 2004; Kyburg & Smokler, 1964). Second, as we explore learning to make better predictions about uncertain events (i.e., learning to match probabilities), we are able to introduce an important notion from cybernetics called *negative feedback* (Ashby, 1956; Wiener, 1948, 1950). Cybernetics is a field closely linked to information theory and is interested in how information can be used to adapt or control actions on the world. Our goal is to use artificial neural networks as a lingua franca that will permit us to establish firm links between probability learning, animal learning, information theory, and cybernetics. In particular, these links are provided by an explicit understanding of the error-based learning represented at the bottom of Figure 1-6 in terms of negative feedback.

To begin, let us take as our working hypothesis that the cues we process during learning provide us information that establishes expectations about what is about to occur. To understand the import of this working hypothesis, we need to define what "information" is. Section 2.2 does this; Section 2.3 builds on this definition to describe how probability theory is linked to information theory.

2.2 Twenty Questions

2.2.1 Decreasing Uncertainty

We learn about our world by acquiring information. What is information? In the most general sense, "a statement or observation is informative if it tells us something we did not already know" (Attneave, 1959, p. 1). Similar accounts are provided by others (Cherry, 1957; Garner, 1975; MacKay, 2003).

The mathematical theory of information provides a more technical definition (Shannon & Weaver, 1949; Wiener, 1948). According to it, to acquire information is to reduce uncertainty. As the primary topic of this monograph is with how humans and other agents adapt to an uncertain world, let us begin by exploring some ideas about uncertainty that arise in the context of information theory.

The parlor game of Twenty Questions can be used to illustrate some of these key ideas (e.g., Attneave, 1959). For instance, Attneave described a version of this game in which one player chooses a particular square on a chessboard and another player tries to identify this square by asking no more than 20 questions. These questions can be answered only with a "yes" or a "no." In one particular instance of this game, Player 2 selects the square g5 (see Figure 2-1) as the target to be identified by Player 1. The following conversation provides an example of how this particular game could proceed:

> Player 1: "Is your square black?"
>
> Player 2: "Yes."
>
> Player 1: "Does your square fall in any row from 1 through 4?"
>
> Player 2: "No."
>
> Player 1: "Does your square have no more than one black square to its right in your square's row?"
>
> Player 2: "Yes."
>
> Player 1: "Does your square fall in rows 5 or 6?"
>
> Player 2: "Yes."
>
> Player 1: "Does your square fall in columns e or f?"
>
> Player 2: "No."
>
> Player 1: "Is your square g5?"
>
> Player 2: "Yes."

With that final answer, the game ends. Player 1 has correctly identified Player 2's square by asking only six questions, winning the game.

Player 1 was able to win this game because the answer to each question provided information. In particular, before the game ends, Player 1 is uncertain about which square on the board is the target. The answer to each question cut this uncertainty in half. That is, if before asking the question Player 1 believed that any one of x different squares was the target, then after hearing the answer she believed that the target now belonged to a set of only $x/2$ possibilities. In other words, the answer decreased Player 1's uncertainty about the target's location.

To illustrate this, let us consider the game's conversation in more detail. At the beginning, there is equal likelihood that the target square could be any one of the

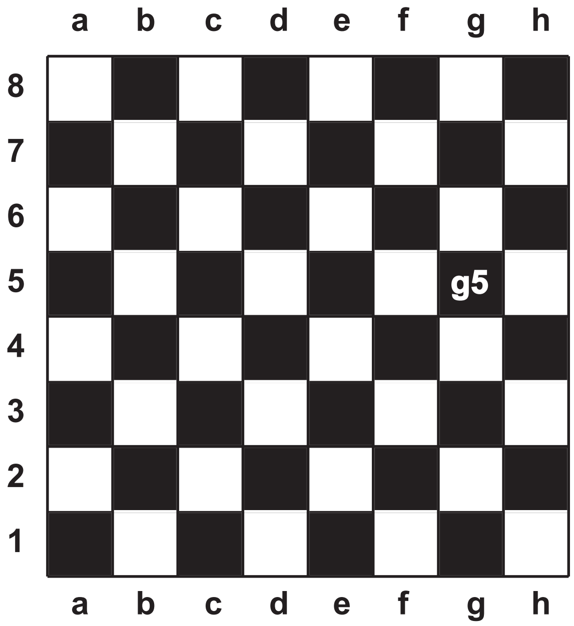


Figure 2-1. A chessboard for playing Twenty Questions, with each row labeled by a number and each column labeled by a letter. In this example, the square g5 has been selected as the target.

64 locations on the chessboard. Player 1's first question about the color of the square halves this uncertainty. If the answer is yes, then the target square must be among the 32 squares that are black; if the answer is no, then the target square must be among the other 32 squares that are white. The remaining questions continue to halve the number of possible locations of the target square. The answer to the second question indicates that the target square must be one of the 16 black squares in the upper half of the chessboard. The answer to the third question means that the target square must be one of the eight black squares in the upper right quadrant of the board. The answer to the fourth question specifies that the target square is one of the four black squares in the lower half of this quadrant. The answer to the fifth question signals that the target square is one of the two black squares within columns g and h and within rows 5 and 6. The answer to the final question names the target square by selecting it from the final two candidates. If Player 2 answered no to this final question, Player 1 would know that the target square must be the other candidate, the black square at h6.

2.2.2 Bits of Information

Mathematical communication theory measures the amount of information in units of *binary digits*, which are typically called *bits*. One bit of information is the amount required to choose between two equally likely possibilities. That is, a bit is the amount of information that halves our uncertainty. For instance, in tossing a fair coin, there are only two possibilities, heads or tails. When we observe that the tossed coin shows heads, we receive one bit of information, because we now know which of two possibilities actually occurred. Similarly, the answer to each of the questions asked by Player 1 in the game above provides one bit of information. This is because each answer chooses between two equally likely sets of target square positions (black vs. white squares, black squares in the upper half of the board vs. black squares in the lower half, etc.).

Because of its digital nature, the conversation provided earlier can be represented as a string of binary digits, with each digit representing the answer to a question, and the order of the digits preserving the order of the questions: 101011. This string of digits is literally the binary address of the target square on the chessboard. If the same set of questions is always asked, but the answers are different (because a different target square is selected), then the different string of digits will represent the binary address of the different target (Attneave, 1959). As there are six questions being asked, and each question leads to one of two responses, this means that there are 2^6 different binary addresses that are possible. The value 2^6 equals 64, which means that it is possible for a set of six binary questions to uniquely identify any possible target square on the board.

Alternatively, one could ask what the smallest number of questions is in order to eliminate the uncertainty about the target location. Information theorists answer this kind of question by taking the base 2 logarithm of the total number of possibilities, because this particular logarithm converts uncertainty into bits. For instance, in our version of the Twenty Questions game $log_2(64) = 6$, which means that a minimum of six bits of information, or six answers to binary questions, are required to locate the target square. However, for this to be true, the answer to each question must be maximally informative—each answer must provide one bit of information. If this is not the case, then more questions are required, as we see next.

From the perspective of information theory, a key characteristic of our example game is that it is binary—there are only two possible answers to each question. Literally, this means that each answer can be represented with a binary digit: 0 for no and 1 for yes. This property of the game leads to measuring information in bits and explains why base 2 logarithms are used to convert uncertainty into bits.

Furthermore, the two possible states of a binary digit have other useful interpretations. They can be interpreted logically as "false" or "true," as is typical in mathematical logic (Boole, 1854/2003). They can also be interpreted physically as the states of electric switches (i.e., open vs.

closed). Indeed, one of the most significant discoveries that led to rapid advances in communication and computer technology was Claude Shannon's realization that the states of electric switches can be translated into the truth functions of Boole's logic and that one could therefore use Boole's logical operations to test, to design, and to simplify electric circuits (Shannon, 1938).

From the perspective of the current monograph, another key characteristic of our example game is that the answers given by Player 2 to each of Player 1's questions are both reliable and valid. Reliable answers are those that are replicable. Player 2's answers are reliable if she provides exactly the same answer to the same question if it is asked repeatedly. Her answers are valid if they are truthful—for instance, if she only answers yes to the first question when the target square really is black. Player 1 would have a much more difficult time playing the game if Player 2's answers lacked reliability and validity. For instance, imagine a situation in which Player 2 only answered yes to the first question 80% of the time when the target square was indeed black. In information theory, this situation would be dealt with by considering the effect of noise on a communication channel. The primary concern of the current monograph is to consider a complementary approach to this problem by exploring methods that people might adopt to deal with a world that is not completely reliable or valid.

As was hinted earlier, Player 1 could have asked a different set of digital questions that are not as efficient, in the sense that she would have to ask many more of these questions to identify the target square. For example, she could simply ask about each of the 64 squares in order: "Is your square a1?" "Is your square b1?" and so on, until the target square is reached. Sometimes this strategy would lead to the game being over quickly, whereas on other occasions it could take a full set of 64 questions to find the target. Assume that Player 2 chooses the location of the target square at random and that each square of the chessboard is therefore equally likely to be the target. Over a large number of games, we would expect that this second strategy for Player 1 requires asking on average 32 questions to find the target.

Both of the approaches to the game of Twenty Questions are binary, in the sense that there are only two possible answers to any question in either strategy. Why then is one strategy more efficient than another strategy? Although either approach uses questions that can be either true or false, this binary state does not mean that the answers to either set of questions provide the same amount of information. The first approach to the game uses questions that provide more information than is provided by the answers to the questions used in the second approach. The mathematical theory of information provides a precise method for measuring the amount of information provided by each answer, as is discussed in the next section.

2.3 Measuring Information

2.3.1 Entropy and Probability

Events or observations that convey information do so by reducing uncertainty. It should therefore not be surprising to find that a measure of uncertainty, probability, is central to defining the measurement of information. This section briefly reviews the standard equations for measuring information. In later sections, we use these equations to raise the possibility that alternative definitions of information might be more appropriate for theories concerned with how agents adapt to an uncertain world.

To begin, let us consider a simple example in which we wish to measure the amount of information conveyed by observing the result of a single toss of a fair coin. By definition, there is an equal chance of this coin showing either heads or tails. That is, $P(H) = P(T) = 0.5$, where $P(H)$ represents the probability of observing heads and $P(T)$ is the probability of observing tails. As there are only two possible events, and each is equally likely, there is one chance in two that the event will occur, so the probability of either is ½ = 0.5.

A slightly different way to consider the coin-tossing example is to map it into the digital scenario that we saw earlier in Twenty Questions. To do so, we must determine the number of bits—the number of answers to digital questions—that are required to generate a unique address or label for each of the possible alternatives that we face. Let m stand for the number of possible alternatives, and let H stand for the required number of bits. From this it follows that $m = 2^H$. This expression can be manipulated to express H in terms of m, as is presented below in Equation 2-5. In Equation 2-5, H represents the amount of information in bits. So, in our coin toss example where $m = 2$, $H = log_2 2 = 1$. This shows that observing the result of a coin toss conveys exactly one bit of information:

$$H = log_2 m \tag{2-5}$$

Equation 2-5 can be applied to situations in which more than two possibilities are equally likely. For instance, consider rolling a die, where if the die is fair there are six equally likely possibilities. In this case, $m = 6$, so $H = log_2 6 = 2.56496$ bits. This indicates that rolling a die provides more information than does tossing a coin. The reason for this is that rolling a die reduces our uncertainty from six possibilities to one, whereas tossing a coin reduces uncertainty only from two possibilities to one. Rolling the die provides more information because there is a greater decrease in uncertainty that is captured by the equation for H.

Equation 2-5 can be used to provide some insight into the information theory structure of the game of Twenty Questions. First, consider the overall goal of that game, which is identifying one out of 64 possible locations on a chessboard. From this perspective, $m = 64$, so $H = log_2 64 = 6$. This shows again that six bits of information are required to find the target square, so it is not surprising to see that Player 1 could win the game by asking only six questions.

We could express the equation for the amount of information in an alternative form, taking advantage of the fact that $H = log_2 m = -log_2(1/m)$. The point of doing this is that when there are m equally likely alternatives, then the probability of any one of these alternatives being selected is $1/m$. In other words, we can state that H in terms of p, where p is the probability of one of the events occurring. This is done in Equation 2-6, which provides an explicit link between amount of information and probability:

$$H = -log_2(\frac{1}{m}) = -log_2 p \qquad (2\text{-}6)$$

The equations that we have been discussing are based on the assumption that each of the possible events is equally likely to be observed. When this assumption is false, Equation 2-6 must be modified to become more general. For example, let us consider the situation in which we toss a coin, but this coin is biased so that $P(H) = 0.8$ and $P(T) = 0.2$. In this situation, there are two possible events, but they occur with different likelihoods so that heads is more likely than tails. Information theory deals with this situation by using the probability of an event occurring to weight the amount of information that the event conveys. This leads to the following equation for the amount of information (Shannon & Weaver, 1949), which is called the Shannon–Weaver measure of *entropy*:

$$H = -\sum p_i log_2 p_i \qquad (2\text{-}7)$$

In Equation 2-7, there are i different alternatives, and p_i provides the probability that each alternative occurs. The equation computes the information associated with the alternative—$log_2(p_i)$—and then weights this by the probability that the alternative occurs (p_i). Thus, the Shannon–Weaver equation does not provide a sum of the total information but instead provides the average amount of information transmitted when an alternative is selected. For example, consider the biased coin example, where the probability of heads is 0.8. Using the Shannon–Weaver measure for this situation, $H = -(0.8 \times log_2(0.8) + 0.2 \times log_2(0.2)) = -(0.8 \times -0.32193 + 0.2 \times -2.32193) = 0.72193$. In other words, although a toss of a fair coin provides one bit of information, the toss of this biased coin provides less than three fourths of a bit. The reason for this discrepancy in information is the fact that there is less uncertainty for the biased coin, because heads are much more likely than tails.

We can use entropy as defined in Equation 2-7 to understand why the Twenty Questions strategy of asking about each of the 64 squares in turn is much less efficient than the strategy illustrated in the conversation that illustrated the game at the start of this chapter. When we ask about each square in turn, the square can either be the target or not. Only one of the squares is the target, which means that the probability of selecting the target on a given question is 1/64 = 0.015625, and the probability of not selecting the target on a given question is 63/64 = 0.984375. When these two probabilities are inserted into Equation 2-7, H is determined to be 0.11612. In other words, although the strategy illustrated in the conversation solved the problem quickly because each question gleaned one bit of information, the second strategy is much more inefficient because the answer to each question will on average provide slightly more than one tenth of a bit of information. As these questions acquire less information, it is not surprising that many more have to be asked in order to locate the target square.

2.3.2 Summary and Implications

In summary, when information is received it reduces uncertainty. The mathematical theory of information measures information in terms of entropy, using a simple equation based on the probabilities that events will occur. This equation was critical for important developments in communication technology in the 20th century, and this section has shown how it can provide insight to a simple example like Twenty Questions.

Information theory inspired research in a variety of difference disciplines. For instance, there was an explosion of interest in applying information theory to experimental psychology (Attneave, 1959). However, the influence of information theory on experimental psychology did not last long (Luce, 2003). One reason for this is that information theory had little to say about important issues (such as the meaning of a message) that were of particular interest to psychologists. Furthermore, information theory may have had little to say about these issues because there are many possible analyses of the term *probability* (Mellor, 2005), and the particular analysis adopted by information theory prevents this theory from addressing specific psychological issues. Section 2.4 explores alternative analyses of probability and their relation to information theory and some of its psychological applications.

2.4 Physical and Subjective Probability

The formal measure of information provided in Equation 2-7 is founded on the assumption that information reduces uncertainty. This is why the probability that an event occurs plays a central role in the Shannon–Weaver definition of entropy. However, it is critical to realize that this formalization is based on assuming a particular definition of probability and that other definitions of probability also exist (Mellor, 2005). We see that changing our definition of probability leads to a very different approach to measuring information.

2.4.1 Physical Probability

One definition of probability is called *physical probability*, because it is based on observations of the world (Mellor, 2005). The physical probability of a particular event is simply the frequency of times it is observed occurring in the world divided by the total number of observations that are made. This is also often called the *frequentist definition of probability* and is primarily associated with Sir Ronald Fisher's (1922) pioneering work in statistics. The probabilities included in the Shannon–Weaver equation for entropy are physical probabilities and are explicitly defined as being frequentist in nature (Ashby, 1956). Thus, according to mathematical information theory (Shannon & Weaver, 1949), the role of information is to reduce uncertainty, and the kind of uncertainty that is reduced concerns the possible states of the physical world. The Shannon–Weaver mode defines entropy exclusively in terms of these worldly possibilities; the measure of information does not depend on any properties of the receiver.

Very shortly after its invention, formal information theory began to influence research in other disciplines. For example, psychology, the Shannon–Weaver equation first appeared in psychology in 1949 (G. A. Miller & Frick, 1949), and within a decade there was an explosion of information theoretic psychological research (Attneave, 1959; Quastler, 1955). Information theory continued to play a prominent role in psychological research into the 1970s (Berlyne, 1960, 1971; Garner, 1975; Leibovic, 1969). However, information theory's influence on psychology has remarkably diminished since that time (Luce, 2003). Instead, modern cognitive psychology emphasizes "the concept of 'information-processing models' in which information theory per se plays no role" (Luce, 2003, p. 185).

Why has information theory essentially disappeared from psychological research? One important reason is that the Shannon–Weaver equation is based on physical probability and measures information independently of any characteristics of the agent who receives this information. That is, entropy depends exclusively on the uncertainty of events in the world. It is not influenced by the needs, desires, beliefs, or interests of the agents who learn about this world by receiving this information. In contrast, psychologists are concerned with the content, meaning, or semantics of messages (Cherry, 1957; Dretske, 1981; MacKay, 1969). These notions are not captured by the Shannon–Weaver equation, which measures information's quantity but not its quality. Furthermore, because formal information theory relies upon physical probability, it ignores the states of agents-as-receivers. Presumably, such states (e.g., an agent's beliefs, desires, or goals) are critical in determining that the content of one received message is more important than is the content of another.

Shannon's classical information measure is not appropriate to describe behavioral processes of biological systems, the reason being that this measure is based solely upon objective properties and cannot, therefore, represent a system's individual knowledge or beliefs nor can it discriminate between events which are of great importance and others which are irrelevant for an individual system (Pfaffelhuber, 1972).

This gulf between what information theory delivers and what psychologists would like it to deliver has led to a variety of proposals for alternative formalizations of information theory (Baldi, 2002; Baldi & Itti, 2010; Belis & Guiasu, 1968; Itti & Baldi, 2009; Pfaffelhuber, 1972; Weiss, 1967). One property shared by these approaches is that they exploit a different definition of probability.

2.4.2 Subjective Probability

Subjective probability, which is sometimes called *credence* or *personal probability*, is a notion of probability that is quite distinct from physical probability (Eagle, 2011; Good, 1983; Jeffrey, 2004; Kyburg & Smokler, 1964; Mellor, 2005). Subjective probability is a likelihood assigned by an agent to an event or to a proposition; it is the probability that the agent *believes* is associated with an event. In short, subjective probability is predicted probability. Because it is founded on an agent's beliefs or expectations, an event's subjective probability can differ from its physical probability. Consider a biased coin. The physical probability of it producing heads might be 0.8.

However, at first glance I might not realize that it is biased and therefore assign 0.5 as the subjective probability of heads. Subjective probabilities are typically encountered in an alternative to frequentist statistics called Bayesian statistics (Savage, 1954, 1962).

There is nothing to prevent a variation of the Shannon–Weaver entropy equation to be formulated using subjective probabilities (Pfaffelhuber, 1972). For instance, Equation 2-8 was suggested by Pfaffelhuber as a measure of subjective entropy (H_s), where p_i is the physical probability of event i, and q_i is its subjective probability of occurrence:

$$H_S = -\sum p_i log_2 q_i \qquad (2\text{-}8)$$

One reason for measuring information using subjective probabilities is that this approach recognizes that biological agents may have different sources of uncertainty to consider. One is the physical probabilities of events or signals in the world that is dealt with by the Shannon–Weaver equation. Another is the set of possible responses that an agent could make to a received signal (Berlyne, 1960, 1971; Garner, 1975; MacKay, 1969, 2003). When a particular response to a signal is evoked, this too reduces uncertainty, but in this case the uncertainty pertains to the agent's states or behavior. The amount of information conveyed by a response could be measured by an equation that uses subjective probability.

This point is important because it can be directly linked to a theoretical perspective that was introduced in Chapter 1. Recall that probabilistic functionalism (Brunswik, 1952, 1955), as depicted earlier in Figures 1-1 and 1-7, recognized a variety of probabilistic relationships that linked the world to an agent's response to it. Some of these relationships were between the world and perceived cues. Other relationships were between the perceived cues and the agent's response. In Chapter 1, we raised the possibility that the former relationships reflected ontic uncertainty, whereas the latter reflected epistemic uncertainty. These different types of uncertainty can be linked to different definitions of probability. In particular, ontic uncertainty should be strongly related to physical probability, but epistemc uncertainty is easier to relate to subjective probability.

One reason for relating epistemic uncertainty to subjective probability is that from the perspective of psychology subjective probabilities should change because of an agent's experience with the world. Consider the biased coin again. On my first encounter with it, my subjective probability of heads could be 0.5, based on my prior experience with typical (fair) coins. However, as I watch it being tossed, and observe heads appearing far more frequently than I expect (given my starting subjective probability), I might revise my subjective probability. One reason that subjective probability is associated with Bayesian statistics is because Bayes's theorem, which we discuss in more detail later in this monograph, can be used to update subjective probability on the basis of new evidence (Edwards, Lindman, & Savage, 1963; McGrayne, 2011; Molina, 1931). In short, although physical probabilities are constant properties of events in the world, subjective probabilities are not and can be modified by learning (Pfaffelhuber, 1972).

Chapter 1 also provided a discussion of probabilities that change with experience, such as when human participants learn to match probabilities or when perceptrons learn to estimate probabilities. Clearly probability judgments in both of these examples change as a result of experience, which indicates that these judgments cannot be physical probabilities. This is the reason that when the perceptron was linked to Brunswik's (1952, 1955) probabilistic functionalism (Figure 1-7), changes in connection weights could alter only epistemic uncertainty (i.e., the probabilistic links between cues and action). As far as probability probability theory is concerned, perceptrons modify subjective probability.

Another example of updating subjective probabilities via learning is provided by a recent mathematical theory of surprise (Baldi, 2002; Baldi & Itti, 2010; Itti & Baldi, 2009). This theory assumes that an agent has a set of models about the world and assigns a subjective probability to each, where this probability indicates the agent's credence in each model. As a result, the agent has a distribution of subjective probabilities. For example, in the coin-tossing example, a person might have various models about a coin like $P(H) = 0.1$, $P(H) = 0.2$, and so on. Each of these models is then assigned a subjective probability; for instance, someone who has never experienced a biased coin might assign a very high subjective probability (e.g., 0.95) to the model $P(H) = 0.5$ and assign very low subjective probabilities to all of the other models. Bayes rule can then be used to update the distribution of subjective probabilities as new evidence is observed. For instance, after watching a number of tosses of the biased coin, this updating would increase the subjective probability assigned to the model $P(H) = 0.8$ and would decrease the subjective probability assigned to the model $P(H) = 0.5$.

The model just described updates subjective probability on the basis of surprise, where surprise is defined in terms of a mismatch between one's predictions about what should happen in the world and one's observations of what actually happens. With respect to the biased coin, one is surprised by seeing so many heads appear as it is tossed; it is this surprise that motivates the updating of subjective probabilities. After an appropriate amount of updating, less surprise occurs because one's expectations about the world have been modified by experience. Important to note, this notion of surprise leads to a different view of information, one that depends upon an agent's states. Recall that it is common to describe information as something that we receive when we learn something that we did not previously know (Attneave, 1959; Garner, 1975; MacKay, 1969). As far as Shannon's theory of information is concerned, what we did not know only concerns the states of the world. In contrast, Baldi's theory of surprise is concerned with our lack of knowledge that results in the poor fit between our model of the world and its actual properties.

This is exactly the view of learning that provides the foundation for the Rescorla–Wagner model that was introduced in Section 2.1. Given the formal connection between modern perceptron and the Rescorla–Wagner model (Dawson, 2008), this also indicates that the gradient descent rule (Equation 1-6) is a method for updating subjective probability. Later in this monograph we relate the properties of this learning rule, and of the simple networks it can train, to the concepts of probability and information that have already been introduced. Let us begin by first describing the properties of a perceptron trained on a simple problem and by then relating this learning to basic notions of probability, information theory, and feedback.

2.5 Information, Learning, and Feedback

2.5.1 Teleological Behavior

In the early 20th century, biology was in the midst of a debate between mechanists and vitalists (Bertalanffy, 1933, 1952). Mechanists believed that organisms were in essence complex machines that could be explained by appealing to physical principles. In contrasts, vitalists believed that living organisms possessed some vital force that was separate from physical principles. One kind of evidence to which vitalists appealed involved entelechy or teleology: the development or behavior of organisms seemed guided or destined to achieve some purpose.

At first glance, perceptron learning seems teleological. Consider the development of probability estimation. This proceeds as if the goal of the perceptron is to match its responses to the actual probabilities of events in the world. From the perspective of vitalism, one could say that the purpose of the perceptron is to adapt to its environment. We shortly see that there is no need to appeal to vitalism to explain probability matching, but let us first describe some perceptron behavior that could be viewed as being teleological.

Consider a simple demonstration in which a modern perceptron learns about the results of tossing two different coins. One of these coins is fair, with a probability of producing heads after being tossed, $P(H)$, of 0.5. The other is biased, with $P(H) = 0.75$. Examples of tosses of these two coins can be presented to a perceptron that has two input units. These units indicate to the perceptron which coin is being tossed. If the fair coin is being tossed, then Input Unit 1 is activated with a value of 1. If the biased coin is being tossed, then Input Unit 2 is activated with a value of 1. If the result of a coin toss is heads, then the desired activity of the perceptron is 1; otherwise, it is 0. Note that these input values simply distinguish one coin from the other. There is no explicit signal to the perceptron that one coin is fair and that the other is not.

The perceptron learns the $P(H)$ of each coin by finding out about the results of a number of tosses of each coin. A training set that repeats tosses of each coin, but that also indicates that tosses of the same coin can lead to different results, can be used to model the probabilistic behavior of the two coins. Such a training set is provided in Table 2-1. This training set consists of eight different stimuli. The first four stimuli represent tosses of the first coin, and the second four represent tosses of the second coin. The $P(H)$ of the fair coin is represented in the training set by having two of these stimuli associated with a desired activity (T_p) of 1, and the other two with a desired activity of 0. That is, if the result of a coin toss is heads, then $T_p = 1$. The second four stimuli in the training set represent tosses of the second, biased coin. The $P(H)$ of the biased coin is represented in the training set by having T_p equal to 1 for three of its tosses, and equal to 0 for the fourth toss.

A modern perceptron with two input units and one output unit was trained on the stimuli represented in Table 2-1. The output unit used a logistic activation function (Equation 1-4 and Figure 1-7). The two connection weights from the input units to the output unit were initialized to random values in the range between −0.1 and 0.1; the output unit's bias was initialized to 0. The weights and bias were trained using the gradient descent learning rule (Equations 1-6 and 1-7) with a learning rate of 0.1. Training was conducted with the Rosenblatt program that is available as freeware (Dawson, 2005). During a single epoch of training, each of the eight stimuli from Table 2-1 was presented to the perceptron, but their order was randomized. The perceptron's bias and weights were updated after every stimulus presentation. The network was trained for

Table 2-1. A training set that presents the results of tossing two different coins.

Pattern	a_1	a_2	T_P
1	1	0	1
2	1	0	1
3	1	0	0
4	1	0	0
5	0	1	1
6	0	1	1
7	0	1	1
8	0	1	0

Note. The tossing of the fair coin is represented by the activity of input unit 1(a_1*); the tossing of the biased coin is represented by the activity of input unit 2 (*a_2*). If the coin produces heads, then the desired perceptron response (*T_p*) is 1. Each input pattern is repeated four times to make the training set probabilistic: The probability of heads for the fair coin is 0.5 (two heads in four tosses) but is 0.75 for the biased coin (three heads in four tosses).*

300 epochs, and its properties were recorded as it learned to examine how its behavior changed during this training.

Figure 2-2 illustrates the perceptron's responses (i.e., its output activity) to each of the coins as it learned about their behavior. Early in training it generates activity of about 0.5 to each, which is expected given an initial bias of 0 and two near-zero weights (see Figure 1-7 and assume a near zero net input and bias). As training proceeds, its response to each coin begins to increase, and then activity to the fair coin decreases while activity to the biased coin continues to rise. After about 150 epochs of training, the perceptron generates correct probability estimates: Its activity to the biased coin has stabilized at 0.75, and its activity to the fair coin has stabilized at 0.5. Its responses do not change from these levels with additional training.

Figure 2-2 illustrates that accurate probability estimation takes a certain amount of training to develop, meaning that the network's probability estimates are poorer during earlier epochs of training, and improve with more training. Figure 2-3 provides a summary of this improvement in performance of the network's responses as training proceeds. This is done by plotting the network's sum of squared error (SSE) as a function of training epoch. The SSE plotted in Figure 2-3 is computed by taking the squared difference between the network's response to each coin and the coin's actual *P(H)* and adding them together. This measures the network's total error in responding, assuming that the network is learning to correctly estimate probabilities. Note that this error starts at a higher value, because early in training the perceptron does not know much about the behavior of the two coins. As training proceeds, there is an exponential decrease in SSE, which asymptotes to zero after about 150 epochs of learning. We saw in Figure 2-2 that after 150 epochs, perceptron responses stabilize at the actual probabilities of the two coins and do not change with additional training. This produces an SSE that stabilizes at zero.

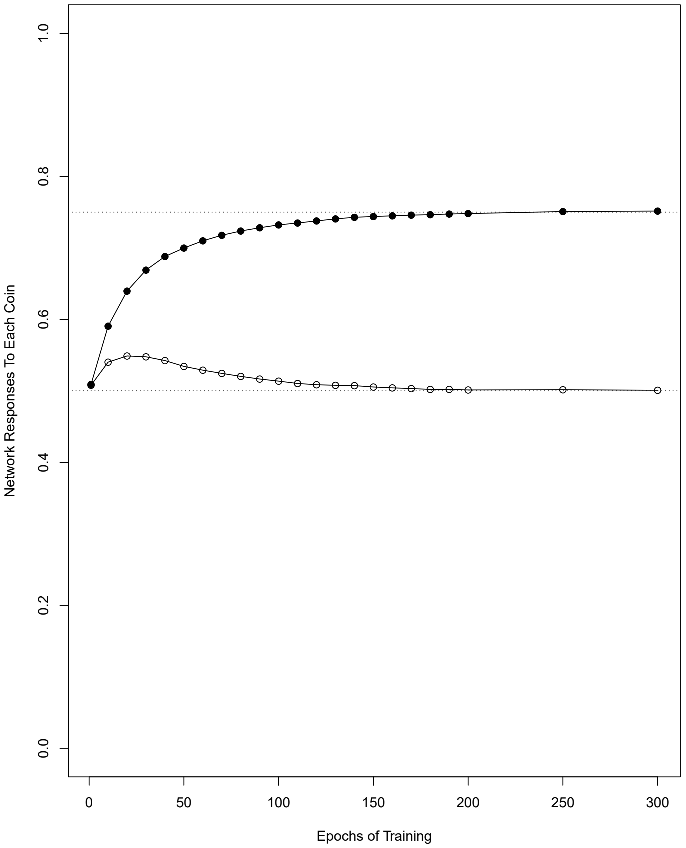


Figure 2-2. Response of the perceptron to the two different coins over 300 epochs of training. The filled circles are responses to the biased coin, and the unfilled circles are responses to the fair coin.

Together, Figures 2-2 and 2-3 illustrate that the perceptron learns about the behaviors of the two coins as it is repeatedly presented the results of their being tossed. Although the perceptron begins with no knowledge about each coin, it quickly learns to match each of their probabilities. This learning is interesting in three respects.

First, both figures indicate that perceptron responses change over time and that there is more learning accomplished early in training than later in training. This latter point is reflected, for instance, in the exponentially decreasing shape of the Figure 2-3 graph. This is interesting because throughout training, the *P(H)* of each coin is held constant. That is, the *physical* probability of each coin revealing heads never changes. What changes instead are the perceptron's probability estimates—its *subjective*

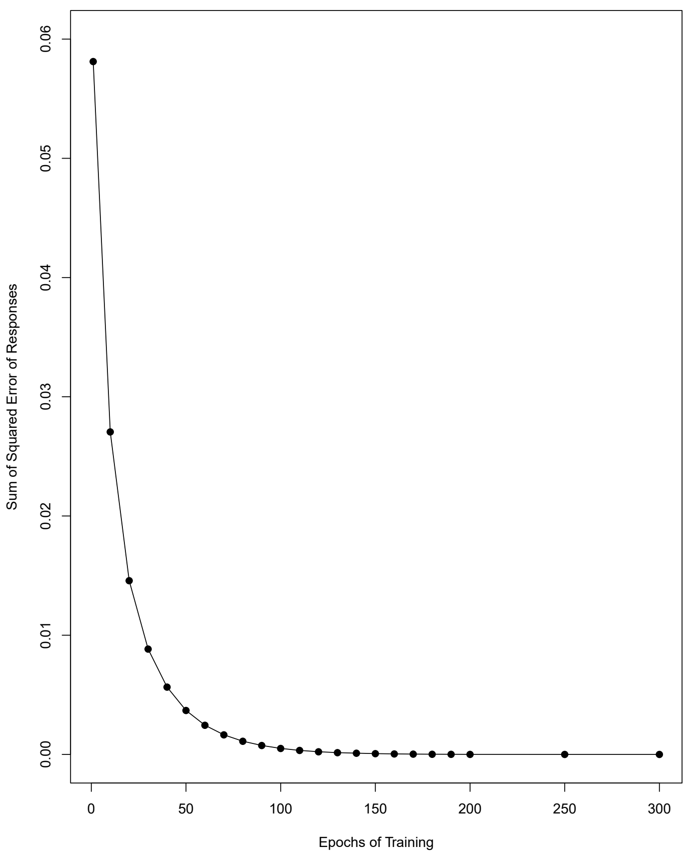


Figure 2-3. Sum of squared error of the perceptron's responses to the two coins as a function of epochs of training.

probabilities. Furthermore, recall the discussion in Section 2.2 that information tells us something that we do not already know. Clearly, Figures 2-2 and 2-3 indicate that the perceptron must be receiving more information earlier in training than later.

Second, although both Figures 2-2 and 2-3 evaluate perceptron performance by comparing its responses to physical probabilities, it is important to realize that these physical probabilities are never explicitly provided in the training set. As it learns, the perceptron is trained to generate either a 1 or a 0 (see the final column in Table 2-1). In terms of what the perceptron is actually trained to do, its error never reaches zero because it never generates a 1 when presented a coin. Instead, it produces a fractional response that eventually matches the physical probability of heads associated with each coin. This is the best that the perceptron can do given the nature of the training set. However, observers of the perceptron with knowledge of the probability structure of its environment can see that its behavior is optimal in the sense of probability estimation. It is from this perspective that an outside observer can evaluate perceptron performance as is done in the two figures.

Third, in the early 20th century the learning illustrated in the two figures might be interpreted as teleological evidence. That is, when confronted with an uncertain environment, the perceptron reorganizes itself to adapt to it as best as possible. However, we now proceed to argue that this sort of vitalistic perspective is unnecessary. Perceptron learning can easily be linked to mechanistic properties that are core ideas in information theory and cybernetics.

2.5.2 Negative Feedback and Learning

In the latter half of the 19th century, researchers became interested in explaining how biological organisms maintain their integrity in the face of varying and uncertain environmental influences. Pioneering French physiologist Claude Bernard proposed that a key function of biological systems was to maintain a constant internal environment (*milieu intérieur*; Foster, 1899). In the early 20th century, American physiologist Walter B. Cannon (1932) elaborated Bernard's ideas with experiments that revealed mechanisms for maintain what Cannon named *homeostasis*. Around the same time, theoretical biologist Ludwig von Bertalanffy (1933, 1952) proposed a related idea, the notion of the organism as an open system. Although the closed systems of physics are doomed by the second law of thermodynamics to become less organized as their entropy increases, an open system can take in new resources to be used to maintain internal organization. A variation on this idea was popularized in a very influential work, called *What Is life?*, by the Nobel Prize–winning Austrian physicist Erwin Schrödinger (1945).

Schrödinger's (1945) essay was influential because its core ideas were strongly related to modern developments in information theory and cybernetics. Although the closed systems of physics necessarily become more disordered or exhibit positive entropy, Schrödinger noted that biological systems resist this trend because they are continually drawing negative entropy from their environment. That is, although the process of being alive produces disorder or positive entropy, these consequences can be reversed by absorbing negative entropy or order from the environment. Important to notey, Schrödinger defined negative entropy as follows:

$$Negative\ Entropy = k \cdot log(\frac{1}{D}) \qquad (2\text{-}9)$$

In Equation 2-9, k is the Boltzman constant (3.2983 × 10−24) and D is some measure of molecular disorder. Note that there is a strong relationship between Schrödinger's definition of negative entropy from physics (Equation 2-9) and the Shannon–Weaver mathematical measure of information (see Equations 2-6 and 2-7). Thus, when Schrödinger described a system as consuming negative entropy from the environment, he was in essence saying that a system is obtaining information from the environment. "Thus the device by which an organism maintains itself stationary at a fairly high level of orderliness (= fairly low level of entropy) really consists in continually sucking orderliness from its environment" (Schrödinger, 1945, p. 73). To suck orderliness from its environment is to receive environmental signals that permit an organism to adjust its internal states to achieve a desired goal (e.g., a stable or organized state). This idea arises from the strong link between mathematical information theory and the field of cybernetics that arose in the middle of the 20th century (Wiener, 1948).

Wiener (1948) used the term *cybernetics* to name "the entire field of control and communication theory, whether in the animal and the machine" (p. 19). Cybernetics was particularly interested in understanding the relationship between an agent's actions and the world. This is because changes in the world caused by these actions could in turn determine the nature of future actions. This circular relationship between agent and environment is called *feedback*.

Cybernetics could rigorously study feedback relationships because it defined machines or agents very abstractly as devices that simply receive signals and then generate signals in response (Ashby, 1956). This functional approach, which focused on information processing and ignored physical composition, permitted cybernetic theories to be applied to both biological agents and mechanical devices. As far as cybernetics was concerned, feedback was a relationship between signals: The environment sent a signal to a device, which elicited a response. This response could in turn influence the nature of future signals received from the environment.

Cyberneticists were particularly interested in using feedback to explain teleological or goal-directed behavior. In this context, the response generated by a machine has a desired goal in terms of the environment. The signal returned by the environment provides information informing the agent about the difference between the desired and the actual state of affairs in the world—called *negative feedback*. The agent uses negative feedback to alter its signal in such a way to make a smaller difference between the desired and actual state of affairs. In other words, iterations of negative feedback enable or guide a system to achieve an intended environmental goal.

Cybernetics' very general definition of *machine* leads to a blurred distinction between agents and environments. For instance, consider Ross Ashby's (1960) homeostat, which was a device consisting of four identical electrical devices that could receive signals from an external user but that also sent and received signals among one another. The signals between components of the homeostat were sent through weighted connections; negative feedback was used to alter these weights. One could describe the homeostat as a four-unit device that interacted with the environment. However, from another perspective the environment for one component of the homeostat includes the device's other three components. From this perspective, the homeostat was a device that used negative feedback to adjust its internal structure to return to a stable state after encountering external or internal disturbances and thus demonstrated how negative feedback could be used to explain homeostasis.

Critically, the homeostat's use of negative feedback to alter its internal structure to achieve a goal (internal stability) is mirrored by perceptron learning. The gradient descent rule used to modify the perceptron's connection weights (Equations 1-6 and 1-7) provides another example of using negative feedback to alter a system's internal structure. The negative feedback is the error signal in the learning rule: $T_j - a_j$. The weight changes defined by gradient descent learning are designed to reduce this error the next time this particular stimulus is encountered by the network. In short, gradient descent learning is an example of learning via negative feedback. Thus, Figure 2-2 illustrates how, as learning trials proceed, negative feedback guides the perceptron's responses to match the physical probabilities associated with each of the coins being tossed.

From the perspective of cybernetics, negative feedback is information that is received by an adaptive system. Negative feedback is a signal that indicates the difference between actual and desired states, and this signal will become smaller as a system adapts to improve its performance. Because negative feedback is information, it is measurable; this measure should become smaller as learning proceeds. What kind of measure reflects this expectation?

According to mathematical information theory, the amount of information provided when a coin toss reveals heads is $-1log_2p_H$, where p_H is the physical probability of heads. As this measure uses physical probability, it is unchanging. We have seen that perceptrons generate subjective probabilities. Thus if we measure the amount of information in the context of this subjective probability (Pfaffelhuber, 1972), the appropriate equation is $-1log_2a_i$, where a_i is the perceptron's response to coin i. This second equation can be considered a measure of the information that the perceptron already has, in the sense that the perceptron's response is based upon its current state. So, the amount of information that is received from a coin toss can be measured as the absolute value of difference between the information provided by the environment and the information that has already been required, as given in Equation 2-10:

$$H = |(-1 \cdot log_2 p_H) - (-1 \cdot log_2 a_i)| \qquad (2\text{-}10)$$

Figure 2-4 provides a graph of the values of Equation 2-10 determined for each of the two coins in the simulation at different points in training. It can be seen that the coin tosses provide the greatest amount of information early in training, because the perceptron has little information about coin behavior. However, the initial state of the perceptron is such that its predictions about the fair coin are much more accurate than its predictions about the biased coin. This is why Figure 2-4 indicates that the perceptron is receiving more information about the biased coin than it is about the fair coin. As training proceeds, the perceptron's responses to the coins become more and more accurate, and the amount of information being received from a toss of either coin drops off exponentially, mirroring the shape of the error curve provided by Figure 2-3.

2.5.3 Imperfect Learning

Figures 2-2, 2-3, and 2-4 suggest that the perceptron is near perfect in its learning about the probability structure of the two-coin environment. After an appropriate amount of training, its responses—its subjective probabilities—match the physical probabilities of each coin producing heads, its SSE has reached zero, and additional coin tosses are providing the perceptron no new information.

Although this interpretation of the graphs is correct from the perspective of an external observer with knowledge of the environment's physical probabilities, it is incorrect from the perspective of the perceptron itself. This is because the perceptron is *not* being trained to generate probabilities. Recall that the training set provided in Table 2-1 indicates that for the perceptron to be correct, it would have to generate a response of 1 or 0 to any stimulus. However, Figure 2-2 shows that the perceptron *never* does this—its responses *always* fall in the range between 0.50 and 0.75. In other words, from the perceptron's perspective it is *always* making a mistake, and is *always* learning, even though from the outside looking in its behavior is optimal.

This suggests that the graphs provided in the three figures do not reflect perfect learning but instead indicate that the perceptron has achieved what systems theorists call a *dynamic equilibrium*. In such a state, the internal components of a system are always changing, but the system's integrity or behavior appears to be constant. The dynamic equilibrium of the perceptron is one in which its responses reflect the physical probabilities of its environment (Dawson & Dupuis, 2012). As learning proceeds, the perceptron has reached a state much like homeostasis: By constantly monitoring its environment (i.e., by learning), it has achieved a state that produces stable behavior. By continually receiving information about the errors it produces, it modifies its structure to keep these errors to a minimum. This is how the perceptron sucks orderliness from its environment (Schrödinger, 1945). In the next section, we explore how the perceptron responds when its environment is suddenly altered.

The (eventually) stable responses of the perceptron raise a second interesting issue. The perceptron is never explicitly trained to generate probabilities, but its responses can clearly be interpreted as conditional probabilities (Figure 2-2). Where do these probabilities come from? Table 2-2 provides the internal structure of the perceptron (its two connection weights and its bias) at different points in its training. These values represent the perceptron's "knowledge"—for lack of a better term—about its world, and how this "knowledge" changes over time.

We can see from Figure 2-2 that this internal structure can be transformed into highly accurate predictions about physical probability. Nevertheless, it is not clear at first glance what the relationship is between network structure and probabilities. Chapters 3 and 4 provide an account of how perceptron structure explicitly represents probability and relate this structure to Bayesian probability models.

2.6 Adapting to Surprise

2.6.1 Learning from Surprises

This chapter began by asking what information is. The broad answer to this question is that after we receive

Table 2-2. The structure of the perceptron after different epochs of training.

Epoch	w_1	w_2	θ
1	0.01115	0.006243	0.025024
50	0.515976	–	0.330293
150	0.700994	–	0.365044
200	0.721087	–	0.366859
250	0.730617	–	0.372237
300	0.734072	–	0.372459

Note. The first column identifies the epoch of training; w_1 *and* w_2 *are the weights from the inputs representing the biased and the fair coin respectively, and* θ *is the bias in the output unit's logistic activation function.*

information, we know something that we did not know before (Attneave, 1959; Cherry, 1957; Garner, 1975; MacKay, 1969). The mathematical theory of information expresses this as reducing uncertainty and defines uncertainty in terms of physical probabilities of events in the world. Contrasting to this notion of information is the companion concept of learning. That is, when we acquire information that reduces our uncertainty about the world, there is a tacit assumption that this information is stored within us. To know something that we did not know before is to learn something new.

The perceptron, and its training, provides a concrete example of such learning, but this learning alters subjective probability, not physical probability. The perceptron is not merely a passive receiver of information. Instead, when it receives a stimulus, it generates a prediction. Learning proceeds by comparing this prediction to the desired value (e.g., Equation 1-6). The perceptron learns about the accuracy of its prediction and does not learn about the uncertainty of stimuli. In other words, the perceptron combines information defined in terms of physical uncertainty with an additional uncertainty, the uncertainty of the response the perceptron generates to its inputs (MacKay, 1969).

The information acquired by the perceptron causes it to change responses, as is evident from Figure 2-2. Keeping this in mind, let us again consider what it means when information is obtained. "Fundamentally it implies that in some circumstance or other my *expectations* will be different. I am now conditionally *ready to react differently*" (MacKay, 1969, p. 60). When information is acquired, predictions or expectations are altered. Conversely, if predictions or expectations are such that they are continually supported by environmental results, then they will not be changed, because no information is obtained (see Figure 2-4). This is the central idea shared by perceptron learning rules such as Equation 1-6 (Dawson, 2008), the Rescorla–Wagner model of associative learning (Rescorla & Wagner, 1972), and the cybernetic notion of negative feedback (Ashby, 1956). All of these theories are grounded in the idea that changes occur only when predictions about the world are not confirmed. Learning is driven by mistakes.

2.6.2 Disrupting Dynamic Equilibrium

Section 2.4.3 argued that the shapes of the curves in Figures 2-2, 2-3, and 2-4 illustrated a dynamic equilibrium. In such an equilibrium, the components of a system change, but the system's overall behavior does not. In the case of the two-coin perceptron simulation, the structure of its weights and bias had reached a point where they were constantly being changed by a learning rule, because the perceptron was never generating completely correct responses. However, these changes were minimal because the network's predictions were as good as possible. As a result, perceptron responses stabilized to match physical probabilities.

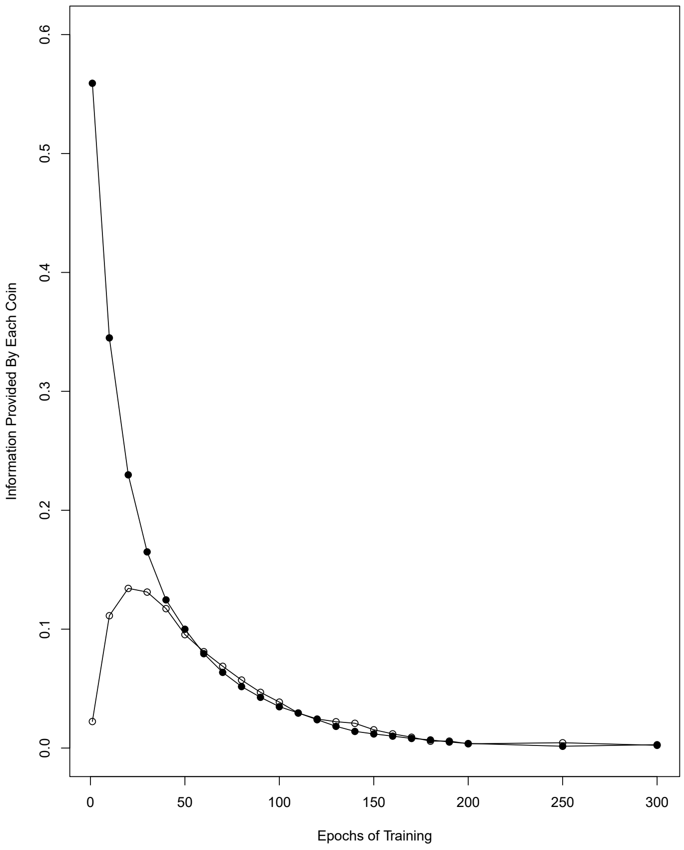


Figure 2-4. The information provided to the network by each coin when a coin toss produces "heads." The information is measured in bits and is determined by Equation 2-10. The filled circles represent the biased coin, and the unfilled circles represent the fair coin.

What is important about the dynamic equilibrium of a perceptron is that because it is always learning (even though its responses have stabilized), it is always ready to adapt to changes in the environment. One source of environmental uncertainty is the stochastic properties associated with its elements, such as the probability of each coin revealing heads. Another source of environmental uncertainty involves changes in these stochastic properties. For instance, imagine that the probability of "heads" for each of the coins suddenly changed at some point during training. In this case, the fact that the perceptron is always in a state in which it learns from surprises permits the network to change its structure to adapt to these environmental changes (Dawson et al., 2009a).

To illustrate this idea, let us consider a second two-coin simulation. In this simulation, the procedure for training the perceptron is identical to that described in Section 2.4.1. For the first 150 epochs, the perceptron learns about the training set whose properties were provided in Table 2-1. However, for the next 150 epochs the training set is different: The fair coin suddenly becomes biased, and the biased coin suddenly becomes fair. This is accomplished by changing one of the fair coin's T_p values from 0 to 1 and by changing one of the biased coin's T_p values from 1 to 0. With this change, *P(H)* for the (originally) fair coin becomes 0.75 and becomes 0.5 for the (originally) biased coin. The responses of the perceptron to each coin during 300 epochs of training are provided in Figure 2-5.

The first half of Figure 2-5 is identical to the first half of Figure 2-2: The 150 epochs of training results in the perceptron responses matching the physical probability of "heads" associated with each coin. In Figure 2-2, these responses were maintained as training proceeded. However, in Figure 2-5 the physical probabilities are switched after Epoch 150. As a result, the expectations that that perceptron has learned are no longer accurate. Its poorer predictions provide more negative feedback, and the perceptron's structure changes to adapt to the new environment. Probability matching to the new physical probabilities has been achieved after a further 150 epochs of training.

This chapter has shown that perceptron learning is driven by the amount of negative feedback (e.g., Equation 1-6). As perceptron responses become more accurate during the first 150 epochs of training, negative feedback decreases. However, when the physical probabilities of the two coins are swapped after 150 epochs of training, the network's predictions immediately become less accurate, because these predictions no longer match the properties of the altered environment. The sudden rise in negative feedback midway through training is presented in Figure 2-6, which provides the sum of squared error for the perceptron as a function of training epochs. The first half of Figure 2-6 is essentially identical to 2-3 and shows an exponential decrease in error as learning proceeds. When the physical probabilities are switched at Epoch 150, there is a massive increase in network error. At this point, network SSE is higher than it was when training began. This is because at the start of training, the network's response to the fair coin was already accurate, and most of the network error was related to its failed predictions about the biased coin. However, the first 150 epochs of training drove the perceptron's responses to both coins equally far (about 0.25) from the switched physical probabilities, producing even greater error. After the switch, the learning processes exponentially decrease the SSE again and it approaches 0 after a further 150 epochs of training.

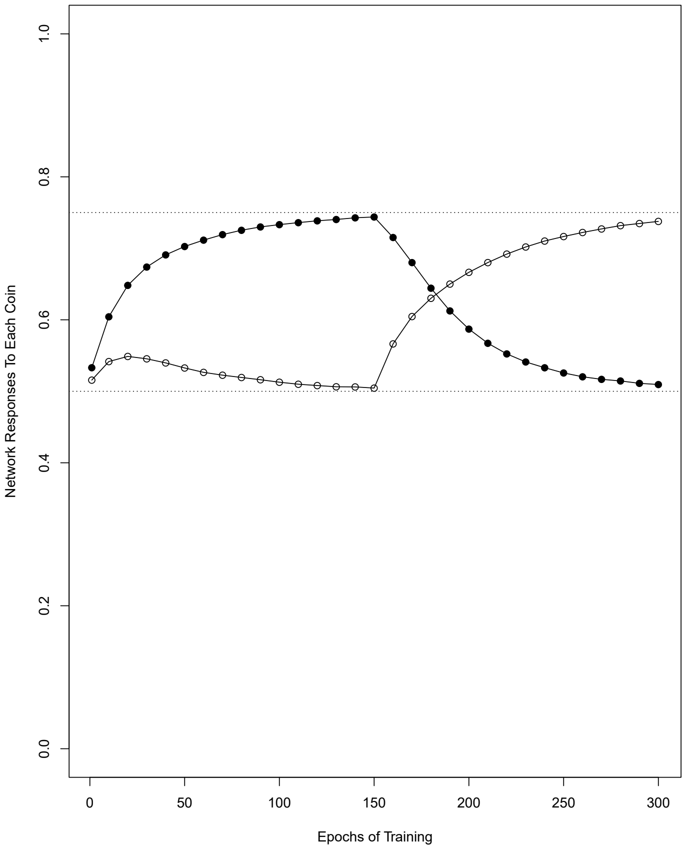


Figure 2-5. Responses of the perceptron to the two different coins over 300 epochs of training. In this simulation, the probability of revealing "heads" switches between the two coins after 150 epochs of training. The filled circles are responses to the (initially) biased coin, and the unfilled circles are responses to the (initially) fair coin.

Figure 2-4 was used to show that as network responses improve, the amount of information received by the network decreases. That is, the network receives less information because its predictions about the environment become more accurate. For this demonstration, the information received by the network was taken as the difference between the amount of information defined in terms of physical probability and the amount of information defined in terms of subjective probability (Equation 2-10). This latter notion of information reflected a measure of what the perceptron already "knows."

Figure 2-7 graphs the same measure for the two coins for the simulation in which physical probability changes midway through training. Not surprising, the first half of this graph is essentially the same as Figure 2-4. The two graphs change when the physical probabilities are changed after Epoch 150. At this point in training, the perceptron suddenly receives an increase in negative feedback because its predictions are less accurate. As a result, there is an abrupt increase in the information provided by each

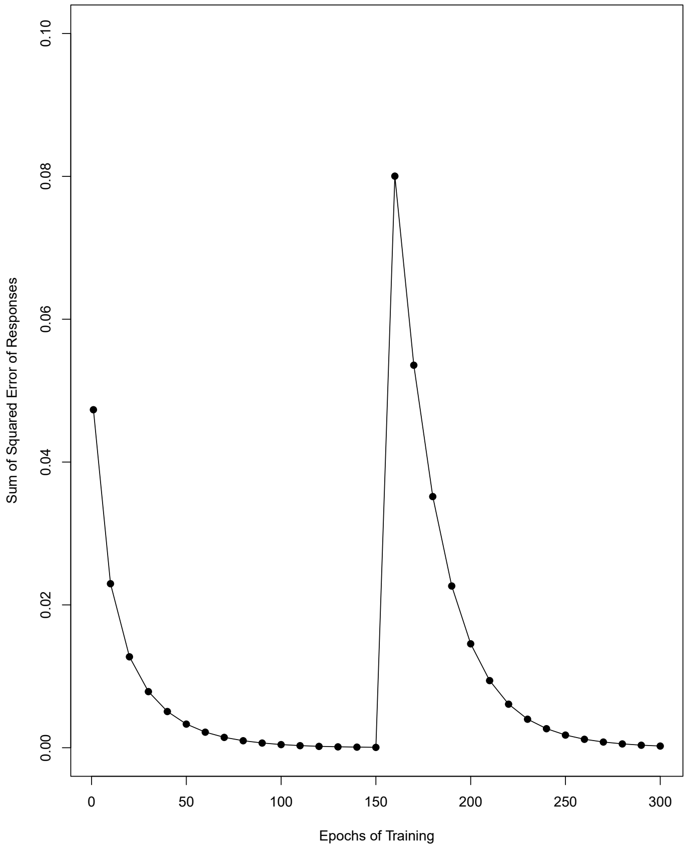


Figure 2-6. Sum of squared error of the perceptron's responses to the two coins as a function of epochs of training. The sudden rise in SSE that occurs after 150 epochs is the result of switching the physical probabilities of the two coins.

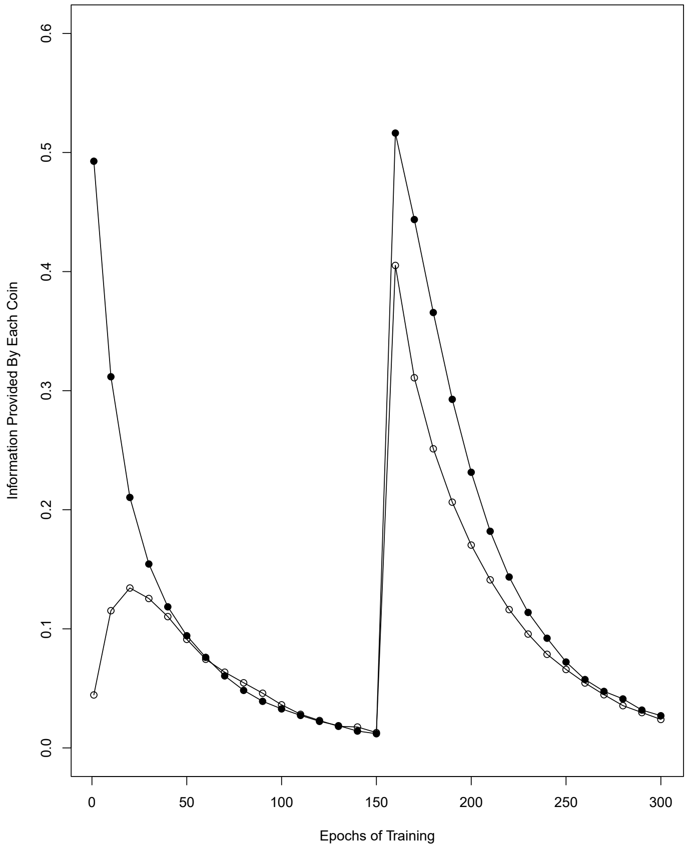


Figure 2-7. The information provided to the network by each coin when a coin toss produces "heads." The information is measured in bits using Equation 2-10. The filled circles represent the (initially) biased coin, and the unfilled circles represent the (initially) fair coin.

coin revealing "heads." Of interest, there is a difference between the information provided by each coin, as indicated by the separation of the two lines around the midpoint of the graph. This difference is because the (originally) fair coin provides less information because its *P(H)* is higher (and therefore less informative) than is the case for the (originally) biased coin. As learning proceeds, there is an exponential decrease in the information provided by the coin tosses, and this value is near zero by Epoch 300.

To summarize our observations about the second two-coin simulation, we have seen that the perceptron's learning process permits its responses (predictions, subjective probabilities) to correctly estimate the physical probabilities of the coins after about 150 epochs of training. We saw earlier (e.g., Figure 2-2) that if these physical probabilities remain unchanged, then the perceptron stabilizes into a dynamic equilibrium. However, in the second simulation this equilibrium is disrupted by changing the physical probabilities of the coins. This disrupts the perceptron's dynamic equilibrium. The perceptron's ongoing learning mechanisms quickly adapt it to the altered environment and attain a new dynamic equilibrium that correctly estimates the new physical probabilities.

What changes permit the perceptron to adapt to the environmental changes? Table 2-3 provides the values of the perceptron's weights and bias after different amounts of training. It can be seen that the first three rows of Table 2-3 have very similar values to the first three rows of Table 2-2, indicating a similar pattern of weight changes.

This is expected, because both perceptrons adapted to the same environment for the first 150 epochs of training. The two tables diverge after Epoch 150. Noticeable changes in the perceptron's structure are already evident in Table 2-3 by Epoch 160. These changes continue to Epoch 300. Notice that the final row in Table 2-3 indicates that the perceptron has a bias similar to the one that it had at Epoch 150 but that the values of its two weights have roughly been swapped at Epoch 300 in comparison with Epoch 150.

Given that the physical probabilities were themselves swapped, it is perhaps not surprising that the perceptron has used this method to deal with the new environment. Of course, for the time being we do not

Table 2-3. The structure of the perceptron after different epochs of training.

Epoch	w_1	w_2	θ
1	0.110867164	0.041364814	0.021338718
50	0.573078	-0.15574	0.286448
150	0.748277	-0.29972	0.317662
160	0.568454	-0.08533	0.352235
200	0.047384	0.387848	0.304338
250	-0.19696	0.627531	0.299676
300	-0.27572	0.720228	0.313614

Note. The first column identifies the epoch of training; w_1 *and* w_2 *are the weights from the inputs representing the (initially) biased and the (initially) fair coin respectively, and* θ *is the bias in the output unit's logistic activation function.*

have an obvious account of how the network structure from Table 2-3 is converted into probabilities; this point was also made about Table 2-2. By the end of Chapter 4, we will have a detailed understanding of how to relate perceptron structure to probability.

2.7 Summary and Implications

2.7.1 Summary

Chapter 1 introduced the perceptron, along with learning rules that permit this kind of network to adapt to uncertain environments. It also demonstrated that perceptrons could learn to estimate the probabilities of the environment. The purpose of Chapter 2 was to relate the topics of Chapter 1 to concepts rooted in the 20th-century study of control and communication: measures of information, physical versus subjective probability, and negative feedback.

Chapter 2 explored the very general definition that to receive information is to reduce uncertainty about the world. This idea was examined by reviewing how mathematical information theory used physical probability to measure information in bits. However, the fact that perceptron responses can be interpreted as probabilities (Section 1.6), combined with the fact that these responses are altered by learning, indicates that perceptron responses are subjective probabilities, not physical probabilities. In other words, perceptrons make predictions about physical probabilities, these predictions are subjective probabilities, and the role of learning is to improve these predictions.

Learning was then related to cybernetics by pointing out that the perceptron learning rule was an obvious example of negative feedback. To be precise, the error term in Equations 1-6 and 1-7 provides negative feedback—the discrepancy between the environment and perceptron predictions. The purpose of learning is to reduce this discrepancy, which is equivalent to reducing negative feedback. This was illustrated with a simple simulation in which a perceptron learned about the behavior of two coins being tossed, one fair and the other biased. It was shown that the responses of the perceptron changed over time and stabilized when probability matching was achieved. That is, the perception stabilized when the perceptron's subjective probabilities were equal to the physical probability of "heads" for each of the coins. This behavior was considered in terms of the actual responses (Figure 2-2), exponentially decreasing errors (Figure 2-3), and decreasing amounts of received information (Figure 2-4).

It was then pointed out that the behavior of the perceptron in the two-coin simulation did not reflect perfect learning but instead achieving a dynamic equilibrium. In this equilibrium, the network is constantly learning because its responses are never completely correct. This constant learning makes small changes to perceptron structure, and the result of these changes is to stabilize its responses. This was illustrated in a second simulation in which the probability of heads for each coin was suddenly swapped midway through training. This led to a sudden increase in negative entropy to which the perceptron quickly adapted by adopting a new internal structure (Figures 2-5, 2-6, and 2-7). Thus, one way to view the perceptron is as a system that is constantly monitoring negative feedback as it copes with an uncertain environment. It maintains a structure that minimized negative feedback by keeping its predicted probabilities close to actual probabilities and is capable of adapting its structure when physical probabilities change.

One consequence of viewing the perceptron in this way is that it recognizes that perceptrons alter their weights only when negative entropy is encountered. For example, imagine a perceptron that is being trained on the two-coin task but that has had its initial weights set so that the perceptron's initial responses are much more accurate. This means that the perceptron will learn little, because its initial predictions will not generate much negative entropy. However, learning mechanisms are still active, and if the situation changes in such a way that the perceptron no longer makes accurate predictions, then it will quickly adapt to reduce the newly detected negative entropy.

2.7.2 Implications

The simple simulations described in this chapter have been used to consider perceptron learning in the context of information theory and cybernetics. The results that we have reported are consistent with other results in the literature. Previous research has shown that perceptrons learn to estimate probabilities when receiving four different and mutually exclusive signals and quickly learn to modify these predictions when reward contingencies change midway through training (Dawson et al., 2009). Perceptrons also learn to estimate probabilities when multiple cues that signal different probabilities of reward are presented at the same time (Dawson & Gupta, 2017), a capability that is explored in more detail in Chapter 4.

We have seen that the link between perceptron learning, information theory, and cybernetics is the notion that perceptrons generate predictions about the environment (subjective probabilities) and then use negative feedback to improve these predictions. One way to interpret a perceptron's response is as its prediction of the likelihood that it will be rewarded given a particular stimulus (Dawson & Dupuis, 2012; Dawson et al., 2009; Dawson & Gupta, 2017). That cognitive systems are driven by generating predictions or expectations is an idea of growing popularity in cognitive science (Clark, 2016; Hohwy, 2013). To some, this idea is very new. For example, Hohwy (2013) claimed "a new theory is taking hold in neuroscience" (p. 1), where this theory is the view that the brain is a mechanism that is constantly minimizing prediction errors. Although it is good news that neuroscientists and cognitive scientists have increasing interest in this theory, one of the implications of Chapter 2 is that this idea is far from new, having been established in cybernetics in the middle of the 20th century (Ashby, 1956, 1960; Wiener, 1948, 1950) and having been exploited by psychological learning theory not long after (Rescorla & Wagner, 1972).

Cognitive science's growing interest in the importance of minimizing errors of prediction has important implications for the study of this phenomenon. These implications are used to structure the approach taken over the next few chapters. Cognitive science proceeds by examining cognitive phenomena at distinct levels of analysis, with each level being associated with different core questions (Dawson, 1998, 2013). At the computational level of analysis, formal methods are employed to answer the question, What information processing problem is a system solving? At the algorithmic level of analysis, empirical methods are used to answer the question, What procedures are being used to solve this information processing problem? At the implementational level of analysis, the techniques of neuroscience explore answers to the question, How are these procedures physically realized by the brain?

Much of Chapter 2 concerned discussions at the algorithmic level of analysis, because this chapter has been primarily concerned with observing the behavior of perceptrons when confronted with uncertain environments. In particular, the two-coin simulations provide empirical results concerning how perceptron responses change over time and how the perceptron responds to changes in its environment. These algorithmic investigations are important because they suggest that perceptrons learn to estimate physical probabilities, and do so using negative feedback.

However, the simulations raise other questions that need to be explored at the computational level of analysis. Although our observations of two different perceptrons in this chapter are consistent with probability estimation, it is important to establish at a more abstract or formal level that "probability estimation" is the information-processing problem that is being solved. In particular, a formal proof that perceptrons generate probabilities would permit us to establish this as a general property of this type of network, one that could be claimed of *any* perceptron that learned about an uncertain environment. In addition, a computational-level analysis of probability matching in perceptrons should provide the means for explaining how network structure (Tables 2-2 and 2-3) is converted into probability estimates.

In addition, it is important to remember that the perceptron is an extremely simple artificial neural network because it does not use intermediate processors called hidden units. It has long been known that this places important limits on the kinds of information processing problems that perceptrons can solve (Minsky & Papert, 1969). A computational analysis of how perceptrons solve probabilistic information problems should not only explain how network structure is translated into probabilities but also show how this structure is limited. That is, a computational analysis should show what kinds of probabilities *cannot* be computed by perceptrons because of their simple structure.

The next two chapters shift the focus of this monograph from the algorithmic level to the computational level. They provide a detailed formal analysis of perceptron structure to show how it is converted into a particular kind of probability and then prove the existence of particular limitations on the kinds of probabilities that a perceptron can generate. Chapter 3 begins this computational analysis by taking seriously the notion developed in Chapter 2 that perceptrons generate subjective probabilities. Such probabilities are typically linked to Bayesian notions concerning probability (Savage, 1954, 1962). Chapter 3 introduces Bayes's theorem and then explores the formal relationship between it and the behavior of the modern perceptron. It does so for a very simple situation, one for which probability judgments are made by processing a single cue. It concludes that perceptrons are Bayesian systems.

Chapter 4 builds on the proof developed in Chapter 3 and explores the Bayesian behavior of perceptrons confronted with environments that are more complicated. In particular, Chapter 4 studies the probabilities generated by perceptrons when they are confronted with more than a single cue. It proves that the perceptron will generate accurate probabilities if the cues are independent of one another. However, if there are interactions between cues, then probability estimates will not be accurate. This is because the simple structure of a perceptron means that it is a naïve Bayesian mechanism; the proof of this claim is the major contribution of Chapter 4.

Chapter 3: Bayes's Theorem, Perceptrons, and Odds Ratios

This chapter provides the first steps of a computational analysis of how perceptrons adapt to uncertain environments. It does so by considering an extremely simple uncertain environment, one in which there is a single cue that can be present or absent, and a single reward that can be given or not. The chapter first describes this environment in the context of contingency theory, which is a particular perspective on associative learning in animals. It introduces two key notions that are central to contingency theory: the 2 × 2 contingency table, and a measure of contingency or association called ΔP. It then uses the 2 × 2 contingency table to introduce Bayes's theorem for solving inverse probability problems. An example inverse probability problem is used to construct a training set for a perceptron, and after training, it is shown that the perceptron's responses are equal to the values predicted by Bayes's theorem. Bayes's theorem, based on the 2 × 2 contingency table, is then translated into the logistic activation theorem, proving that the modern perceptron is a Bayesian mechanism. This proof also reveals that a perceptron's structure—its weight or bias—represents the natural logarithm of a basic element of probability theory called an odds ratio. The chapter ends by returning to contingency theory and proposing that odds ratios, which have been ignored by contingency theorists in psychology, have advantages over other popular measures of contingency like ΔP.

3.1 From Algorithm to Computation

Probability learning, as presented in the context of human experiments and computer simulations in Chapters 1 and 2, has been described using behavioral examples. At the end of Chapter 2, it was pointed out that such evidence is acquired at cognitive science's algorithmic level of analysis. Algorithmic analyses typically proceed by observing the behaviors of agents in either natural or experimental settings (Dawson, 1998, 2013). These observations are then used to support inferences about the information processing that underlies the behavior.

Such algorithmic-level analyses are critical for advancing cognitive science. However, cognitive science cannot rely upon them exclusively. This is because algorithmic analyses only capture a subset of cognitive regularities; to capture other regularities one must move away from the algorithmic level (Pylyshyn, 1984).

For instance, consider searching for the limits of some type of information processing. To do so at the algorithmic level would require observing behavior in a wide variety of experimental or natural settings, all the while looking for situations that cause difficulties (e.g., situations in which agents make particular errors). Such a search can be costly and inefficient, particularly if a researcher does not have specific ideas about what processing is being carried out by an agent or about the limitations of such processing.

Computational-level analyses can be used to inform and to formulate algorithmic investigations of limitations. A computational-level analysis answers the question, What information-processing problem is being solved? (Marr, 1982; Richards, 1988). It does so by performing formal analyses to determine the function that maps the stimuli that an agent receives into the responses that the agent generates. That is, at the computational level a cognitive scientist views an agent in exactly the same way that cyberneticists (Ashby, 1956, 1960) viewed a machine: abstractly, as a device that computes an input–output function. A computational-level analysis uses a formal language, like mathematics or logic, to prove that an agent can be described as computing a particular function. That computational analyses are expressed as formal proofs provides a generality and permanence that is not possible with empirical methods: "The power of this type of analysis resides in the fact that the discovery of valid, sufficiently universal constraints leads to conclusions … that have the same permanence as conclusions in other branches of science" (Marr, 1982, p. 331).

The result of a computational analysis is important because it can be used to guide investigations at other levels of analysis. For example, imagine that a computational analysis has proven that an agent is computing some input–output function *f()* and that it is not computing some other function *h()*. This proof rigorously specifies the input–output mapping used to describe the agent's behavior. Thus, it indicates not only what the agent is capable of performing but also what the agent is *not* capable of performing. Such a specification can be used to guide the algorithmic search for an agent's limitations.

For example, by the end of Chapter 4 we will have established that a perceptron that receives signals from multiple cues computes a naïve Bayesian probability function. This means that the perceptron will generate correct probability estimates when there are no interactions between the cues. However, as interactions between cues become larger and larger, there will be greater errors in the perceptron's responses. This result permits us to focus our experimental studies of human participants by pointing to particular limitations of performance (Chapter 7). In particular, we can determine whether human participants are also naïve Bayesians by examining whether human probability matching is also hindered by interactions among cues. In short, a limitation established at the computational level can be used to motivate empirical investigations at the algorithmic level.

The result of a computational analysis is also important because it can provide a rich understanding of how a system's properties are (formally) involved in mediating an input–output mapping. For instance, simulation results provided in earlier chapters indicate that it is plausible to interpret a perceptron's response as an estimate of the likelihood of receiving award given a particular set of signals. However, our cursory examination of perceptron structure (weights and bias; see Tables 2-2 and 2-3) did not clearly indicate how it was capable of converting inputs into probabilities. The computational analysis that is presented in this chapter, and continued in the next chapter, rectifies this situation. It will explicitly define the kind of probability that a perceptron is generating and clearly indicate the relationship between perceptron structure and the computing probability.

The purpose of Chapter 3 is to lay the foundation for a computational analysis of probability estimation in perceptrons. The chapter begins by introducing a general definition of a very simple uncertain environment that is a variation of a probability discrimination task. This environment involves a single cue or signal, and a reward that occurs with a particular frequency when the cue is present and with a different frequency when the cue is absent. The structure of this environment can be represented as a set of frequencies organized in a simple 2×2 contingency table.

The chapter then proceeds to use this 2×2 contingency table to define a particular model of probabilistic inference, Bayes's theorem. Bayes's theorem is used to define an inverse probability problem, which is the probability that a particular hypothesis is true given that certain evidence has been observed. This theorem is important in statistical inference and strongly tied to the notion of subjective probability introduced in Chapter 2.

Chapter 3 then briefly continues with an algorithmic-level examination of a very simple perceptron, one in which a single input unit is connected to a single output unit. A 2×2 contingency table is used to define a training set for this perceptron. At the end of training, it is shown that the response generated by the perceptron when the single cue is present is identical to the probability computed by applying Bayes's theorem to the contingency table.

The heart of Chapter 3 follows and is concerned with a computational-level analysis of a perceptron faced with a single, uncertain cue. This computational analysis proceeds by using the elements of the contingency table to convert Bayes's theorem into an equation that has the same form as the logistic activation function used by the perceptron (Equation 1-4). In short, we prove that we can translate Bayes's theorem into a perceptron's structure. This shows that this simple perceptron is a device that computes Bayes's theorem. Furthermore, the details of this proof provide a rigorous account of how the structure of a perceptron computes Bayesian probability. It is proven that the weight and bias of this simple perceptron are defined in terms of basic probability elements known as odds ratios.

The computational analysis in Chapter 3 is based on a simple problem (involving one uncertain cue) because this leads to a tractable mathematical analysis of a very simple network (involving a single input unit). After establishing this approach, we will be in a position in Chapter 4 to extend it to situations that are more complex. This in turn permits us to understand mathematically the limitations imposed by using perceptrons to adapt to these more complex environments.

3.2 Contingency and Uncertainty

In Chapter 2, a particular psychological theory of associative learning, the Rescorla–Wagner model of Pavlovian conditioning (Rescorla & Wagner, 1972), was introduced as an example of learning via negative feedback. In this model, when a learning agent is presented a stimulus, it makes a prediction about its consequences. It learns when this prediction is not accurate and updates the association between stimulus and response in such a way that its error will be reduced the next time the stimulus is encountered. Important to note, there is a formal relationship between the Rescorla–Wagner model and the training of perceptrons (Dawson, 2008; Gluck & Bower, 1988; Sutton & Barto, 1981). This formal relationship provides one example of how an important model of animal learning can be recast into a learning rule for an artificial neural network.

The purpose of this section is to introduce another such relationship, by relating a different view of animal learning called *contingency theory* to the modern perceptron.

Classical conditioning (Pavlov, 1927) is an experimental paradigm that provides the evidence underlying a particular view of learning by humans and other biological agents. In classical conditioning, the presentation of an unconditioned stimulus (US) causes the learner to make an unconditioned response (UR). Another stimulus, the conditioned stimulus (CS) does not cause this response. However, by pairing presentations of the CS with the US, an association between the CS and the response is learned. After a period of such classical conditioning, the CS also causes the response, which is now labeled the conditioned response (CR). Classical conditioning is an example of exploiting one of the basic laws of association psychology, the law of contiguity (Warren, 1921), where the contiguity involves the temporal pairing of the US, CS, and the UR.

The preceding description provides what Rescorla (1967) called the "American" view of Pavlovian conditioning. This view focuses exclusively on the frequency of pairings of the US (or, more generally, reinforcement) and the CS. This, in turn, leads to a theory that relies completely on excitatory mechanisms—that is, the ability of the CS to signal an imminent US.

An alternative approach to formalizing associative learning is called *contingency theory* (Rescorla, 1967, 1968). The "American" view of associative learning examines only the frequency of pairings between the CS and the US. In contrast, contingency theory looks not only at this pairing but also at the frequency with which the CS and the US are *not* paired. That is, learning is related to two conditional probabilities: the probability of the US given the CS—P(US|CS)—and the probability of the US in the absence of the CS: P(US|~CS). In contingency theory, learning occurs only if there is a difference between these two conditional probabilities. Note that defining learning in terms of this difference permits inhibitory processes to be important: If P(US|~CS) is higher than P(US|CS), then an agent can learn that the CS is a signal that the US is *not* forthcoming.

Studies of contingency theory require experimental paradigms that differ from those used to study classical conditioning. One such paradigm was pioneered in the 1960s (Jenkins & Ward, 1965). Jenkins and Ward presented their participants with problems in which each of two responses (pressing Button 1 or pressing Button 2) was paired with one of two possible outcomes (score or no score). After 60 trials of this type of learning, participants used a 100-point scale to rate the degree of control they believed that their responses had on the outcome. This rating measures beliefs about contingency, where contingency is an assessment of the strength of the relationship between a stimulus and a response.

For this methodology to inform contingency theory, the contingency between response and outcome must be manipulated. Jenkins and Ward (1965) did so by considering their design in the context of a simple 2×2 contingency table like the one provided in Table 3-1. This table provides the frequencies with which different stimulus-response pairings occur. For example, the value *a* in Table 3-1 is the number of times that a participant scored when that person pressed Button 1, and the value *b* is the number of times that a participant did not score when that person pressed the same button.

Table 3-1. A generic 2×2 contingency table Jenkins and Ward's (1965) study.

	Score	**~Score**
Press Button 1	a	b
Press Button 2	c	d

Note. Each row is associated with one of a subject's possible actions. Each column is associated with an outcome, indicating whether the action leads to a score. Each of the four letters in the contingency table stands for a frequency. For instance, a indicates the number of times that pressing Button 1 leads to a score

By manipulating the values of four frequencies in Table 3-1, one can vary the strength of the relationship, or the contingency, between action and outcome. Jenkins and Ward quantified this contingency as the difference between the conditional probability of scoring when Button 1 is pressed—P(*Score*|*Press Button 1*)—and the conditional probability of not scoring when Button 1 is pressed: P(*~Score*|*Press Button 1*). This difference between conditional probabilities is known as ΔP, and it can be computed from the four frequencies in Table 3-1 using Equation 3-1. Jenkins and Ward manipulated the Table 3-1 cell frequencies to create different learning situations in which ΔP assumed values of 0, 0.3, 0.6, or 0.8. Of interest, one of the main findings of this study was that ΔP was not strongly related to subjects' judgments of control!

$$\begin{aligned} \Delta P &= \frac{a}{a+b} - \frac{c}{c+d} \\ &= \frac{ad-bc}{(a+b)(c+d)} \end{aligned} \tag{3-1}$$

Perhaps because of this negative experimental result, contingency theory was dormant for a while. This changed with a seminal study in the late 1970s (Alloy & Abramson, 1979). Alloy and Abramson employed a version of the Jenkins and Ward paradigm to study contingency judgments of students who were either depressed or not. Their main result was that depressed participants made contingency judgments that were highly consistent with the objective contingencies produced by manipulating the cell frequencies of Table 3-1. This study was followed by a long history of studies exploring human sensitivity to contingencies in the world (Shanks, 2007). One consequence of research into human contingency learning has been a debate about whether such learning is best explained in terms of associative mechanisms or in terms of cognitive inference. Another interesting property of this research is that it almost exclusively operationalizes contingency in terms of differences between conditional probabilities, such as ΔP (Allan, 1980; Allan & Jenkins, 1980, 1983; P. W. Cheng, 1997; P. W. Cheng & Holyoak, 1995; P. W. Cheng & Novick, 1990, 1992; Jenkins & Ward, 1965; Ward & Jenkins, 1965; Wasserman, Dorner, & Kao, 1990).

Contingency theory is particularly relevant to the main theme of this monograph, which concerns how agents can adapt to an uncertain world. One reason for this is evident from a close examination of Table 3-1. If the values of, for instance, *a* and *b* are both greater than zero, then this table represents an uncertain world. This is because if both of these frequencies are greater than zero, then pressing Button 1 does not reliably lead to the same response—in some instances this action is rewarded with a score, but in other instances this same action is not rewarded. In other words, experiments like those by Jenkins and Ward (1965) that manipulate contingency by varying the Table 3-1 frequencies can be seen as experiments that study learning about uncertainty. In fact, Table 3-1 provides the probability structure of an extremely simple probability discrimination task of the sort that was introduced in Chapter 1.

Contingency theory is also strongly related to another core topic of this monograph, the modern perceptron. In earlier chapters, we already interpreted perceptron responses as being conditional probabilities (i.e., estimates of the probability of reward given the presence of particular cues). Is perceptron behavior related to contingency? It has been proven that when perceptron learning reaches a dynamic equilibrium of the sort illustrated in Figure 2-2, the difference between the perceptron's response when a cue is present and its response when a cue is absent equals ΔP (Dawson & Dupuis, 2012). In other words, comparing perceptron behavior in the presence and absence of cues provides a measure of contingency.

The current chapter exploits the notion of contingency in yet another fashion in order to relate perceptrons to probability theory. It uses the conditional probabilities from Table 3-1—the probabilities used to compute ΔP—as a bridge between topics that have already been introduced (perceptrons, probability) and a related topic, Bayesian inference. Bayesian inference involves computing conditional probabilities, in particular the probability that some hypothesis is true given that some evidence is also true. In this chapter, we introduce Bayes's theorem of conditional probabilities for the particularly simple situation that can be represented with a 2×2 contingency table like the one provided in Table 3-1. We then explore the ability of a modern perceptron to learn about contingencies that can be represented in this fashion. In other words, we explore teaching perceptrons in a fashion that is analogous to the Jenkins and Ward (1965) paradigm. To preview the main point of this chapter, we discover that the modern perceptron learns to make judgments of conditional probabilities that are equivalent to those predicted by Bayes's rule. This means that modern perceptrons can literally be understood as Bayesian mechanisms. Furthermore, we can use the frequencies in Table 3-1 to define the weight and bias of a modern perceptron that mimics the conditional probabilities defined by that table.

This result leads to another reason that contingency theory is related to the current chapter. As just noted, human studies of contingency learning focus on the ΔP metric. However, other important metrics are more common when one moves outside of this particular literature and considers the general statistical analysis of contingency tables (Christensen, 1997; Hosmer & Lemeshow, 2000; Ku & Kullback, 1974; Lindley, 1964; Rudas, 1998). One of these metrics is the natural logarithm of a probability measure called the odds ratio. This chapter shows that this measure defines the weight of a modern perceptron that learns to perform Bayesian inference on a 2×2 contingency table. This, combined with the known relationship between perceptrons and associative learning in general (Dawson, 2008), raises the possibility that perceptrons can inform contingency theory. In particular, the interpretation of perceptrons that learn to make contingency judgments suggests that the natural logarithm of the odds ratio is an interesting alternative to ΔP.

3.3 Bayes's Theorem and Cognitive Science

Consider the following scenario: You are given an urn of marbles, and you are told that 25% of them are white and the remaining 75% are black. You are then asked the following question: If five times in a row you randomly draw a marble from this urn and then put it back in, what is the probability that you will produce the following sequence of marbles: white, white, white, black, and black?

This question is known as an a posteriori probability problem. Its solution requires using knowledge about a population to make an inference about a sample taken from that population. In the example problem, we know that in this population the probability of drawing a white marble—P(W)–is one out of four and that the probability of drawing a black marble—P(B)—is three out of four. With this knowledge, combined with our knowing that these probabilities do not change (because the marbles is replaced after being drawn) and our knowing that each draw is independent of the others, we can answer this question by multiplying the probabilities of each individual event occurring. That is, $P(\mathrm{WWWBB}) = P(\mathrm{W})\bullet P(\mathrm{W})\bullet P(\mathrm{W})\bullet P(\mathrm{B})\bullet P(\mathrm{B}) = (¼)3\bullet(¾)2 \approx 0.00879$.

The solution of a posteriori probability problems is at the heart of the frequentist approach to statistics (Neyman, 1937). According to this approach, conducting an experiment is akin to taking a single sample from a population of such experiments. A statistic is computed for this sample (e.g., a *t* test, an *F* test) and is then compared to a known population distribution of this statistic (e.g., a *t* distribution, an *F* distribution). If the value of the sample statistic is sufficiently unlikely to be drawn by chance from the population distribution (often a probability of .05 or smaller), then the researcher concludes that the result of the experiment is highly likely due to an independent variable and therefore reveals a statistically significant effect. This is an example of an a posteriori application of probability because an inference about the sample is made by comparing it to a known population.

Now consider a different version of the probability problem. You are told that marbles are being independently drawn from an urn and that after being drawn, a marble is returned the urn. You are then informed that with five such draws, the result was the following sequence of marbles: white, white, white, black, and black. You are then asked a different kind of probability question: Given this result, what is the probability that the urn from which the marbles were drawn consisted of 25% white marbles and 75% black marbles?

This second question is an example of what is known as an a priori probability problem or an *inverse probability* problem: Given the existence of some evidence *E*, what is the likelihood that this evidence resulted from hypothetical cause *H* (Molina, 1931)? This is an inverse probability problem because the direction of inference is opposite from that of the earlier a posteriori probability problem. In this second case, the task is to use knowledge of particular evidence to make an inference about the properties of an unknown population.

Informally, inverse probability problems are solved by hypothesizing a number of populations that could produce the evidence. For example, we might approach the marble problem by hypothesizing one urn that contains 100% white marbles, a second urn that contains 99% white marbles and 1% black marbles, a third urn that contains 98% white marbles and 2% black marbles, and so on. We assign each of these possible populations a subjective probability, where this probability is our initial estimate of how likely it is that it is producing the evidence. We then use the evidence to update our probabilities; with each new piece of evidence, some of these probability estimates will increase, whereas others will decrease. For instance, as soon as we draw the first white marble, the probability associated with the hypothetical urn containing 100% black marbles is reduced to zero. Bayesian inference provides mathematical procedures that can be used to update these probabilities based on evidence, with the goal of assigning the highest probability to the most likely hypothetical population. These procedures provide the basis for Bayesian statistics (Cornfield, 1962, 1967, 1969; Edwards et al., 1963; Jeffreys, 1939; McGee, 1971; Savage, 1954, 1962, 1972). Although Bayesian statistics were controversial through most of the 20th century, they are now widely established (Kruschke, 2011; Lunn, Jackson, Best, Thomas, & Spiegelhalter, 2012; Woodward, 2012) and have even achieved mainstream popularity (McGrayne, 2011; Silver, 2012).

In this chapter, our interest is in establishing a formal relationship between Bayesian inference and the modern perceptron. To do so, we start with the simplest situation, where instead of considering a large number of hypothetical causes, we consider only one, which can be of two states only, true or not (*H* or ~*H*). In this simplest case, the evidence from which we make our inference can also be of only two states, true or not (*E* or ~*E*). In this simplest scenario, Bayesian inference applies an equation with roots that can be traced back to the 18th century (Bayes, 1763) and is now known as Bayes's theorem:

In Equation 3-2, $P(H|E)$, called the posterior probability, is the conditional probability of interest, the probability that *H* is true given that *E* is true. The conditional probability $P(E|H)$, called the *likelihood*, is the probability that *E* is true if one assumes that *H* is true. The probability term $P(H)$, called the *prior*, is the probability that *H* is true. The sum of products in the denominator normalizes the expression to ensure that the computed value for $P(H|E)$ remains in the range from 0 to 1.

$$P(H|E) = \frac{(P(E|H)\cdot P(H)}{\left((P(E|H)\cdot P(H)\right) + \left((P(E|{\sim}H)\cdot P({\sim}H)\right)} \tag{3-2}$$

A 2×2 contingency table, of the type that was introduced in Section 3.2, expresses the data required to perform Bayesian inference using Equation 3-2. Table 3-2 illustrates the general form of this contingency table. Its two rows correspond to the two possible states of the evidence, and its two columns correspond to the two possible states of the hypothesis. The four letters inside the table represent frequencies of co-occurrence.

Table 3-2. A generic 2×2 contingency table for applying Bayes'

	H	**~H**
E	a	b
~E	c	d

Note. E indicates that some evidence is true, whereas ~E indicates that the evidence is not true. Similarly, H indicates that the hypothesis is true, whereas ~H indicates that the hypothesis is false. Each of the four letters in the contingency table stands for a frequency. For instance, a indicates the number of observations for which both the evidence and the hypothesis are true.

The four cell frequencies in Table 3-2 provide all of the information required to define the probabilities required by Equation 3-2. To be specific, $P(E|H)$ is $a/(a + c)$, $P(H)$ is $(a + c)/(a + b + c + d)$, $P(E|{\sim}H)$ is $b/(b + d)$, and $P({\sim}H)$ is $(b + d)/(a + b + c + d)$. To perform Bayesian inference for this single hypothesis case, we use the contingency table to compute these probabilities and then insert the values of the probabilities into Equation 3-2 to determine the inverse probability.

Consider an example in which Bayes's theorem is used to measure the relationship between mammograms and breast cancer (McGrayne, 2011). In this example, the evidence is whether a mammogram is positive for breast cancer (E) or not (${\sim}E$). The hypothetical cause related to this evidence is whether a patient has breast cancer (H) or not (${\sim}H$). In this example of 1,000 hypothetical patients, there are eight patients who have positive mammograms and have breast cancer. There are 95 patients who have positive mammograms but do not have cancer. There are two patients who have negative mammograms but do have cancer. Finally, there are 895 patients who have negative mammograms and do not have cancer. Table 3-3 presents this information in a 2×2 contingency table.

Table 3-3. A contingency table for the mammogram/ breast cancer example of Bayes'

	H	**~H**
E	8	95
~E	2	895

Of concern in the example problem from McGrayne (2011) is the following question: If a patient's mammogram is positive, then what is the probability that she has cancer? The answer to this question comes from using Bayes's theorem to determine the conditional probability $P(H|E)$. The four cell frequencies in Table 3-3 are all that are needed to accomplish this. Using the four probability equations that were given in the discussion of Table 3-2, we can determine that $P(E|H)$ equals 0.8, $P(H)$ equals 0.01, $P(E|{\sim}H)$ equals 0.9595959595, and $P({\sim}H)$ equals 0.99. When these four probabilities are inserted into Equation 3-2, $P(H|E)$ is computed to equal approximately 0.0777. In other words, given the Table 3-3 data, Bayes's rule indicates that a positive mammogram means that there is less than an 8% chance of having breast cancer.

One can also modify Equation 3-2 to compute a different posterior probability, $P(H|{\sim}E)$. Equation 3-3 presents the required modification. When Table 3 is used to compute the components of Equation 3-3, we find that P($H|{\sim}E$) equals approximately 0.0026, which is the probability that a patient has breast cancer even though her mammogram did not detect it.

$$P(H|{\sim}E) = \frac{(P({\sim}E|H)\cdot P(H))}{(P({\sim}E|H)\cdot P(H)) + (P({\sim}E|{\sim}H)\cdot P({\sim}H))} \tag{3-3}$$

Equations 3-2 and 3-3, and generalizations of them, provide the foundations of Bayesian statistics (Edwards et al., 1963; Savage, 1954, 1962). Traditional statistics, founded upon the frequentist notion of physical probability, focus upon comparing a property of a sample to a known population distribution of this property to test the statistical significance of a null hypothesis. In essence, it uses a probability value—p value—associated with this statistical comparison to determine whether some effect is either present or not. There is a growing concern about the limits of this approach because it is recognized that p values do not measure the size of an effect or the importance of a result; alone a p value is not a good measure of the evidence supporting a model or hypothesis (Wasserstein & Lazar, 2016). Accompanying such unease is a growing movement toward New Statistics—adopting statistical methods that focus on improving estimates of effect sizes and reducing our uncertainty about these estimates (Cumming, 2013, 2014). Not surprising, Bayesian statistics may serve as the foundational formalism for the New Statistics (Kruschke & Liddell, 2017), particularly as introductory texts and computer algorithms for performing such analyses are now widely available (Albert, 2009; Kruschke, 2011; Lunn et al., 2012).

Important to note, Bayesian methods do not merely provide mathematical procedures for statistical inference. They are enormously flexible. Bayesian methods have long been used to develop optimal procedures for searching for difficult targets, such as sunken vessels (Stone, 1975). Bayesian models also have a long history in the pattern recognition literature (Duda & Hart, 1973; Duda, Hart, & Stork, 2001). Although Savage's (1954) classic text is titled *The Foundations of Statistics*, it has been described in far broader terms as presenting Savage's theory of Bayesian decision making (Binmore, 2009).

As many interpret Bayes's rule as describing a type of decision making or reasoning, it is not surprising that many cognitive scientists are exploring the possibility that human cognition is essentially Bayesian in nature. Bayesian probability theory is argued to provide a more appropriate formalism for describing cognition than does the truth-functional logic that was used to establish cognitive science in the 20th century. "Cognition should be understood in terms of probability theory, the calculus of *un*certain reasoning, rather than in terms of logic, the calculus of certain reasoning" (Oaksford & Chater, 2007, p. 7). This position is particularly evident in the cognitive science of human reasoning and rationality, which explicitly appeals to Bayesian probability (Binmore, 2009; Chater & Oaksford, 1999; Glymour, 2001; Lee & Wagenmakers, 2013; Oaksford & Chater, 1991, 1998, 2001, 2007). However, Bayesian cognitive science is not limited to this domain. For example, Bayesian accounts of neural systems are also appearing (Doya, 2007; Nessler, Pfeiffer, Buesing, & Maass, 2013; Rao, Olshausen, & Lewicki, 2002), not to mention similar accounts of musical cognition (Temperley, 2004) or of attention and perception (Itti & Baldi, 2009).

What is the relationship between Bayes's rule and Bayesian cognitive science? At the most general level, it would be expected that an agent who employed Bayesian cognition would generate responses that are consistent with Bayes's theorem. That is, an agent who employed subjective probabilities, and who updated these probabilities in accordance with Equation 3-2 or some variant, could be described as "being Bayesian." In Bayesian cognitive science, Bayes's theorem should provide a normative account of the behavioral relationship between states of evidence and hypothetical causes.

In cognitive science, when formal equations or proofs are used to specify the input/output function mediated by an information processor, it is said that this processor is being described at the computational level of analysis (Marr, 1982). By formally describing input/output regularities, a computational analysis identifies the information-processing problem being solved by the system. Explorations of Bayesian theories of human reasoning (Oaksford & Chater, 2007, 2009; Oaksford, Chater, & Larkin, 2000) have been deliberately performed at this computational level. Oaksford and Chater (e.g., 2007) pointed out that a computational theory of reasoning provides an account of a system's competence or of its ideal behavior. Bayesian probability is their choice for a computational account of human reasoning and rationality.

Important to note, cognitive science realizes that computational analyses are necessary, but not sufficient, to serve as explanations. Other levels of analysis must also be explored (Dawson, 1998, 2013). For instance, at the algorithmic level, researchers determine what information-processing methods or procedures are used to perform particular computations. Similarly, at the architectural and implementational levels, cognitive scientists explore how these information-processing procedures are brought to life in a particular physical device like the brain.

The need for these additional levels of investigation has led to an important criticism of Bayesian cognitive science: its alleged failure to ground computational theory in particular algorithms or mechanisms (Bowers & Davis, 2012; Jones & Love, 2011). Indeed, critics argue that Bayesian computation is incompatible with psychologically plausible procedures for carrying them out, such as artificial neural networks (Jones & Love, 2011).

The Bayesian approach suggests that learning involves working backward from sense data to compute posterior probabilities over latent variables in the environment, and then determining optimal action with respect to those probabilities. This can be contrasted with the more purely feed-forward nature of most extant models, which learn mappings from stimuli to behavior and use feedback from the environment to directly alter the internal parameters that determine those mappings (Jones & Love, 2011).

Furthermore, although there is a growing literature concerning the Bayesian nature of neural mechanisms (Doya, 2007; Rao et al., 2002), skeptics point out that this evidence is behavioral and not biological (Bowers & Davis, 2012). "If neuroscience is to provide any evidence for the theoretical Bayesian perspective, the key question is what non-behavioral evidence exists that neurons compute in this way? The answer is none" (Bowers & Davis, 2012, p. 404).

In spite of such criticisms, Bayesian cognitive scientists like Oaksford and Chater (e.g., 2007) are still reluctant to explore their theory of reasoning at less abstract levels of Marr's (1982) tri-level hypothesis. They believe that Bayesian algorithms or mechanisms will at best only approximate their computational theory.

> We suspect that, in general, the probabilistic problems faced by the cognitive system are simply too complex to be solved directly, by probabilistic calculation. Instead, we suspect that the cognitive system has developed relatively computationally 'cheap' methods for reaching solutions that are "good enough" probabilistic solutions to be acceptable. (Oaksford & Chater, 2007, p. 15)

In the next section, however, we explore the possibility that Oaksford and Chater's concern about computationally cheap methods for Bayesian computations is premature. We first establish that a simple modern perceptron, trained with a gradient descent learning rule, is quite capable of computing Bayes's theorem for one hypothesis (i.e., Equation 3-2) to a high degree of accuracy. We then turn to establishing a formal relationship between the structure of a trained modern perceptron and Equation 3-2. The main point to be delivered is this: Modern perceptrons are Bayesian mechanisms.

3.4 A Bayesian Mechanism

Earlier in this chapter, we illustrated how to use Bayes's rule to compute posterior probabilities for data represented by a 2×2 contingency table. We now consider a related question: What is the simplest artificial neural network that can learn to generate these same probabilities? Modern perceptrons can learn to estimate probabilities (Dawson et al., 2009); earlier chapters presented additional examples of this ability. Formal analyses of the modern perceptron show that its output can literally be interpreted as a conditional probability: the probability of reinforcement given the presence of a particular stimulus (Dawson & Dupuis, 2012).

Bayes's rule is a conditional probability rule: It generates the probability that a particular hypothesis is true given the condition that some data are true (Equation 3-2). Given that a modern perceptron generates conditional probabilities, is such a device capable of generating the posterior probabilities defined by Bayes's rule? To answer this question, consider a training set based on the hypothetical mammogram/breast cancer example provided in Table 3-3.

The training set consists of 1,000 patterns, where each pattern in the training set represents the results of a single patient's mammogram paired with her cancer diagnosis. When E is true, it is encoded with a stimulus value of 1; $\sim E$ is encoded with a stimulus value of 0. When H is true, the perceptron is trained to produce a response of 1; when it is not true ($\sim H$), the perceptron is trained to produce a response of 0. The 1,000 different patterns in the training set reflect the frequencies of the 2×2 contingency table for this problem (Table 3-3). That is, eight patterns pair a stimulus of 1 with a response of 1, 95 pair a stimulus of 1 with a response of 0, two patterns pair a stimulus of 0 with a response of 1, and 895 patterns pair a stimulus of 0 with a response of 0.

This training set requires the simplest modern perceptron imaginable: one input unit to represent mammogram evidence, and one output unit to represent cancer diagnosis. Prior to training the single connection weight in the network has its weight ($w1$) randomly assigned a value from the range between −0.1 and 0.1. The bias θ is initialized to a value of 0. The network is trained to map each stimulus to the desired response using the gradient descent learning rule (Dawson, 2004, 2005). A learning rate of 0.1 is employed. The order of pattern presentation is randomized every epoch, where a single epoch consists of presenting each of the 1,000 training patterns once. The network is trained for a set number of epochs; this training is sufficient to produce sufficiently small and unchanging sum of squared error. This approach to stopping training recognizes that this perceptron will never generate the correct response to each pattern in the training set, because when the same stimulus is presented to the network (e.g., 1), it can lead to opposite responses (0 or 1) depending on the training pattern.

Consider the behavior of one example perceptron, trained for 5,000 epochs on this problem. At the end of this training, its squared output error, summed over the 1,000 training patterns, is 9.374. This network's single weight $w1$ is 3.466, and its bias θ (of its logistic activation function) is −5.948. When this perceptron's input unit is activated with a value of 1, the logistic activity of its output unit is 0.0771. This activity provides the network's judgment of $P(H|E)$ and is nearly identical to the value computed from Equation 3-2 that was reported earlier in this chapter. This conditional probability deviates from Bayes's rule only at the fourth decimal point. Similarly, when the perceptron's input unit is activated with a value of 0, it generates an activity of 0.0026. This activity reflects the network's judgment of $P(H|\sim E)$ and is nearly identical to the value for this conditional probability when computed using Equation 3-3.

In sum, our empirical investigations of a simple version of an inverse probability problem clearly indicate that a modern perceptron produces an input/output mapping that is consistent with Bayes's theorem (Equations 3-2 and 3-3). Let us now turn to a formal examination of the relationship between Bayes's rule and the structure of such a perceptron.

3.5 Inside a Bayesian Mechanism

What is the formal relationship between the structure of Bayes's equation and a perceptron with behavior that is consistent with this equation? To answer this question about a modern perceptron, we must explore the relationship between Bayes's theorem and the logistic activation function. To do so we take advantage of the well-established relationship between the logistic equation and probability theory (C. M. Bishop, 1995, 2006; Hastie, Tibshirani, & Friedman, 2009; McClelland, 1998). We use this relationship to translate Bayes's theorem into a form that resembles the logistic equation. We then use this resemblance to define the values of the perceptron's weight and bias. In the following we prove that the perceptron's weight and bias can be expressed using frequencies taken from the 2×2 contingency table that was used to define the training set. Furthermore, these expressions relate perceptron structure to new elements of probability theory: odds ratios.

Our first step is to convert Bayes's rule (Equation 3-2) into a format that is structurally similar to the logistic function. This is accomplished (Jordan, 1995) by dividing both the numerator and denominator of Equation 3-2 by *P*(*E*|*H*) · *P*(*H*), as shown in Equation 3-4:

$$P(H|E) = \frac{P(E|H)\cdot P(H)}{\big(P(E|H)\cdot P(H)\big)+\big(P(E|\sim H)\cdot P(\sim H)\big)} \qquad (3\text{-}4)$$

$$= \frac{1}{1+\dfrac{P(E|\sim H)\cdot P(\sim H)}{P(E|H)\cdot P(H)}}$$

The next step is to replace the probabilities in Equation 3-4 with the appropriate ratios of contingency table frequencies, where these frequencies are represented by the four letters provided in Table 3-2. Equation 3-5 illustrates that *P*(*H*|*E*) can be expressed in a very simple equation that includes the ratio between *b* and *a*.

$$P(H|E) = \frac{1}{1+\dfrac{P(E|\sim H)\cdot P(\sim H)}{P(E|H)\cdot P(H)}}$$

$$= \frac{1}{1+\dfrac{\left(\frac{b}{b+d}\right)\cdot\left(\frac{b+d}{a+b+c+d}\right)}{\left(\frac{a}{a+c}\right)\cdot\left(\frac{a+c}{a+b+c+d}\right)}} \qquad (3\text{-}5)$$

$$= \frac{1}{1+\left(\frac{b}{a}\right)}$$

Now we exploit our empirical results obtained from training the perceptron on the problem represented by the 2×2 contingency table. We demonstrated that, after training, when the network's input unit is turned "on," the network generates activity that is an excellent approximation of the value of P(*H*|*E*) delivered Bayes's rule. From this observation, let us express P(*H*|*E*) in terms of the network's logistic activation function. This is done by recognizing that when the signal *a1* is 1, the net input to the logistic is *w*1. Equation 3-6 expresses P(*H*|*E*) in terms of this logistic and then equates this to the simple expression for P(*H*|*E*) derived above in Equation 3-5.

$$P(H|E) = \frac{1}{1+e^{-(w_1+\theta)}} = \frac{1}{1+\left(\frac{b}{a}\right)} \qquad (3\text{-}6)$$

Equation 3-6 reveals a direct relationship between perceptron components and elements of the contingency table, as is made explicit in Equation 3-7:

$$e^{-(w_1+\theta)} = \frac{b}{a} \qquad (3\text{-}7)$$

By taking the natural logarithm of both sides of Equation 3-6 and by multiplying both sides by −1, we can express the two parameters of the network trained on the Bayesian problem in terms of two of the frequencies taken from the contingency table (Equation 3-8).

$$w_1 + \theta = \ln(a) - \ln(b) = ln\left(\frac{a}{b}\right) \qquad (3\text{-}8)$$

Equation 3-8 expresses a relation between two components of the perceptron to two elements from the contingency table. Our next task is to find a single expression for each of the perceptron components. We begin this task by using the approach illustrated in Equations 3-4 through 3-7 to express θ by itself in terms of contingency table frequencies. When the input unit to the trained perceptron is equal to 0, perceptron activity can be interpreted as being *P*(*H*|~*E*). However, in this second situation, the net input being passed into the logistic function is equal to 0 because the there is no signal being sent from the input through *w*1. Thus, the only contributor to logistic activity is θ. Therefore, to solve for θ we manipulate Equation 3-3 in the same way that we manipulated Equation 3-2.

We first express the Bayesian equation for *P*(*H*|~*E*) (given earlier as Equation 3-3) in a structural form that can be related to the logistic equation. This is done by dividing both the numerator and the denominator of Equation 3-3 by *P*(~*E*|*H*) · *P*(*H*), as shown in Equation 3-9:

$$P(H|{\sim}E) = \frac{P({\sim}E|H)\cdot P(H)}{\left(P({\sim}E|H)\cdot P(H)\right)+\left(P({\sim}E|{\sim}H)\cdot P({\sim}H)\right)} = \frac{1}{1+\frac{P({\sim}E|{\sim}H)\cdot P({\sim}H)}{P({\sim}E|H)\cdot P(H)}} \quad (3\text{-}9)$$

Second, we replace the probability terms in Equation 3-9 with contingency table frequencies taken from Table 3-2. Equation 3-10 shows that $P(\mathrm{H}|{\sim}\mathrm{E})$ can be expressed in a very simple equation that includes the ratio between *d* and *c*.

$$P(H|{\sim}E) = \frac{1}{1+\frac{P({\sim}E|{\sim}H)\cdot P({\sim}H)}{P({\sim}E|H)\cdot P(H)}} = \frac{1}{1+\frac{\left(\frac{d}{b+d}\right)\cdot\left(\frac{b+d}{a+b+c+d}\right)}{\left(\frac{c}{a+c}\right)\cdot\left(\frac{a+c}{a+b+c+d}\right)}} = \frac{1}{1+\left(\frac{d}{c}\right)} \quad (3\text{-}10)$$

Next, we recognize that when the perceptron is generating $P(H|{\sim}E)$ its input unit has zero activity. This means that the only term in the exponent that is part of the logistic equation is θ. Equation 3-11 expresses $P(H|{\sim}E)$ in terms of this version of the logistic and equates it to Equation 3-10.

$$P(H|{\sim}E) = \frac{1}{1+e^{-(\theta)}} = \frac{1}{1+\left(\frac{d}{c}\right)} \quad (3\text{-}11)$$

We again take advantage of the structural similarities of the two expressions in Equation 3-11 to state an obvious equality:

$$e^{-(\theta)} = \frac{d}{c} \quad (3\text{-}12)$$

Finally, we can take the natural logarithm of both sides of Equation 3-11 and multiply both sides by −1 to express θ in terms of contingency table frequencies, as shown in Equation 3-13:

$$\theta = \ln(c) - \ln(d) = ln\left(\frac{c}{d}\right) \quad (3\text{-}13)$$

Equations 3-9 through 3-13 provide an expression for θ alone. This expression (Equation 3-13) can be substituted into Equation 3-8 to solve for $w1$, because the θ used to compute $P(H|E)$ is the same θ that is used to compute $P(H|{\sim}E)$. That is, the one network, that has only one value of θ, can compute both conditional probabilities by turning its input unit on or off. Equation 3-14 shows the result of solving for $w1$:

$$\begin{aligned} w_1 &= \ln(a) - \ln(b) - \theta \\ &= \ln(a) - \ln(b) - \ln(c) + \ln(d) \\ &= \ln(ad) - \ln(bc) \\ &= \ln\left(\frac{ad}{bc}\right) \end{aligned} \quad (3\text{-}14)$$

For the cancer example introduced earlier, if one places the four frequencies from the contingency table into Equation 3-14, then the resulting value for the network's connection weight is approximately 3.629. Similarly, placing the appropriate frequencies into Equation 3-13, the resulting value for the network's θ is approximately −6.104. The observed values for the trained network described earlier are solid approximations of these ideal values.

There are several interesting implications of the derivations provided in this section. The first is the most important: We have demonstrated a formal equivalence between Bayes's theorem (Equation 3-2) and the components of the modern perceptron. The modern perceptron can learn to generate Bayesian posterior probabilities for cases involving a single hypothesis and a single source of evidence. The ideal values for the weight and bias of this perceptron were determined by using contingency table frequencies as a means of translating between Bayes's rule and the logistic activation function. Clearly, the modern perceptron is a Bayesian mechanism.

Second, there is an important theoretical implication that arises from the Equation 3-14 expression for the perceptron's weight. This weight is equal to the natural logarithm of a particular ratio of contingency table frequencies, *ad*/*bc*. This expression, called the *odds ratio*, is one of the most important measures of association for a contingency table (Agresti, 2002; Y. M. M. Bishop, Fienberg, & Holland, 1975; Rudas, 1998). Discovering it inside the network has some important implications for psychological investigations of how humans and animals learn contingencies. The next section introduces the notion of odds and odds ratios and then considers the relationship between these measures and psychological contingency theory.

Given the relationship between the connection weight and the odds ratio, we expect that the network's bias θ should relate to odds as well. Equation 3-13 indicated that θ is related to the ratio *c/d*. This simple ratio is literally the odds of *H* relative to *~H* but only for those cases involving *~E* (Rudas, 1998). That θ reflects such odds parallels the odds ratio interpretation of *w* nicely because θ is the sole source of output activity when the input unit is off.

We saw earlier that there is a general belief that algorithmic or implementational accounts can at best only approximate Bayesian computations (Bowers & Davis, 2012; Jones & Love, 2011; Oaksford & Chater, 2007). From the perspective of perceptrons, this makes perfect sense: Bayes's theorem (Equation 3-2) requires multiplying various probabilities together, and multiplication is not a primitive operation for a perceptron (e.g., net inputs are sums, not products, of signals). Our formal analysis reveals that the modern perceptron has discovered an elegant solution to circumvent this limitation: Because of its logistic activation function, it works in the world of logarithms, which translates multiplication and division into addition and subtraction, operations permitted in typical network architectures. Then perceptron then lets the activation function compute the antilogarithm of the network's calculations, by passing these logarithmic calculations into the exponent of *e* in its logistic activation function.

3.6 Odds, Odds Ratios, and Contingency

At the time of the National Hockey League's all-star game in 2016, the odds against the Montreal Canadiens winning the Stanley Cup were set at 50:1. What do these odds mean? In gambling, "odds against" typically express the ratio of wagers made by two parties when placing a bet. Having 50:1 odds mean that the bookmaker would place a bet that is 50 times the value of the bet made by the gambler. In this case, if the gambler bets $1 and the Canadiens win the cup, the gambler would profit by $50. Otherwise, the gambler would lose their $1 to the bookmaker.

In a fair bet, the gambling odds also reflect the probability of the event being wagered on. This means that one can translate betting odds into probabilities, and vice versa. Equation 3-15 provides the expression to calculate the probability of an event *x* occurring from the odds against *x*, represented as *O(x)*. For the hockey example, if one substitutes the fraction 50/1 for *O(x)* then Equation 3-15 indicates that *P(x)*—the probability of winning the cup—is approximately .0196. It is because this probability is so small that the bookmaker is willing to wager such a large amount against the gambler's much smaller bet.

$$P(x) = \frac{1}{O(x) + 1} \tag{3-15}$$

Like Bayesian probabilities, betting odds are constantly updated as circumstances change. Consider another hockey example. In 2015, the Montreal Canadiens started their season with a nine-game winning streak, and their odds of winning the championship had climbed as high as 15:2. However, their world-class goaltender Carey Price was injured in a December game against the New York Rangers. By February, their odds of winning the cup fell to 28:1, reflecting the impact of this event on bookmarkers' views of Montreal's future.

Quantitatively speaking, what was the perceived effect of Price's injury on Montreal's fate? One approach for measuring this effect is to compute the odds ratio (Rudas, 1998). The odds ratio is literally a ratio between the odds in one situation and the odds in another situation. For example, to calculate the odds ratio in the context of Carey Price's health, one divides the fraction 15/2 (the odds against Montreal winning the cup with Price healthy) by the fraction 28/1 (the odds against Montreal winning the cup with Price injured). The resulting odds ratio is approximately 0.268. This indicates that Price's status had a huge effect on bookmakers' view of the team. This is because the further that the odds ratio departs from a value of 1, the greater is the effect of the event being considered. If the presence or absence of an event has no effect, then the odds ratio calculated for it will be close to or equal to 1.

This discussion indicates that the odds ratio measures the effect of some situation being true in contrast to the situation in which it is false. From this perspective, it is not surprising to discover that the odds ratio is found at the heart of various procedures that predict the likelihood of an event occurring from a set of predictors, each of which may be either present or absent.

One such procedure is *logistic regression* (Christensen, 1997; Cramer, 2003; Hosmer & Lemeshow, 2000). Logistic regression predicts an outcome by summing the signals associated with a set of predictors, where each signal indicates whether some predictor is present. These signals are each weighted by coefficients, summed together, and then transformed into a probability using the logistic equation. Each coefficient indicates the effect on the outcome of each predictor, assuming that the effects of the other predictors are held constant. Important to note, in logistic regression each coefficient is equal to the natural logarithm of the odds ratio associated with a predictor.

If the coefficients in logistic regression are equal to natural logarithms of odds ratios, then it is not surprising that the connection weight in the modern perceptron that we are exploring in this chapter is also equal to a natural logarithm of the odds ratio. This is because the end result of fitting logistic regression to data is functionally equivalent to the end result of training a modern perceptron on the same data (Schumacher, Rossner, & Vach, 1996). In a recent study, where four predictor cues were combined to determine the probability of reward, it was found that the weights and bias of a modern perceptrons trained on this data were almost identical to the coefficients and constant of logistic regression equations fit to the same data (Dawson & Gupta, 2017).

To end this chapter, let us return to contingency theory that was introduced in Section 3.2. What might discovering the odds ratio in the perceptron's structure mean for psychological contingency theory?

A key issue in the literature on learning contingencies is the debate about what the appropriate metric of contingency or association might be (Allan, 1980; Allan & Jenkins, 1980, 1983). Allan and Jenkins developed five contingency measures for a 2×2 contingency table like Table 2 and argued that the one that provides the best account of human data is the difference in conditional probabilities is defined as ΔP (Equation 3-1), which we saw earlier is equal to $P(H|E)$ - $P(H|{\sim}E)$.

There is one important metric that Allan and Jenkins do *not* consider—the odds ratio. This is surprising because we have seen that the odds ratio is the preferred measure of association for statisticians when they analyze contingency tables. Discovering the odds ratio in the connection weight of the modern perceptron—whose behavior can deliver ΔP (Dawson & Dupuis, 2012)—points to a new metric of contingency worthy of consideration in the psychological literature.

What is the relationship between ΔP and the natural logarithm of the odds ratio? Let us start to answer this question empirically. Wasserman et al. (1990) reported the results of contingency experiments that involve using 25 different 2×2 contingency tables. Using these tables, they discovered that ΔP was highly correlated with human contingency ratings. We took their 25 contingency tables, which had values of ΔP that ranged from 0.60 to −6.0, and computed the natural logarithm of the odds ratio for each one. For this sample of contingency data, the correlation between ΔP and $ln(ad/bc)$ was essentially perfect (r = .999). Given this relationship between the two metrics, it is no surprise that the natural logarithm of the odds ratio generated equally high correlations with the human judgments of contingency that Wasserman et al. report (r = .94 for one reported study, r = .93 for another).

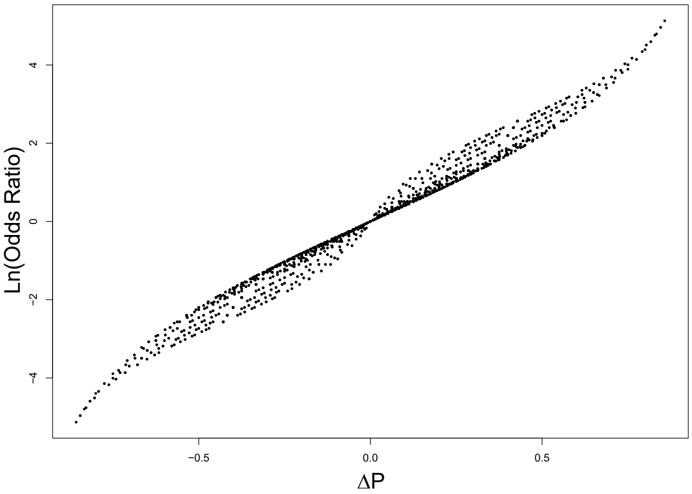


Figure 3-1. A scatterplot of the relationship between ΔP and the ln(odds ratio) for 2,401 different 2X2 contingency tables created to generate a range of ΔP values. See text for details.

To explore this sort of relationship further, a larger set of contingency tables can be created. This was done by taking seven possible values for a cell frequency in a 2×2 contingency table: 5, 15, 25, 35, 45, 55, and 65. We used assigned the values to a, b, c, and d in Table 2, using all possible combinations, to create a set of 2,401 (i.e., 7×7×7×7) contingency tables. For this large set of tables ΔP ranged from about 0.857 to −0.857. The correlation between ΔP and the *ln*(odds ratio) for this set of tables was again very large, being equal to 0.992.

Figure 3-1 provides the scatter plot of each of the ΔP and ln(odds ratio) pairs for this set of 2,401 contingency tables. This graph, which is a tight propeller-shaped cloud, clearly illustrates the strong correlation between the two variables. It also indicates that one cannot write a function that translates ΔP into the *ln*(odds ratio). This is because a single value of ΔP maps onto more than one value of *ln*(odds ratio), and vice versa. This property spreads the scatter plot into its propeller shape, instead of a function that can be graphed with a continuous line.

Figure 3-1 shows that there is a very strong relationship between ΔP and the *ln*(odds ratio). Given this relationship, are there any reasons for replacing the former with the latter in psychological studies of human contingency judgments? There are at least three reasons for replacing ΔP with a metric based on the odds ratio.

First, ΔP (Equation 3-1), as we saw in our earlier discussion of Bayes's theorem, is only a computational-level description of behavior. For instance, an agent whose contingency judgments are consistent with ΔP can be described as being sensitive to this statistic. However, this sensitivity does not require the agent to represent mentally a table like Table 3-1, nor does it require the agent to know Equation 3-1 to make these consistent judgments. We are in a position to say this because the perceptron is a device that behaves *as if* it computes ΔP (Dawson & Dupuis, 2012) but does not represent or calculate Equation 3-1, nor does it represent a contingency table. Instead, it uses a psychologically plausible learning rule (Dawson, 2008) to update the association between an input signal and an output response. Computationally speaking, this associative mechanism is functionally equivalent to representing contingency as ΔP. However, in terms of algorithmic or implementational properties, ΔP is not present in it. If we consider the perceptron as a plausible mechanism for representing contingency, then we need to consider describing contingency in terms of properties that are directly represented in this mechanism, that is, in terms of the odds ratio.

Second, there are important technical limitations on the definition of ΔP. In particular, ΔP cannot be computed if both *a* and *b*, or if both *c* and *d*, equal 0. This is because this produces an undefined fraction because its numerator is divided by 0. Of course, the proof that relates perceptron properties to Bayes's theorem also requires that contingency table frequencies (in particular *b* and *c* are greater than zero). If this is not the case, then fractions in Equations 3-13 and 3-14 are undefined. Important to note, this is a limitation of the proof and not of the capabilities of the modern perceptron. For example, we can build training sets from the cancer example (Table 3-3) that set two of the four frequencies to 0. This is done by training the perceptron on a subset of problems—only those patterns that belong to one row of Table 3-3, or only those patterns that belong to one column of Table 3-3. Perceptrons learn to estimate the probabilities of these subsets, even though none of them permits ΔP to be defined. Again, the algorithmic properties of the associative mechanism trump the computational definition of Equation 3-1.

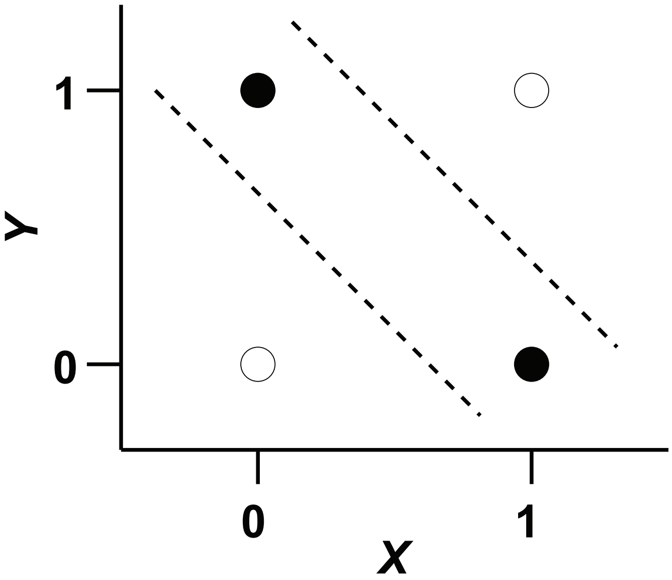
Third, ΔP is defined for a 2×2 contingency table only. Clearly if we are interested in understanding human contingency judgments, then we must consider situations that can only be captured by higher dimensional versions of Tables 3-1 or 3-2. These more complex situations involve inferring the probability of a hypothesis using multiple sources of evidence. As these situations cannot be defined in a 2×2 table, one cannot use Equation 3-1 to calculate ΔP for them. A more complex notion of differences between conditional probabilities is required. One example is logistic regression (Christensen, 1997; Cramer, 2003; Hosmer & Lemeshow, 2000), which employs odds ratios to quantify the effect of a predictor under the assumption that all other predictors are held constant. A second, and related, example is using a more flexible notion of conditional probability to explore human reasoning under uncertainty (P. W. Cheng, 1997; P. W. Cheng & Holyoak, 1995; P. W. Cheng & Novick, 1990, 1992).

A third example is the modern perceptron. The current chapter examined the relationship between the simplest perceptron—one that has only one input—and the simplest version of Bayes's theorem. This analysis revealed a formal relationship between this perceptron and Bayes's rule when only one source of evidence is required. Of course, a modern perceptron is not limited to a single input unit. Given this, and given the established relationship between perceptrons and logistic regression (Dawson & Gupta, 2017; Schumacher et al., 1996), we should establish formal relations between more complex perceptrons and alternative versions of Bayes's rule. The next chapter establishes this fact and at the same time reveals some interesting limitations on modern perceptrons when faced with these more complex situations. In later chapters, we are particularly interested in determining whether human adaptation to uncertain environments is subject to similar limitations.

Chapter 4: Perceptrons Are Naïve Bayesians

Chapter 3 established the relationship between perceptron probability estimation and Bayes's theorem when a single cue predicts the probability of reward. The current chapter extends these results to the situation in which more than one cue signals reward. It begins by considering the limitations of perceptrons in terms of a logical property, linear nonseparability. It then introduces an alternative notion of limitation that is related to interactions between different cues. A series of simulations in which perceptrons are trained on Boolean operators whose truth is uncertain indicates that the notion of interaction is more relevant than is the notion of linear nonseparability. The chapter then presents a formal analysis of the probability estimation of perceptrons when two cues is presented, using techniques very similar to those described in Chapter 3. This analysis proves that when multiple cues are involved, the probability estimates generated by perceptrons are defined by the naïve Bayes equation. This is a version of Bayes's theorem that assumes that there are no interactions between the signals provided by different cues and reveals a particular limitation of the ability of perceptrons to estimate probabilities.

Figure 4-1. The pattern space for XOR. Each dot in the space represents one of the four possible stimuli, which are X, Y pairs. If a dot is black, the correct response to the stimulus is 1; if a dot is black, the correct response is 0. The dashed lines indicate how the pattern space requires two "cuts" to separate all the "on" patterns from all the "off" patterns. See text for details.



4.1 The Limits of Perceptrons

4.1.1 Old Connectionism

In general, an artificial neural network implements a mapping from inputs to outputs. The perceptrons introduced in earlier chapters are particularly simple examples of artificial neural networks because they consist of only input and output units. Networks that are more complex have intermediate layers of processors, called hidden units, that detect complex interactions between input signals before passing information about these interactions on to output units.

Because of this simplicity, a perceptron is an example of what is called Old Connectionism (Medler, 1998). Old Connectionism developed two kinds of artificial neural networks. The first were very powerful networks because they contained many intermediate layers of processors (McCulloch & Pitts, 1943). However, the connection weights of these networks had to be defined by hand: These networks did *not* learn. The second were simple networks that could be trained; however, their simplicity placed constraints on what could be learned. Perceptrons (Rosenblatt, 1958, 1962) provide a prototypical example of this second type of Old Connectionism.

4.1.2 Linear Separability

A detailed formal analysis of the limits of perceptrons (Minsky & Papert, 1969) is famous for leading to decreased interest in artificial neural networks during the cognitive revolution (Papert, 1988). Figure 4-1 illustrates a key limitation using the *exclusive or* (XOR) operation, which is examined in more detail later in this chapter. This logical operation is computed over two inputs, X and Y, which can each have values of either 0 or 1. XOR(X, Y) returns a value of 1 only when one of these inputs has a value of 1 and the other has a value of 0. If both inputs equal 0, or if both inputs equal 1, then XOR(X, Y) returns a value of 0. Figure 4-1 illustrates this set of desired responses by representing the four possible stimuli for XOR in a *pattern space*. In this pattern space, each input pattern is represented as a point at a particular location in space. This particular space is two-dimensional, because each dimension is used to represent the possible values of an input variable, and XOR is defined using two variables. The four dots represent the four possible patterns (i.e., X, Y pairs) that can be presented to the network. The x-axis provides the possible values of X, and the y-axis provides the possible values of Y. The color of each dot in Figure 4-1 indicates the desired response of the XOR operation to that particular stimulus. If a dot is black,

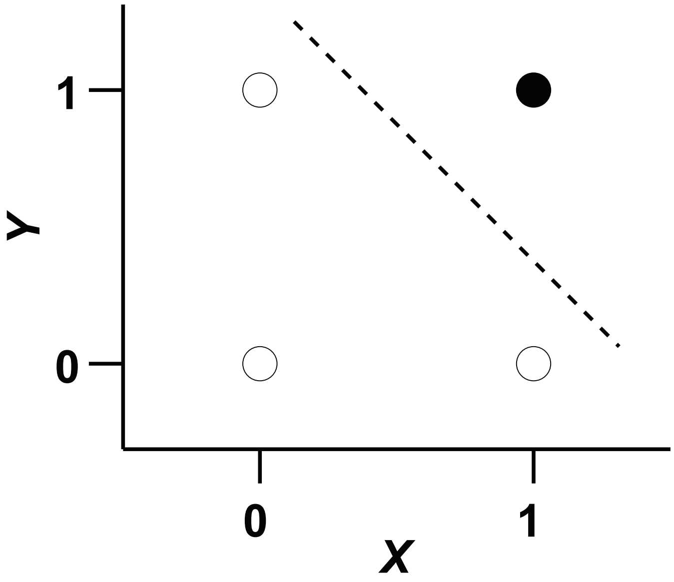


Figure 4-2. The pattern space for AND. Each dot in the space represents one of the four possible stimuli, which are X, Y pairs. If a dot is black, the correct response to the stimulus is 1; if a dot is black, the correct response is 0. The dashed line indicate how the pattern space requires only one straight "cut" to separate all the "on" patterns from all the "off" patterns. This proves that this is a linearly separable problem.

then the desired response to the stimulus is 1; if a dot is white, then the desired response is 0.

A pattern space illustrates how pattern recognition problems are to be solved. In general, to solve a problem a system must partition the pattern space into separate regions. Each region can contain only patterns that lead to the identical response. The two dashed lines in Figure 4-1 illustrate this type of solution to XOR. They carve the pattern space into three distinct regions. The region in the middle of the figure contains only the two patterns that require a response of 1. Each of the other two regions contains a single pattern that requires a response of 0. In general, an artificial neural network can be described as a system that solves pattern recognition problems by discovering the correct manner in which to carve a pattern space into the appropriate decision regions (Lippmann, 1989).

Figure 4-1 shows that XOR is a *linearly nonseparable problem*. This is because *two* cuts through the pattern space are required to separate all the "on" patterns from all the "off" patterns. That two cuts are required to partition this pattern space shows that XOR is *not* linearly separable. A problem is linearly separable only when a *single* straight cut through the pattern space separates all the "on" stimuli from all the "off" stimuli.

Figure 4-2 provides an example of a different pattern space, one that is linearly separable. This pattern space is for another logical operation AND(*X, Y*), which is true only when both X and Y have values of 1, and which is false otherwise. For a perceptron to solve this problem, it must carve the pattern space in such a way that the pattern that requires a response of 1 is in a different region than any of the other three patterns that require a response of 0. A single straight cut, illustrated by the dashed line in Figure 4-2, accomplishes this, showing that AND is a linearly separable problem.

The standard critique of perceptrons is that they are limited to computing linearly separable mappings between stimuli and responses (Minsky & Papert, 1969). Thus a perceptron can learn to compute the linearly separable operator AND(*X, Y*) but cannot learn to compute the linearly nonseparable operator XOR(*X, Y*). The reason for this limitation is that the typical activation function in a perceptron's output (i.e., either the threshold function or the logistic equation) can make a single straight cut only through a pattern space, and thus a perceptron's output units are constrained to solve linearly separable problems.

The connectionist revolution in the 1980s began when researchers discovered new learning rules, such as backpropagation of error (Rumelhart et al., 1986). These new learning rules permitted networks with an intermediate layer of hidden units to be trained (see the earlier discussion in Section 1.7). Hidden units can learn to detect complex higher order features and thus enable networks to learn complex (and linearly nonseparable) input/output mappings. In recent years, researchers have discovered new learning rules that train extremely powerful systems called *deep belief networks* (Hinton, 2007; Hinton et al., 2006; Mohamed, Dahl, & Hinton, 2012; Sarikaya, Hinton, & Deoras, 2014). These networks use many intermediate layers of hidden units; they have launched a modern revival of connectionist artificial intelligence research.

4.1.3 Beyond Linear Separability

This chapter explores the ability of perceptrons to estimate probabilities using more than one source of evidence. We prove that in this situation, perceptrons are still Bayesian mechanisms but implement a version of Bayes's theorem that imposes limits on probability estimation that are different than limitations related to linear separability. With this proof in hand, we are confronted with a particular issue: Is there a reason to view these limited Bayesian mechanisms as being of psychological interest, or should we instead explore more modern networks that include hidden units? To preview the answer to this question that is provided by the results of some psychological experiments described in Chapter 7, there is reason to believe that human probability matching faces similar limitations to those described formally in the current chapter.

Historically, the linear separability of problems is the only limitation of perceptrons that is considered. Nevertheless, in the context of the current monograph, the definition of linear separability requires assuming that the world is certain. That is, this definition assumes that each stimulus is associated with only one (the correct) response. That is, in Figures 4-1 and 4-2, each stimulus dot is mapped onto a single color. In addition, linear separability is itself used to make a definite or certain declaration—the assertion that a particular problem can or cannot be solved by a perceptron.

As soon as we recognize the uncertainty of the world, we discover that there are other characteristics for defining perceptron limitations. For example, in this chapter we show that the ability of perceptrons to estimate probabilities is strongly affected by whether two cues interact. Figure 4-1 demonstrated that XOR is linearly nonseparable. However, from a different perspective XOR also demonstrates a statistical interaction between the inputs *X* and *Y*. We see next that when perceptrons estimate probabilities using multiple cues, perceptrons are limited in the sense that they are blind to interactions between predictors.

Figure 4-3 illustrates the general sense of an interaction with which we are concerned. It also depicts responses to XOR(*X, Y*), but in a different fashion than was used in Figure 4-1. In this new figure, the *x*-axis of both graphs provides the values of *X*, and the *y*-axis of both graphs provides the response of the XOR operator. The graph on the left of Figure 4-3 illustrates the effect of *X* on the response when the value of *Y* is 0, whereas the graph on the right of Figure 4-3 illustrates the effect of *X* on the response when the value of *Y* is 1. The dashed line in each graph simply highlights the effect of changing the value *X* in each graph. In the graph on the left, changing the value of *X* from 0 to 1 increased the XOR response. This is reflected in the positive slope of the dashed line. In contrast, in the graph on the right, a change in the value of *X* from 0 to 1 decreases the XOR response. This is reflected in the negative slope of the dashed line. The two different-sloped lines indicate the presence of an interaction, because the effect of *X* depends on (or interacts with) the value of *Y*.

This notion of interaction is important for two reasons. First, an interaction is distinct from linear nonseparability. For example, Figure 4-1 illustrates the point that XOR is linearly nonseparable, whereas Figure 4-3 shows that it also exhibits an interaction. However, one can also find an interaction in a linearly separable problem. This is illustrated in Figure 4-4, which is similar in structure to Figure 4-3 but which illustrates the nature of the AND operation. Even though AND is a linearly separable operation (Figure 4-2), Figure 4-4 still reveals an interaction. This is because the different slopes of the dashed line in each of the two graphs in Figure 4-4 indicates that the effect of *X* on the response depends on (or interacts with) the value of *Y*.

The notion of interaction becomes central when perceptrons are trained on uncertain problems. For example, in Section 4.2 we study the performance of perceptrons on a variety of logical problems like XOR and AND that are defined probabilistically or contingently. This means that instead of being true for some stimuli all of the time, and false for some other stimuli all of the time, the perceptron learns that some stimuli are true only 70% of the time and that other stimuli are false only 70% of the time. The best that these perceptrons can do for these contingent problems is estimate probabilities. However, we see that perceptrons have difficulty doing this for some problems. Furthermore, these difficulties appear for both linearly

Figure 4-3. The interaction between cues X and Y for the XOR problem. The graph on the left shows how changes in X affect the response when Y = 0, and the graph on the right shows how changes in X affect the response when Y = 1. The fact that the dashed lines in the two graphs have different slopes reveals an interaction, because the effect of X depends on the value of y.

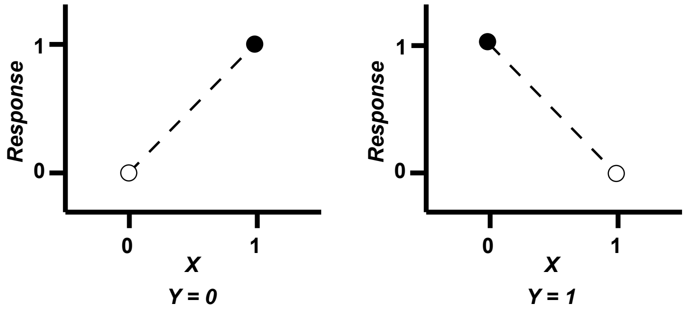
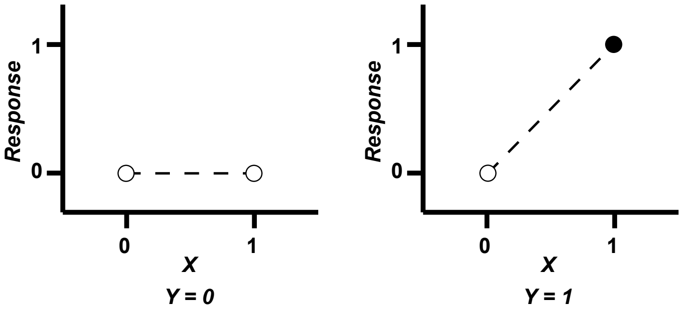


Figure 4-4. The interaction between cues X and Y for the AND problem. The graph on the left shows how changes in X affect the response when Y = 0, and the graph on the right shows how changes in X affect the response when Y = 1. The fact that the dashed lines in the two graphs have different slopes reveals the interaction, because the effect of X depends on the value of y.



nonseparable and linearly separable problems. This is because linear separability is not the only limitation faced when perceptrons estimate probabilities using multiple cues. Perceptrons also have difficulties when a problem involves interactions between cues.

4.2 Learning Probable Boolean Operations

4.2.1 Boolean Algebra

Perceptrons are limited in what they can learn to do because they do not include hidden units. How might this limitation affect their ability to make probability judgments when more than one source of evidence is under consideration? Let us explore this issue empirically by training perceptrons to make judgments that are based on two sources of evidence.

George Boole invented mathematical logic in the late 19th century (Boole, 1854/2003). Part of Boole's formalism considered logical operations defined over two variables, where each of these two variables could be either true or false, and the result of the logical operation was also either true or false. Boole's original logic was the basis of Claude Shannon's modern insight that electric circuits could be described logically (Shannon, 1938).

Table 4-1 provides the 16 basic operations that define modern Boolean algebra. In this table, the two variables over which a logical operation is defined are *X* and *Y*. If one of these variables is false, it is represented with the value 0; if it is true, it is represented with the value 1. With this representation, there are four possible combinations of variables for any logical operation. The results of applying a particular operation to each of these four combinations are also provided in Table 4-1. For example, Contradiction(*X, Y*) returns a value of 0 (it is false) for each of the four possible combinations of the two input variables. Similarly AND(*X, Y*), returns a 1 (it is true) if both *X* and *Y* are true, and returns a 0 for each of the other three input combinations.

The 16 Boolean operators provided in Table 4-1 were central to the invention of the logical neuron (McCulloch & Pitts, 1943). McCulloch and Pitts designed logical neurons—each of which was a combination of two input weights and a single threshold—for 14 of these 16 Boolean operators. Each of the McCulloch–Pitts neurons were digital, in the sense that it was assumed that each of their inputs could only have values of 0 and 1 and that a neuron would only output a value of 0 or 1. The only two operators that could not be computed by a McCulloch–Pitts neuron were XOR(*X, Y*) and its negation ~XOR(*X, Y*). Of all of the Boolean operators, these are the only two that are linearly nonseparable. As a McCulloch–Pitts neuron is a variant of a perceptron, and is therefore limited to solving linearly separable problems, it cannot represent a solution to linearly nonseparable problems like XOR or ~XOR.

Table 4-1. The 16 logical operations of modern Boolean algebra.

Operation	X	Y	Result	Operation	X	Y	Result
Contradiction	0	0	0	**AND(*X, Y*)**	0	0	0
	0	1	0		0	1	0
	1	0	0		1	0	0
	1	1	0		1	1	1
AND(~X, ~Y	0	0	1	**~XOR(*x, y*)**	0	0	1
	0	1	0		0	1	0
	1	0	0		1	0	0
	1	1	0		1	1	1
AND(~*X, Y*)	0	0	0	***Y***	0	0	0
	0	1	1		0	1	1
	1	0	0		1	0	0
	1	1	0		1	1	1
~X	0	0	1	***X* ⊃ *Y***	0	0	1
	0	1	1		0	1	1
	1	0	0		1	0	0
	1	1	0		1	1	1
AND(~*X, Y*)	0	0	0	**X**	0	0	0
	0	1	0		0	1	0
	1	0	1		1	0	1
	1	1	0		1	1	1
~Y	0	0	1	***Y* ⊃ *X***	0	0	1
	0	1	0		0	1	0
	1	0	1		1	0	1
	1	1	0		1	1	1
XOR(*X, Y*)	0	0	0	***OR*(*X, Y*)**	0	0	0
	0	1	1		0	1	1
	1	0	1		1	0	1
	1	1	0		1	1	1
~AND(*X, Y*)	0	0	1	**Tautology**	0	0	1
	0	1	1		0	1	1
	1	0	1		1	0	1
	1	1	0		1	1	1

Note. Each operation is described as a function of two variables, X and Y, which are "true" when their value is 1 and "false" when their value is 0. The result is the value of the operator given the values of X and Y, which is either 1 (true) or 0 (false). Each operator is defined using the same four X, Y pairs, but each operator differs from the others in terms of its results when considered across the four input patterns.

4.2.2 Learning an Uncertain Boolean Algebra

In this monograph, our interest is in the probabilistic properties of devices like perceptrons. How do the limitations of perceptrons translate into a domain where the mapping from inputs to response is uncertain? To explore this question, we created probabilistic versions of each

Table 4-2. The responses of 16 perceptrons trained on probabilistic versions of the logical operations of modern Boolean algebra.

Operator	X	Y	Probability of Reward	Perceptron	Operator	X	Y	Probability of Reward	Perceptron
Contradiction	0	0	0.30	0.30	**AND(~X, ~Y**	0	0	0.70	0.62
	0	1	0.30	0.30		0	1	0.30	0.38
	1	0	0.30	0.30		1	0	0.30	0.38
	1	1	0.30	0.30		1	1	0.30	0.18
~X	0	0	0.70	0.70	**AND(~*X*, *Y*)**	0	0	0.30	0.38
	0	1	0.70	0.70		0	1	0.70	0.63
	1	0	0.30	0.30		1	0	0.30	0.18
	1	1	0.30	0.30		1	1	0.30	0.38
~Y	0	0	0.70	0.70	**AND(*X*, ~*Y*)**	0	0	0.30	0.38
	0	1	0.30	0.30		0	1	0.30	0.18
	1	0	0.70	0.70		1	0	0.70	0.62
	1	1	0.30	0.30		1	1	0.30	0.38
Y	0	0	0.30	0.30	**XOR(*X*, *Y*)**	0	0	0.30	0.50
	0	1	0.70	0.70		0	1	0.70	0.50
	1	0	0.30	0.30		1	0	0.70	0.50
	1	1	0.70	0.70		1	1	0.30	0.50
X	0	0	0.30	0.30	**~AND(*X*, *Y*)**	0	0	0.70	0.82
	0	1	0.30	0.30		0	1	0.70	0.63
	1	0	0.70	0.70		1	0	0.70	0.62
	1	1	0.70	0.70		1	1	0.30	0.38
Tautology	0	0	0.70	0.70	**AND(*X*, *Y*)**	0	0	0.30	0.18
	0	1	0.70	0.70		0	1	0.30	0.38
	1	0	0.70	0.70		1	0	0.30	0.38
	1	1	0.70	0.70		1	1	0.70	0.63
					~XOR(*X*, *Y*)	0	0	0.70	0.50
						0	1	0.30	0.50
						1	0	0.30	0.50
						1	1	0.70	0.50
					X* ⊃ *Y	0	0	0.70	0.62
						0	1	0.70	0.82
						1	0	0.30	0.38
						1	1	0.70	0.63
					Y* ⊃ *X	0	0	0.70	0.62
						0	1	0.30	0.38
						1	0	0.70	0.82
						1	1	0.70	0.63
					OR(*X*, *Y*)	0	0	0.30	0.38
						0	1	0.70	0.63
						1	0	0.70	0.62
						1	1	0.70	0.82

Note. For each stimulus, the known probability of reward is provided alongside the response of the perceptron after 1,000 epochs of training on the problem.

of the Table 4-1 operators. In our training set, "true" was operationalized as "true with a probability of 0.70," and "false" was operationalized as "false with a probability of 0.70." That is, a true stimulus is one that has a likelihood of reward of 0.70, whereas a false stimulus is one that has a likelihood of reward of 0.30.

We constructed training sets for each operator that consisted of 40 stimulus patterns instead of four patterns as follows: We duplicated each of the four different stimuli (i.e., each *X, Y* pair) 10 times. Then we mapped the inputs onto a desired response contingently. For instance, for a pair of inputs that were "70% true," the perceptron was rewarded (provided a desired response of 1) for seven of these instances, whereas the other three of these instances were not rewarded (were paired with a desired output of 0). Similarly, for a pair of inputs that were "70% false," the perceptron was not rewarded for seven of these stimulus pairs, whereas it was rewarded for the remaining three pairs. If a modern perceptron is able to respond correctly to such a contingent problem, then it should estimate the problem's probabilities. That is, after training is complete it should generate a response of 0.70 to inputs that are true in Table 4-1, and it should generate a response of 0.30 to inputs that are false in Table 4-1.

We trained modern perceptrons on each of the 16 training sets for contingent versions of the Boolean operators using the gradient descent rule (Dawson, 2005). For each perceptron, the two connection weights were randomly initialized to values from the range between −0.1 and 0.1, whereas the perceptron's bias was initially set to 0. The learning rate was 0.1, and each perceptron was trained for 1,000 epochs. With this amount of training, the perceptrons had achieved an equilibrium state of the type described in Chapter 2. Table 4-2 provides the response of a trained perceptron to each of the possible pairs of inputs for each of these contingent logical operations.

An examination of Table 4-2 indicates that there is a marked difference between the behavior of perceptrons faced with logical versions of the operators and the behavior of perceptrons trained on contingent versions of the same operators. Recall that McCulloch–Pitts neurons are capable of representing 14 of the 16 possible Boolean operators. Modern perceptrons can also learn to generate correct responses to the same 14 Boolean operators (Dawson, 2005). However, Table 4-2 reveals that perceptrons generate correct responses—that is, perceptrons estimate probabilities correctly—for only six of the 16 operators when they are defined probabilistically. The six operators that lead to correct performance are listed on the left side of Table 4-2. The remaining 10 operators on the right side of Table 4-2 present behavior in which perceptrons fail to correctly estimate the probabilities of the training sets. Although the perceptrons do a reasonable job of estimating probabilities for these problems—the correlation between observed perceptron responses and expected responses is 0.798—it is not perfect. In contrast, the correlation between perceptron responses and the expected probabilities for the six problems on the left of Table 4-2 is perfect.

Table 4-2 indicates that a factor different from linear separability is at play when perceptrons attempt to estimate the probabilities of contingent versions of the Boolean operators. This is because eight of the problematic operators on the left of Table 4-2 are, logically speaking, linearly separable. This is not to say that linear separability is not relevant. For instance, an examination of the right side of Table 2 indicates that the probability estimation performance of perceptrons for the two linearly nonseparable problems (XOR, ~XOR) is poorer than the performance for the remaining problems on that side of the table. Nevertheless, it is obvious that some factor other than linear separability is playing an important role.

What might this factor be? Consistent with the discussion from Section 4.1, all of the problems on the right side of Table 4-2 involve some sort of interaction between the two input signals. That is, correct probability estimation behavior requires that a perceptron be sensitive to an interaction between the *X* and *Y* signals. Remember that such an interaction can be present for a linearly separable problem like AND(*X, Y*), as we saw earlier in the discussion of Figure 4-4. In the next section, we conduct a formal analysis of modern perceptrons that are faced with evaluating hypotheses based on two sources of evidence. This analysis provides a precise explanation for why perceptrons fail to estimate probabilities for some probabilistic logical problems, even when logical versions of these problems are linearly separable.

4.3 Conditional Independence, Naïve Bayes, and Perceptrons

4.3.1 Measuring Conditional Dependence

To this point in this chapter, we have simply been noting that perceptrons are limited in the sense that they do not detect interactions between cues. Let us now be a bit more precise by defining the notion of the interaction between cues in the context of probability theory.

In probability theory, two events are said to be independent if the probability that one occurs is completely unaffected by the occurrence of the other. For instance, the successive tosses of a coin are independent because the result of one toss has no effect whatsoever on the result of the next toss. Mathematically this can be expressed by the relation provided in Equation 4-1. This equation states that the probability of event *X* occurring given that event *Y*—that is, *P*(*X*|*Y*)—has occurred is simply equal to the probability of event *X* occurring. This is only true if the two events are independent—if *Y* has no effect on *X*.

$$P(x) = P(X|Y) = P(X) \tag{4-1}$$

Another consequence of the independence of the two events *X* and *Y* concerns determining the probability that both events occur together—that is, *P*(*X* ∩ *Y*). With two independent events, the probability of their co-occurrence is simply the probability of one event multiplied by the probability of the other as is defined in Equation 4-2:

$$P(X \cap Y) = P(X) \cdot P(Y) \tag{4-2}$$

Similar relations can be defined for conditional probabilities. For instance, consider two conditional probabilities: one the probability that *X* occurs given that *H* has occurred (*P*(*X*|*H*) and the other the probability that *Y* occurs given that *H* has occurred (*P*(*Y*|*H*). What is the probability that both *X* and *Y* occur given that *H* has occurred (*P*(*X* ∩ *Y*|*H*))? If *X*|*H* and *Y*|*H* are conditionally independent of one another, then this probability is defined by a variation of Equation 4-2 that is provided as Equation 4-3:

$$P(X \cap Y|H) = P(X|H) \cdot P(Y|H) \tag{4-3}$$

To say that two events are independent—that they do not interact—is to assert that equations like the three just provided are true. If these equations are not true, then the two events are not independent. However, it is important to realize that when the two events are not independent, there are varying degrees of dependence that are possible. For instance, consider Equation 4-1. If this equation is not true, then *X* and *Y* are dependent, because *P*(*X*|*Y*) and *P*(*X*) have different values. However, the degree of this dependence will be reflected in the size of the difference. For instance, if the difference between *P*(*X*|*Y*) and *P*(*X*) is small, then we might say that *X* and *Y* are slightly dependent. However, if as the difference between *P*(*X*|*Y*) and *P*(*X*) becomes larger and larger, then we recognize that the dependence between *X* and *Y* is larger and larger. It will be convenient to quantify this notion of "degree of dependence," particularly because we see that as an increase in this quantity decreases the ability of perceptrons to estimate probabilities decreases.

Let us introduce a measure of conditional independence that can be applied to the probabilistic Boolean operators that were introduced in Section 4.2.2. Each of these operators involves two signals (*X* and *Y*) that can either be present or absent. Each combination of these signals is presented several times. For instance, in Section 4.2.2 each combination of signals was duplicated 10 times. For some of these presentations, the combination was rewarded, but for the others it was not. The structure of this task can be represented by a 2×2×2 contingency table like the one presented in Table 4-3. This table represents the number of times that each combination of signals was rewarded, as well as the number of times that each combination was not rewarded.

Table 4-3. General form of a 2×2×2 contingency table for two signals (X, Y) that can lead to a reward (R).

	R				**~R**		
	Y	**~Y**	**Sum**		**Y**	**~Y**	**Sum**
X	*a*	*b*	*a+b*	**X**	*e*	*f*	*e+f*
~X	*c*	*d*	*c+d*	**~X**	*g*	*h*	*g+h*
Sum	*a+c*	*b+d*	*a+b+c+d*	**Sum**	*e+g*	*f+h*	*e+f+g+h*

Note. Each lowercase letter in a cell stands for a frequency. For instance, a is the number of times that there is a reward when X and Y are both true, whereas e is the number of times that there is no reward when X and Y are both true.

The information that is provided in a contingency table such as Table 4-3 can be used to measure the degree of dependence between the signals *X* and *Y* by computing log likelihood ratios (Woolf, 1957). Let *O* be the observed frequency of a result (i.e., one of the frequency values in Table 4-3), and let *E* be the expected frequency of this result under the assumption that signals are independent. For example, in Table 4-3, *a* is the observed number of times that there is a reward when *X* and *Y* are both present. The expected value for this cell is the sum of its row in the table (*a*+*b*) times the sum of its column of the table (*a*+*c*) divided by the total number of frequencies in the table (*a*+*b*+*c*+*d*). That is, for this cell *E* = (*a*+*b*)(*a*+*c*)/(*a*+*b*+*c*+*d*). For the situation in which there is no reward when *X* and *Y* are both present, a different cell in the contingency table is considered. In this case, *O* = *e* and *E* = (*e*+*f*)(*e*+*g*)/(*e*+*f*+*g*+*h*).

The log-likelihood ratio is for a particular cell is *E* • (*ln*(*O*) – *ln*(*E*)). Eight likelihood ratios can be calculated for Table 4-3. If these eight values are summed together, and this sum is multiplied by 2, the result is the called the criterion G (Woolf, 1957). That is, $G = 2\bullet\Sigma O(\ln(O) - \ln(E))$. When X and Y are completely independent, G will have a value of zero. As X and Y become more and more dependent, the value of G increases.

To show this, let us demonstrate how to calculate G for three of the different probabilistic Boolean operators from Section 4.2.2. To begin, Table 4-4 provides the contingency table for probabilistic operator X. In Table 4-2, it was shown that a perceptron could learn to estimate perfectly the probabilities of reward for this problem because X and Y are independent in this problem. When G is calculated from Table 4-4, its value is 0, confirming the independence of the two signals.

Table 4-4. The contingency table for the probabilistic Boolean operator X.

	R				~R		
	Y	**~Y**	**Sum**		**Y**	**~Y**	**Sum**
X	7	7	14	**X**	3	3	6
~X	3	3	6	**~X**	7	7	14
Sum	10	10	20	**Sum**	10	10	20

Next, Table 4-5 provides the contingency table for probabilistic operator AND(X,Y). In Table 4-2, it was shown that a perceptron could not learn to estimate perfectly the probabilities of reward for this problem. This is because X and Y are not independent in this problem. When G is calculated from Table 4-5, its value is 1.61. The fact that this value is greater than 0 indicates that there is dependence between the two signals.

Table 4-5. The contingency table for the probabilistic Boolean operator AND(X,Y).

	R				~R		
	Y	**~Y**	**Sum**		**Y**	**~Y**	**Sum**
X	3	3	14	**X**	7	7	14
~X	3	7	6	**~X**	7	3	20
Sum	6	10	20	**Sum**	14	10	24

Finally, Table 4-6 provides the contingency table for probabilistic operator XOR(X,Y). In Table 4-2, it was shown that a perceptron could not learn to estimate the probabilities of reward for this problem because X and Y are not independent in this problem. When G is calculated from Table 4-6, its value is 6.58. This value is greater than 0, which indicates that there is dependence between the two signals. In addition, G is larger for Table 4-6 than it is for Table 4-5, which indicates that there is more dependence between X and Y for XOR than is the case for AND. This in turn explains why perceptron probability estimates are more accurate for AND than for XOR in Table 4-2.

Table 4-6. The contingency table for the probabilistic Boolean operator XOR(X,Y).

	R				~R		
	Y	**~Y**	**Sum**		**Y**	**~Y**	**Sum**
X	3	3	14	**X**	7	7	14
~X	3	7	6	**~X**	7	3	20
Sum	6	10	20	**Sum**	14	10	24

4.3.2 Naïve Bayes

The previous section defined the notion of the independence between two variables from the perspective of probability theory (e.g., Equation 4-3). Let us now consider a version of Bayes's theorem that assumes that two variables are independent in this sense.

Bayes's theorem can be extended to deal with problems that cannot be represented in a 2×2 contingency table. That is, Bayes's rule can assess posterior probability in the context of more than one source of evidence. For example, consider the situation in which two sources of evidence (X and Y) can both be either true or false. Bayes's rule for computing the posterior probability of a hypothetical cause H when both X and Y are true is given as Equation 4-4, which is an extension of Equation 3-2 because it employs two sources of evidence instead of just one.

$$P(H|X\cap Y) = \frac{P(X\cap Y|H)\cdot P(H)}{\left(P(X\cap Y|H)\cdot P(H)\right)+\left(P(X\cap Y|\sim H)\cdot P(\sim H)\right)} \tag{4-4}$$

Because Equation 4-4 is concerned with two sources of evidence, it is not applied to a situation that is simple enough to be represented with a 2×2 contingency table of the sort that was investigated in Chapter 2. Instead, Equation 4-4 applies to data that can be represented using a 2×2×2 contingency table. The general form of this type of contingency table is illustrated in Table 4-7. In essence, Table 4-7 is a nested set of two 2×2 contingency tables. The first provides the frequencies of co-occurrences of the states of X and Y that map onto H; the second provides the frequencies of co-occurrences of the states of X and Y that map onto $\sim H$. One can derive all of the probabilities required for Equation 4-1 from the frequencies in this table. For instance, $P(X \cap Y|H)$ equals $a/(a+b+c+d)$, whereas $P(X \cap Y|\sim H)$ equals $e/(e+f+g+h)$. Similarly, $P(H) = (a+b+c+d)/(a+b+c+d+e+f+g+h)$, while $P(\sim H) = (e+f+g+h)/(a+b+c+d+e+f+g+h)$.

Table 4-7. General form of a 2×2×2 contingency table for two sources of evidence (X, Y) and one hypothetical cause (H).

	H	
	Y	**~Y**
X	*a*	*b*
~X	*c*	*d*

	~H	
	Y	**~Y**
X	*e*	*f*
~X	*g*	*h*

Note. Each lowercase letter in a cell stands for a frequency. For instance, a is the number of times that H is true when X and Y are both true.

Important to note, more than one version of Bayes's rule is available for situations in which multiple sources of evidence are considered. In the pattern recognition literature, extensions of Equation 4-4 become too computationally expensive as the number of features to consider climbs. As a result, Bayesian pattern recognition systems often use an equation called *naïve Bayes* (Duda & Hart, 1973; Friedman, Geiger, & Goldszmidt, 1997). Naïve Bayes makes the overly strong assumption that different types of available evidence are conditionally independent. Informally, this is the assumption that there are no statistical interactions between different sources of evidence. This assumption simplifies computing posterior probabilities. In addition, when violations of the conditional independence assumption occur, the resulting (incorrect) probabilities still permit satisfactory pattern classification. Equation 4-5 provides the naïve Bayes equation for the case in which only two sources of evidence are considered.

$$P(H|X \cap Y) = \frac{P(X|H) \cdot P(Y|H) \cdot P(H)}{\big(P(X|H) \cdot P(Y|H) \cdot P(H)\big) + \big(P(X|\sim H) \cdot P(Y|\sim H) \cdot P(\sim H)\big)} \quad (4\text{-}5)$$

Note that a comparison between Equation 4-5 and Equation 4-4 indicates how Bayes's theorem changes when conditional independence is assumed. In particular, Equation 4-5 simplifies Equation 4-4 by using assumed independence—to be precise, Equation 4-3—to rewrite a conditional probability that ANDs two variables (e.g., $P(X \cap Y|H)$ into the produce of two simpler conditional probabilities ($P(X|H) \bullet P(Y|H)$). This translation of terms will be true only if X and Y are truly independent of each other.

4.3.3 Perceptrons and Naïve Bayes

Let us now show the formal equivalence between the structure of a perceptron and the naïve Bayes's rule (Equation 4-5) for any of the Table 4-2 problems. In general, our proofs follow the same logic as the proof developed for the single variable problem earlier in Chapter 3. However, because the Table 4-2 problems involve two binary inputs, the proofs are slightly more involved, for we must consider four input situations.

To begin, let us consider the general properties of a modern perceptron trained on any of the Table 4-2 operators. This perceptron has two inputs that send signals to a single output unit. Let the weight of the connection between the input unit for X and the output unit be represented as wx, and let the weight of the connection between the input unit for Y and the output unit be wy. When an input variable is true, the input representing that variable sends a signal of 1; if the input variable is false, then its input sends a signal of 0. Thus, when provided an input pattern that indicates that both X and Y are true, the net input for the output unit is $wx + wy$. This net input is combined with the output unit's bias θ and passed into the logistic function given in Equation 4-6 to generate the perceptron's estimate of $P(H|X \cap Y)$.

$$\begin{aligned} &P(H|X \cap Y) \\ &= f(X \cap Y) \\ &= \frac{1}{1 + e^{-(w_x + w_y + \theta)}} \end{aligned} \quad (4\text{-}6)$$

We can now take a different equation for $P(H|X \cap Y)$, the naïve Bayes's rule given before as Equation 4-2. By dividing both the numerator and the denominator of this equation by the numerator of Equation 4-5, we convert the naïve Bayes's rule into a format that is structurally similar to the logistic function. Equation 4-7 provides this version of the naïve Bayes's equation, and replaces its probabilities with the appropriate ratios of cell frequencies taken from Table 4-7.

$$\begin{aligned} &P(H|X \cap Y) \\ &= \frac{1}{1 + \dfrac{P(X|\sim H) \cdot P(Y|\sim H) \cdot P(\sim H)}{P(X|H) \cdot P(Y|H) \cdot P(H)}} \\ &= \frac{1}{1 + \dfrac{(e+f) \cdot (e+g) \cdot (a+b+c+d)}{(a+b) \cdot (a+c) \cdot (e+f+g+h)}} \end{aligned} \quad (4\text{-}7)$$

Because we know that perceptrons generate conditional probabilities—consider the results on the right hand side of Table 4-2—we now equate Equations 4-6 and 4-7, as shown in Equation 4-8:

$$\begin{aligned} P(H|X \cap Y) &= \frac{1}{1+e^{-(w_x+w_y+\theta)}} \\ &= \frac{1}{1+\frac{(e+f)\cdot(e+g)\cdot(a+b+c+d)}{(a+b)\cdot(a+c)\cdot(e+f+g+h)}} \end{aligned} \tag{4-8}$$

From the structural similarities between the two expressions, an obvious equality exists as shown in Equation 4-9:

$$\begin{aligned} &e^{-(w_x+w_y+\theta)} \\ &= \frac{(e+f)\cdot(e+g)\cdot(a+b+c+d)}{(a+b)\cdot(a+c)\cdot(e+f+g+h)} \end{aligned} \tag{4-9}$$

By taking the natural logarithm of both sides of Equation 4-9 and then multiplying both sides by −1, we finally express the sum of three network structures (the two weights and the bias) in terms of contingency table frequencies that were derived from the naïve Bayes's rule. This is presented as Equation 4-10:

$$\begin{aligned} &w_x + w_y + \theta \\ &= -ln\left(\frac{(e+f)}{(a+b)}\right) - ln\left(\frac{(e+g)}{(a+c)}\right) - ln\left(\frac{(a+b+c+d)}{(e+f+g+h)}\right) \end{aligned} \tag{4-10}$$

We now repeat the procedure just detailed for the remaining three combinations of input variables. Equation 4-11 provides the relation between the logistic and the naïve Bayes versions of P(H|X∩~Y). Equation 4-12 provides the expression of network structures in terms of contingency table frequencies after identifying the equivalence between the logistic and probabilistic versions of Equation 4-11, taking the natural logarithms, and multiplying through by −1.

$$\begin{aligned} &P(H|X \cap {\sim}Y) \\ &= \frac{1}{1+\frac{P(X|{\sim}H)\cdot P({\sim}Y|{\sim}H)\cdot P({\sim}H)}{P(X|H)\cdot P({\sim}Y|H)\cdot P(H)}} \\ &= \frac{1}{1+\frac{(e+f)\cdot(f+h)\cdot(a+b+c+d)}{(a+b)\cdot(b+d)\cdot(e+f+g+h)}} \\ &= \frac{1}{1+e^{-(w_x+\theta)}} \end{aligned} \tag{4-11}$$

$$\begin{aligned} &w_x + \theta \\ &= -ln\left(\frac{(e+f)}{(a+b)}\right) - ln\left(\frac{(f+h)}{(b+d)}\right) - ln\left(\frac{(a+b+c+d)}{(e+f+g+h)}\right) \end{aligned} \tag{4-12}$$

Similarly, Equation 4-13 provides the relation between the logistic and the naïve Bayes versions of P(H|~X∩Y), and Equation 4-14 expresses network structures in terms of contingency table frequencies after carrying out the sequence of steps just described.

$$\begin{aligned} &P(H|{\sim}X \cap Y) \\ &= \frac{1}{1+\frac{P({\sim}X|{\sim}H)\cdot P(Y|{\sim}H)\cdot P({\sim}H)}{P({\sim}X|H)\cdot P(Y|H)\cdot P(H)}} \\ &= \frac{1}{1+\frac{(g+h)\cdot(e+g)\cdot(a+b+c+d)}{(c+d)\cdot(a+c)\cdot(e+f+g+h)}} \\ &= \frac{1}{1+e^{-(w_y+\theta)}} \end{aligned} \tag{4-13}$$

$$\begin{aligned} &w_y + \theta \\ &= -ln\left(\frac{(g+h)}{(c+d)}\right) - ln\left(\frac{(e+g)}{(a+c)}\right) - ln\left(\frac{(a+b+c+d)}{(e+f+g+h)}\right) \end{aligned} \tag{4-14}$$

Finally, Equation 4-15 provides the relation between the logistic and the naïve Bayes versions of P(H|~X∩~Y), and Equation 4-16 expresses the bias of the output unit in terms of contingency table frequencies after carrying out the sequence of steps just described.

$$\begin{aligned} &P(H|{\sim}X \cap {\sim}Y) \\ &= \frac{1}{1+\frac{P({\sim}X|{\sim}H)\cdot P({\sim}Y|{\sim}H)\cdot P({\sim}H)}{P({\sim}X|H)\cdot P({\sim}Y|H)\cdot P(H)}} \\ &= \frac{1}{1+\frac{(g+h)\cdot(f+h)\cdot(a+b+c+d)}{(c+d)\cdot(b+d)\cdot(e+f+g+h)}} \\ &= \frac{1}{1+e^{-(\theta)}} \end{aligned} \tag{4-15}$$

$$\begin{aligned} &\theta \\ &= -ln\left(\frac{(g+h)}{(c+d)}\right) - ln\left(\frac{(f+h)}{(b+d)}\right) - ln\left(\frac{(a+b+c+d)}{(e+f+g+h)}\right) \end{aligned} \tag{4-16}$$

Equation 4-16 provides an expression for θ alone. By subtracting it from Equation 4-12, we can now express *wx* on its own in terms of contingency table frequencies, as given in Equation 4-17. Equation 4-17 also simplifies this expression into a single natural logarithm:

$$
\begin{aligned}
& w_x \\
&= -ln\left(\frac{(e+f)}{(a+b)}\right) - ln\left(\frac{(f+h)}{(b+d)}\right) - ln\left(\frac{(a+b+c+d)}{(e+f+g+h)}\right) - \theta \\
&= -ln\left(\frac{(e+f)}{(a+b)}\right) - ln\left(\frac{(f+h)}{(b+d)}\right) - ln\left(\frac{(a+b+c+d)}{(e+f+g+h)}\right) \\
&\quad - \left(-ln\left(\frac{(g+h)}{(c+d)}\right) - ln\left(\frac{(f+h)}{(b+d)}\right) - ln\left(\frac{(a+b+c+d)}{(e+f+g+h)}\right)\right) \\
&= ln\left(\frac{(g+h)}{(c+d)}\right) - ln\left(\frac{(e+f)}{(a+b)}\right) \\
&= ln\left(\frac{\frac{(g+h)}{(c+d)}}{\frac{(e+f)}{(a+b)}}\right) \\
&= ln\left(\frac{(a+b)\cdot(g+h)}{(e+f)\cdot(c+d)}\right)
\end{aligned}
\tag{4-17}
$$

We can also subtract the expression for θ (Equation 4-16) from Equation 4-14 to express *wy* on its own in terms of contingency table frequencies. This is done in Equation 4-18, which also simplifies this expression into a single natural logarithm:

$$
\begin{aligned}
& w_y \\
&= -ln\left(\frac{(g+h)}{(c+d)}\right) - ln\left(\frac{(e+g)}{(a+c)}\right) - ln\left(\frac{(a+b+c+d)}{(e+f+g+h)}\right) \\
&\quad - \theta \\
&= -ln\left(\frac{(g+h)}{(c+d)}\right) - ln\left(\frac{(e+g)}{(a+c)}\right) - ln\left(\frac{(a+b+c+d)}{(e+f+g+h)}\right) \\
&\quad - \left(-ln\left(\frac{(g+h)}{(c+d)}\right) - ln\left(\frac{(f+h)}{(b+d)}\right) - ln\left(\frac{(a+b+c+d)}{(e+f+g+h)}\right)\right) \\
&= ln\left(\frac{(f+h)}{(b+d)}\right) - ln\left(\frac{(e+g)}{(a+c)}\right) \\
&= ln\left(\frac{\frac{(f+h)}{(b+d)}}{\frac{(e+g)}{(a+c)}}\right) \\
&= ln\left(\frac{(a+c)\cdot(f+h)}{(b+d)\cdot(e+g)}\right)
\end{aligned}
\tag{4-18}
$$

In our earlier analysis of the single input perceptron in Chapter 3, we discovered that its connection weights were a function of the odds ratio for a 2×2 contingency table (i.e., the ratio of the products of the two diagonals; Rudas, 1998). The final expressions in Equations 4-14 and 4-15 also have the appearance of being odds ratios, for they are natural logarithms of the ratios of products. Further analysis indicates that each weight represents the association between an input unit and the output unit in terms of an odds ratio. However, in this case, it is an odds ratio that ignores the potential effects of any other input variables.

Consider alternative presentations of Table 4-7, the generic 2×2×2 contingency table. One could create a 2×2 table that related *X* to *H* by collapsing (i.e., summing) Table 4-7 over *Y*. We present the result as Table 4-8A. The odds ratio for Table 4-8A is the ratio of the products of its diagonals, ((*a*+*b*)·(*g*+*h*))/((*c*+*d*)·(*e*+*f*)). Note that the final expression of Equation 4-17 is the natural logarithm of this odds ratio. Similarly, one could create a 2×2 table that related *Y* to *H* by collapsing (i.e., summing) Table 4-7 over *X*. We present the result as Table 4-8B. The odds ratio for Table 4-8B is ((*a*+*c*)·(*f*+*h*))/((*b*+*d*)·(*e*+*g*)). Note that the final expression of Equation 4-18 is the natural logarithm of this odds ratio. In short, each connection weight is a function of the odds ratio that expresses the association between a single source of evidence and the output, collapsing over the contribution of other available evidence.

Table 4-8. General form of collapsing 2×2×2 contingency table (Table 2) across input variables. (A) The 2×2 contingency table relating X to H, collapsing over y. (B) The 2×2 contingency table relating Y to H, collapsing over X.

	H	**~H**
X	*a+b*	*e+f*
~X	*c+d*	*g+h*

	H	**~H**
X	*a+c*	*e+g*
~X	*b+d*	*f+h*

This analysis provides the formal definition of a Bayesian perceptron's connection weights for problems involving more than two sources of evidence (i.e., involving more than two input units). Each connection weight is the natural logarithm of the odds ratio of the 2×2 contingency table that relates one source of evidence to the output, collapsing over all other possible sources of evidence. Each of these effects can be described mathematically as the natural logarithm of a particular odds ratio. This finding is important, because odds ratios also define the coefficients obtained when one uses logistic regression to predict the state of a hypothesis using multi-

Table 4-10. The responses of 16 perceptrons trained on probabilistic versions of the logical operations of modern Boolean algebra.

Predicate	X	Y	Naïve Bayes	Perceptron	Predicate	X	Y	Naïve Bayes	Perceptron
Contradiction	0	0	0.30	0.30	**AND(~X, ~Y)**	0	0	0.60	0.62
	0	1	0.30	0.30		0	1	0.39	0.38
	1	0	0.30	0.30		1	0	0.39	0.38
	1	1	0.30	0.30		1	1	0.22	0.18
~X	0	0	0.70	0.70	**AND(~X, Y)**	0	0	0.39	0.38
	0	1	0.70	0.70		0	1	0.60	0.63
	1	0	0.30	0.30		1	0	0.22	0.18
	1	1	0.30	0.30		1	1	0.39	0.38
~Y	0	0	0.70	0.70	**AND(X, ~Y)**	0	0	0.39	0.38
	0	1	0.30	0.30		0	1	0.22	0.18
	1	0	0.70	0.70		1	0	0.60	0.62
	1	1	0.30	0.30		1	1	0.39	0.38
Y	0	0	0.30	0.30	**XOR(X, Y)**	0	0	0.50	0.50
	0	1	0.70	0.70		0	1	0.50	0.50
	1	0	0.30	0.30		1	0	0.50	0.50
	1	1	0.70	0.70		1	1	0.50	0.50
X	0	0	0.30	0.30	**~AND(X, Y)**	0	0	0.78	0.82
	0	1	0.30	0.30		0	1	0.61	0.63
	1	0	0.70	0.70		1	0	0.61	0.62
	1	1	0.70	0.70		1	1	0.40	0.38
Tautology	0	0	0.70	0.70	**AND(X, Y)**	0	0	0.22	0.18
	0	1	0.70	0.70		0	1	0.39	0.38
	1	0	0.70	0.70		1	0	0.39	0.38
	1	1	0.70	0.70		1	1	0.60	0.63
					~XOR(X, Y)	0	0	0.50	0.50
						0	1	0.50	0.50
						1	0	0.50	0.50
						1	1	0.50	0.50
					X ⊃ Y	0	0	0.61	0.62
						0	1	0.78	0.82
						1	0	0.40	0.38
						1	1	0.61	0.63
					Y ⊃ X	0	0	0.61	0.62
						0	1	0.40	0.38
						1	0	0.78	0.82
						1	1	0.61	0.63
					OR(X, Y)	0	0	0.40	0.38
						0	1	0.61	0.63
						1	0	0.61	0.62
						1	1	0.78	0.82

Note. For each stimulus, the probability of reward computed using the naïve Bayes equation is provided alongside the response of the perceptron after 1,000 epochs of training on the problem.

ple sources of information. Previous work has shown that there exists formal equivalence between a perceptron that uses the logistic activation function and logistic regression (Schumacher et al., 1996). We see next that one can relate the structure (weights and bias) of a perceptron trained on an uncertain training set to the coefficients of a logistic regression performed on the same training set.

When we analyzed the structure of the one input unit perceptron in Chapter 3, we also found that its bias value θ was a function of the odds relating *H* to *~H*, but only in cases for which there was no evidence (i.e., for *~X*). An analogous situation holds for the more complex perceptron. Table 4-9A presents the 1×2 contingency table that relates *~X* to *H*, collapsing over *y*. It is simply the bottom row of Table 4-8A. For this table, the odds of *H* relative to *~H* is the ratio of the two cells, (*c*+*d*)/(*g*+*h*). Similarly, Table 4-9B presents the 1×2 contingency table that relates *~Y* to *H*, collapsing over *X*. It is simply the bottom row of Table 4-8B. For this table, the odds of *H* relative to *~H* is (*b*+*d*)/(*f*+*h*).

Table 4-9. (A) The 1×2 contingency table relating ~X to H, collapsing over y. (B) The 1×2 contingency table relating ~Y to H, collapsing over X.

	H	**~H**
~X	*c+d*	*g+h*

(A)

	H	**~H**
~X	*b+d*	*f+h*

(A)

A simple reworking of Equation 4-16 relates components of the expression for θ to these two sets of odds. Equation 4-19 takes Equation 4-16 and reexpresses it after making its first two components positive instead of negative. The first component of Equation 4-19 is the natural logarithm of the odds computed from Table 4-9A. The second component of Equation 4-19 is the natural logarithm of the odds computed from Table 4-9B. The final component of Equation 4-19 is also a natural logarithm of odds—in this case, the odds of *H* relative to *~H* collapsing over all different states of evidence! Again, this definition of θ could be extended for perceptrons that generate probabilities using more than two sources of evidence.

$$\begin{aligned}\theta &= -ln\left(\frac{(g+h)}{(c+d)}\right) - ln\left(\frac{(f+h)}{(b+d)}\right) - ln\left(\frac{(a+b+c+d)}{(e+f+g+h)}\right) \\ &= ln\left(\frac{(c+d)}{(g+h)}\right) + ln\left(\frac{(b+d)}{(f+h)}\right) - ln\left(\frac{(a+b+c+d)}{(e+f+g+h)}\right)\end{aligned} \tag{4-19}$$

4.3.4 Perceptron Responses and Naïve Bayes

The equations just presented have developed the relationship between the structure of perceptrons and the naïve Bayes equation (Equation 4-5). This development has been based on the assumption that perceptrons implement this form of Bayes's theorem, as is made explicit in Equation 4-5. Do we have evidence for making this assumption?

Such evidence is available to us by considering again the performance of the perceptrons trained on the contingent versions of the Boolean operators. Table 4-10 provides this same performance but compares it to a different set of predictions. Although Table 4-2 compared perceptron performance to probabilities predicted from the full version of Bayes's theorem (Equation 4-4), Table 4-10 instead compares perceptron performance to the probabilities predicted from the naïve Bayes equation (Equation 4-5). An examination of Table 4-10 reveals that in each case, perceptron responses match the probability predicted by naïve Bayes. The correlation between the naïve Bayesian column and the perceptron column of Table 4-10 is 0.9975. This provides clear evidence that perceptrons are naïve Bayesian mechanisms. That is, they can predict probability of reward from multiple cues but do so assuming that there are no interactions between cues. This issue is discussed in more detail in the next section.

We can use Equation 4-3 to provide additional insight into the behavior of the perceptrons that were trained on the probabilistic versions of the Boolean operators. We can determine the values of $P(X|H)$, $P(Y|H)$, and $P(X\cap Y|H)$ for each of these training sets. For instance, $P(X|H)$ is equal to the number of times that *X* is presented and rewarded divided by the total number of times that any pattern in the training set is rewarded. Table 4-11 presents the values of these three probabilities for each of the 16 training sets. Its fourth column also provides the value of the product of the second and third columns (i.e., $P(X|H)\bullet P(Y|H)$). The penultimate column in Table 4-11 indicates whether Equation 4-3 is true for each training set. That is, if $P(X|H) \bullet P(Y|H)$ is equal to the value $P(X\cap Y|H)$ that was computed for the training set, then the value in this column is 1, and it indicates that *X* and *Y* are conditionally independent signals. If $P(X|H) \bullet P(Y|H)$ is not equal to the value $P(X\cap Y|H)$ that was computed for the training set, then the value in this column is 0, and it indicates that *X* and *Y* are not conditionally independent signals. The final column in Table 4-11 provides the value of *G* (Section 4.3.1) calculated for each probabilistic operator. Note that *G* is only equal to 0 for operators in which *X* and *Y* are conditionally independent.

Table 4-11 reveals that the degree of conditional dependence (measured by G) predicts the degree of incorrectness in perceptron performance. For instance, consider the AND operator with $G = 1.61$. The sum of squared differences between the four desired responses and the four actual perceptron responses for this operator in Table 4-2 was about 0.03, indicating that the perceptron's performance, though not perfect, was still quite good. The smaller value for G for this operator, as well as the small squared difference between the values of $P(X|H) \bullet P(Y|H)$ and $P(X \cap Y|H)$ for this operator (0.002), indicates that although it is not conditionally independent the violation of this assumption is small. This means that the perceptron's performance is reasonable.

Table 4-11. The analysis of the conditional independence between X and Y for each of the probabilistic versions of training sets used to teach perceptrons about Boolean algebras

Predicate	P(X|H)	P(Y|H)	P(X|H)	P(X	Independent	G
Contradiction	0.500	0.500	0.250	0.250	1	0.00
AND(~X, ~Y)	0.375	0.375	0.141	0.188	0	1.61
AND(~X, Y)	0.625	0.375	0.234	0.188	0	1.61
~X	0.500	0.300	0.150	0.150	1	0.00
AND(X, ~Y)	0.375	0.625	0.234	0.188	0	1.61
~Y	0.300	0.500	0.150	0.150	1	0.00
XOR(X, Y)	0.500	0.500	0.250	0.150	0	6.58
~AND(X, Y)	0.417	0.417	0.174	0.125	0	1.61
AND(X, Y)	0.625	0.625	0.391	0.438	0	1.61
~XOR(X, Y)	0.583	0.417	0.243	0.292	0	6.58
Y	0.700	0.500	0.350	0.350	1	0.00
X	0.583	0.417	0.243	0.292	0	1.61
X	0.500	0.700	0.350	0.350	1	0.00
Y	0.417	0.583	0.243	0.292	0	1.61
OR(X, Y)	0.583	0.583	0.340	0.292	0	1.61
Tautology	0.500	0.500	0.250	0.250	1	0.00

In contrast, now consider the XOR operator. The sum of squared differences between the four desired responses and the four actual perceptron responses for this operator in Table 4-2 was 0.16. This is an order of magnitude larger than the perceptron's error for AND, showing that the perceptron's performance for XOR is much poorer. The squared difference between the values of $P(X|H) \bullet P(Y|H)$ and $P(X \cap Y|H)$ for this operator in Table 4-11 is 0.01, which is also an order of magnitude larger than was the case for AND. The value of G for this operator (6.58) is about 4 times larger than the value of G for AND. This indicates XOR involves a larger violation of the conditional independence assumption—that there is a stronger interaction between X and Y for XOR—and this in turn produces poorer probability matching in the perceptron.

Indeed the amount of conditional dependence between X and Y in a probabilistic Boolean training set almost perfectly predicts the perceptron performance that was provided earlier in Table 4-2. For each of the 16 operators the sum of squared differences between the probability of reward and the perceptron response for each set of cues was calculated as a measure of perceptron accuracy. This measure was then correlated with the correlated with the G values provided in Table 4-11. This correlation equals 0.998, indicating that G is almost a perfect predictor of perceptron accuracy. As G increases, the accuracy of perceptron estimates of probability decreases.

One minor issue remains. The proofs in Chapter 3 demonstrated that perceptrons instantiated Bayes's theorem, whereas the proofs in the current chapter indicate that perceptrons implement the naïve Bayes rule. Are these two findings consistent? The answer to this question is straightforward: When only one source of evidence is available, standard Bayes and naïve Bayes are identical, because a single source of evidence cannot interact with itself. We therefore can arrive at the general conclusion that modern perceptrons are naïve Bayesians.

In the next chapter, we pursue the implications of this general conclusion in a series of computer simulations. It begins by briefly considering the notion of interaction in the context of conditional dependence. Chapter 5 then proceeds to report the results of a number of computer simulations that are motivated by the formal results that we have developed in Chapter 4.

4.4 Signals From Three or More Cues

The formal analysis developed in Section 4.3 assumes that the probability of reward is signaled by two cues. Important to note, essentially the same analysis could be developed for situations involving three or more cues. These extended proofs would begin by characterizing the reward contingencies with more complex contingency tables. For instance, if reward is being signaled by three cues, then the 2×2×2 contingency table provided as Table 4-7 would be replaced with a 2×2×2×2 contingency table that would take into account a third cue that could be present or not. Similarly, a proof concerning a four-cue environment would begin with a 2×2×2×2×2 contingency table. Once a more complex contingency table is defined, then the variables in its cells can be used to define the probabilities for a naïve Bayesian equation that uses more than two sources of evidence. With these probabilities in hand, they can be translated into the form of the logistic equation using a procedure similar to that used in Section 4.3 or earlier in Section 3.4. The only difference is that more equations have to be developed to consider the predictions based on additional signals. The nature of the equations would be analogous to those described earlier in this chapter. That is, each connection weight would be defined by the natural logarithm of the odds ratio associated with a particular cue.

As this monograph proceeds, we are not concerned with providing equations that provide the odds ratios when more than three cues are involved. Given the formal analyses provided in Section 4.3, these additional equations are not required, because they will not provide any new information beyond that which has already been demonstrated in the current chapter.

However, when later chapters explore the behavior of perceptrons that learn from more than two cues, we use a simple procedure to confirm that perceptron structure (weights and bias) conform to odds ratios as Section 4.3 indicates. We take advantage of the fact that when logistic regression uses multiple cues to predict an outcome, the coefficients in the regression equation are equal to the natural logarithm of the odds ratio associated with each predictor (Christensen, 1997; Cramer, 2003; Hosmer & Lemeshow, 2000). This means that we can use logistic regression to predict the outcome (reward) using the same training sets used to train perceptrons. If the perceptron structure takes the expected form, then its weights and bias should be identical to the corresponding coefficients in a logistic regression equation (Dawson & Gupta, 2017). To preview a main result of Chapter 5, this relation between logistic regression coefficients and perceptron structure is shown to be true for five different simulations in which reward is signaled by three different cues.

4.5 Summary and Implications

4.5.1 Summary

Chapter 3 provided empirical evidence and formal proofs that when perceptrons learn the probability of reward signaled by a single cue, the probability that is generated is that defined by Bayes's theorem. Chapter 4 has extended the Chapter 3 material to consider situations in which the probability of reward is signaled by two or more cues. The main result of Chapter 4 is that a perceptron that learns probabilities signaled by multiple cues is still a Bayesian mechanism, but one of a particular type that has specific limitations. In multiple cue situations, perceptrons compute the probability of reward in accordance with the naïve Bayes rule, which means that these probabilities are founded on the assumption that the different signals do not interact with one another. That is, these perceptrons assume that the predictions supplied by the various cues are independent of one another and are therefore blind to interactions between cues.

4.5.2 Implications

The formal analyses provided in Section 4.3 provide an important extension to those that were presented in Chapter 3. They confirm the general notion that perceptrons are Bayesian mechanisms. However, they also highlight important limitations of the ability of perceptrons to estimate probabilities because of their simple structure. In particular, if the probability of reward is signaled by the interaction between two or more cues, then a perceptron will be blind to this signal. Its probability estimates are limited, by the perceptron's simple structure, to the naïve Bayesian assumption that the signals of reward from different cues are mutually independent.

If we are to consider perceptrons as potential models of human or animal probability matching, then the limitation revealed in the current chapter is crucial. This is because if perceptrons can model this behavior, then it must be the case that human or animal probability matching is also impeded when signals from different predictive cues interact with one another.

Chapter 5 returns to the algorithmic level to set the stage for exploring the relationship between perceptron probability estimation and corresponding behavior in other agents. It presents a series of simulations in which three different cues are used to signal the probability of reward. One reason for these simulations is that they are closely related to the structure of the card-choice task that was introduced in Chapter 1. Thus, the simulations to be presented in Chapter 5 will position us to understand possible factors that might affect human performance on the card-choice task—assuming that perceptrons are relevant models of this performance. A second reason for these simulations is that they are structured in such a way to explore different ways to characterize the limitations that perceptrons face when estimating probability. In particular, the simulations contrast two notions of these limitations. One is a logical limitation (linear nonseparability) that is traditionally used to define what tasks can be learned by a perceptron, and what tasks cannot (Minsky & Papert, 1969). The other is a quantitative limitation (conditional dependence) that emerges from probability theory. We see in the next chapter that it is this second approach to perceptron limitations that is more useful for predicting situations in which perceptrons have difficulty estimating probability. Later we consider if it is also useful for making similar predictions for human performance in a probability discrimination task.

Chapter 5: Estimating Reward Probability From Three Cues

The purpose of this chapter is to explore empirically some formal predictions from Chapter 4. Those formal analyses indicated that perceptrons are naïve Bayesian mechanisms that are blind to interactions between cues that signal rewards. In terms of probability theory, these interactions introduce conditional dependence. Chapter 5 reports the results of five simulation studies in which perceptrons are trained about environments that provide three different cues, where these cues are uncertain signals of reward. The first simulation studies the case in which all three cues are conditionally independent. The next two simulations introduce conditional dependence by using logical relations between two cues to introduce an interaction. One uses the linearly separable AND relation between the two cues, whereas the other uses the linearly nonseparable XOR relation between them. When either logical relation is true a high probability of reward is signaled, which introduces a higher degree of conditional dependence. These two simulations are then repeated but with a lower probability of reward signaled by the interaction. This reduces the conditional dependence of the two cues while maintaining their logical relationship. In general, the results of these five simulations confirm the formal prediction that perceptrons are naïve Bayesian mechanisms when faced with three uncertain cues. Perceptrons are highly accurate probability matchers when conditional dependence is low and are less accurate when conditional dependence is high. The amount of conditional dependence in a training set is a far better predictor of perceptron probability estimation than is the logical nature of the interaction between cues. This leads to the conclusion that when perceptrons are learning about uncertain environments, conditional dependence provides a much better account of their limitations than does the more traditional notion of linear nonseparability.

5.1 Probability Estimation From Three Independent Cues

5.1.1 Rationale for Simulations

The main conclusion drawn from the formal analyses provided in Chapter 4 is that perceptrons are blind to interactions between cues. We know that they are blind to these interactions, because we have shown that when processing multiple signals, their probability estimates are equivalent to those generated by the naïve Bayes equation. This means that perceptrons compute the probability of reward that is signaled by multiple cues under the assumption that each cue is conditionally independent of the others.

Simulation studies are consistent with these formal analyses. Chapter 4 demonstrated that perceptrons could perfectly match probabilities using two sources of evidence. However, this required the two sources of evidence to be conditionally independent. If there is conditional dependence between the two sources of evidence—if they interact—then perceptrons are not able to estimate probabilities perfectly. The degree of error in their probability estimates indicates the degree to which the conditional independence assumption is violated. The greater the interaction between the two sources of evidence, the poorer the perceptron's probability estimation performance.

With these formal and empirical results in mind, we now proceed to a series of simulation studies in which perceptrons learn to probability match when three sources of evidence are concerned. These simulations provide an analog to the card-choice task introduced in Chapter 1, because in that task each card was identified by the binary states of three cues. In the simulations described in Chapter 5, a variety of conditions are studied. In particular, we vary degrees of conditional dependence as well as the logical

Table 5-1. The probability of reward for each of the eight possible cue states of a training set in which the effects of three cues are conditionally independent. See text for details.

Input	Probability	Probability	Expected	Set 1	Set 2	Set 3	Set 4	Set 5
(0,0,0)	P(R|~A~B~C)	0	0.00	0.00	0.00	0.00	0.00	0.00
(0,0,1)	P(R|~A~BC)	P(R|C)	0.40	0.4	0.35	0.375	0.4	0.45
(0,1,0)	P(R|~AB~C)	P(R|B)	0.20	0.15	0.175	0.225	0.275	0.15
(0,1,1)	P(R|~ABC)	P(R|B) + P(R|C) - (P(R|B) * P(R|C))	0.52	0.5	0.475	0.55	0.575	0.55
(1,0,0)	P(R|A~B~C)	P(R|A)	0.14	0.15	0.15	0.1	0.075	0.075
(1,0,1)	P(R|A~BC)	P(R|A) + P(R|C) - (P(R|A) * P(R|C))	0.484	0.475	0.45	0.45	0.55	0.475
(1,1,0)	P(R|AB~C)	P(R|A) + P(R|C) - (P(R|A) * P(R|C))	0.312	0.25	0.35	0.375	0.275	0.225
(1,1,1)	P(R|ABC)	P(R|A) + P(R|B) + P(R|C) - (P(R|A)* P(R|B)) - (P(R|A)* P(R|C)) - (P(R|B)* P(R|C)) + (P(R|A) * P(R|B) + P(R|C))	0.587	0.675	0.475	0.55	0.625	0.575

structure of the rules used to decide whether the perceptron should be rewarded. Our analysis of perceptron performance reveals that it is completely consistent with the conclusions that we have drawn from our detailed analysis of the two-predictor case. Later we compare these results to those of human participants in the card-choice task.

5.1.2 Network Architecture and Training Set

Let us begin by describing the results of training perceptrons on a task in which the three cues used to signal reward are conditionally independent. The reward signaled by the cues in this condition is uncertain in the same sense that we have been considering up to this point in the monograph: sometimes a signal (a pattern of cues) will lead to a reward, and other times the same signal will not lead to a reward. During training, a perceptron is presented a stimulus (some combination of three cues, each of which is either present or not), is rewarded or not (is trained to turn on or not), and then updates its bias and connection weights. This procedure is then repeated for the next stimulus until training is completed. With this procedure, one can consider the perceptron to be an example of an extremely simple reinforcement learning system (Sutton & Barto, 1998).

In this first simulation, each cue signals that there is a certain probability of being rewarded. The probability of being rewarded that is signaled by one cue is conditionally independent of the probability that is signaled by the other two cues. There are several purposes for conducting this simulation study. First, it provides empirical evidence that is consistent with the formal analyses from Chapter 4. Second, it replicates the findings of a similar study that examined probability estimation in perceptrons presented four independent cues (Dawson & Gupta, 2017). Third, this first simulation study serves as the control condition for several other studies that are reported later in this chapter. These studies explore the effect that introducing conditional dependence between cues has on probability estimation. Our understanding of those effects requires us to compare those results with those we are about to report. Fourth, the conditions being explored in this first simulation—in particular, the values of the probability of reward associated with each cue—are compared to those used in human experiments that are reported in Chapter 7. Thus, these simulations will collect data that permits us to compare probability estimation of networks to probability matching by humans under analogous conditions.

We trained modern perceptrons comprising a single output unit and three input units. The input units represented the presence or absence of three cues: Cue A, Cue B, and Cue C. For example, the input pattern (1, 0, 0) indicates the presence of Cue A and the absence of the other two cues. We used each of the eight possible cue configurations (given this binary coding) in a training set.

Each cue signals a different probability of reward. In this simulation, the presence of Cue A indicated a reward probability of 0.14, the presence of Cue B indicated a reward probability of 0.20, and the presence of Cue C indicated a reward probability of 0.40. These reward probabilities were selected to ensure that the overall likelihood of reward—the probability of reward across all of the training patterns—was approximately 0.33. This overall likelihood of reward was maintained for the other simulations that

Table 5-2. The probability of reward (P(R)) and the mean responses of perceptrons to stimuli after being trained on the different conditionally independent three-cue training sets.

	Training Set 1		**Training Set 2**		**Training Set 3**		**Training Set 4**		**Training Set 5**	
Input	P(R)	Mean	P(R)	Mean	P(R)	Mean	P(R)	Mean Response	P(R)	Mean
0,0,0	0	0	0	0.115	0	0.109	0	0.098	0	0.085
0,0,1	0.4	0.45	0.35	0.314	0.375	0.335	0.4	0.399	0.45	0.419
0,1,0	0.15	0.15	0.175	0.186	0.225	0.225	0.275	0.194	0.15	0.140
0,1,1	0.5	0.55	0.475	0.447	0.55	0.545	0.575	0.596	0.55	0.559
1,0,0	0.15	0.075	0.15	0.168	0.1	0.148	0.075	0.136	0.075	0.102
1,0,1	0.475	0.475	0.45	0.418	0.45	0.418	0.55	0.490	0.475	0.469
1,1,0	0.25	0.225	0.35	0.263	0.375	0.293	0.275	0.259	0.225	0.166
1,1,1	0.675	0.575	0.475	0.558	0.55	0.631	0.625	0.681	0.575	0.607
	R^2	***0.973***	R^2	***0.859***	R^2	***0.900***	R^2	***0.931***	R^2	***0.963***

Note. Each mean summarizes the performance of 20 perceptrons. The final row indicates the squared correlation (R^2) between the eight perceptron responses and the eight probabilities of reward for each of the training sets.

are reported later as well as for the different conditions in the human experiments that are described in Chapter 7.

Using a methodology that paralleled a previous simulation study (Dawson & Gupta, 2017), we constructed five training sets. Each training set consisted of 320 input patterns; we replicated each of the eight possible cue configurations 40 times in a training set. Each input pattern was either rewarded (desired output activity = 1) or not (desired output activity = 0). We determined the reward for each pattern using the following procedure: First, each input pattern in the training set signals a particular probability, as is shown in the first two columns of Table 4-8. Under the assumption that each cue is conditionally independent of the others, and that there is no reward when no cues are present, the three assigned probabilities, that is, $P(R|A) = 0.14$, $P(R|B) = 0.20$, $P(R|C) = 0.40$, can be used to compute the probability of reward for each input pattern. This is done by the additional rules for probabilities, which are applicable when conditional independence is true (Rozanov, 1977). The probability equations are provided in the third column of Table 5-1; the expected value for each equation is provided in Table 5-1, Column 4. With these expected probabilities in hand, for each training set, rewards were determined stochastically. This was done by taking an input pattern and computing the expected probability of reward for that pattern. Then a random number was selected. If this number was greater than the expected probability, then the pattern was not rewarded. Otherwise, it was rewarded. This procedure was repeated for each of the 320 patterns in a training set.

The stochastic procedure just described is not guaranteed to generate reward probabilities that are identical to the expected probabilities determined by the Table 5-1 equations. The final five columns in Table 5-1 provide the actual probability of reward for each of the eight possible input patterns. Although these columns are not exact matches to the expected column, each is highly correlated with it, with each correlation being over 0.98. We used chi-square tests to compare the probability of reward for each type of pattern in a training set to the expected probability generated by the addition rule for multiple cues. None of the training sets differed significantly from the expected values.

5.1.3 Perceptron Training

Following previous methodology (Dawson & Gupta, 2017), we trained 20 different perceptrons on each of the five training sets using a gradient descent rule with a learning rate of 0.05, with connection weights randomly set in the range from −0.1 to 0.1 prior to training, and with the bias θ of the logistic activation function initialized to 0. Training was accomplished with the gradient descent rule that was introduced in Chapter 1. During one epoch of training, we presented a network each of the 320 patterns; the learning rule modified connection weights and the bias after each pattern presentation. We randomized the order of input pattern presentations every epoch. Training proceeded for 2,500 epochs; we then recorded network responses to each of the eight possible input patterns, as well as the structure of the perceptron.

5.1.4 Perceptron Performance

An examination of the 20 perceptrons trained on the same training set revealed very high degrees of similarity between them, both in terms of their responses to each type of stimulus and in terms of their internal structure. Because of this similarity, Table 5-2 provides the mean response of the 20 perceptrons to each type of stimulus for each of the five training sets. Table 5-2 demonstrates that perceptron responses provide excellent estimates of the probability structure of the training sets. In Table 5-2, the actual probability of reward for each stimulus is provided for each training set; beside these actual probabilities are the average perceptron responses to each stimulus type. There is a strong correspondence between actual proba-

Table 5-3. The comparison between components of a logistic regression (Logit) and the average structure of a perceptron (Network) for each of the five conditionally independent three-cue training sets.

	Training Set 1		**Training Set 2**		**Training Set 3**		**Training Set 4**		**Training Set 5**	
Source	Logit	Network	Logit	Network	Logit	Network	Logit	Network	Logit	Network
θ	-2.680	-2.488	-2.296	-2.045	-2.381	-2.104	-2.466	-2.221	-2.637	-2.376
w	0.710	0.605	0.569	0.447	0.437	0.357	0.381	0.370	0.295	0.199
w	0.780	0.688	0.701	0.565	1.024	0.867	0.989	0.798	0.730	0.561
w	1.988	1.884	1.397	1.265	1.569	1.417	1.942	1.813	2.177	2.050
	R^2	**1.000**	R^2	**0.999**	R^2	**1.000**	R^2	**0.999**	R^2	**0.998**

Note. The final two rows indicate the squared correlation (R^2) between perceptron structure and logistic regression coefficients.

bilities and perceptron responses, indicating excellent probability estimation has been learned by the networks.

To assess the quality of probability estimation, we computed the squared correlation (R^2) between the eight perceptron responses and the eight actual probabilities of reward. This measure is reported for each training set in the final row of Table 5-2. It supports the conclusion that the perceptrons are excellent probability matchers. The worst performance, for Training Set 2, produces an R^2 of .859, indicating that perceptron responses capture almost 86% of the variance in the expected probabilities. The best performance, for Training Set 1, produces an R^2 of .986, indicating that perceptron responses capture almost 99% of the variance in the expected probabilities.

The perceptron performance provided in Table 5-2 indicates that poorest probability estimation occurred when no cues were present. This overestimation of the actual probability, which is determined exclusively by the perceptron's bias, was observed in each of the 100 perceptrons and replicates previous results when perceptrons are trained on four independent cues (Dawson & Gupta, 2017).

5.1.5 Perceptron Structure

Previous work has shown that there exists formal equivalence between a perceptron that uses the logistic activation function and logistic regression (Schumacher et al., 1996). We saw this equivalence in our earlier formal analysis of the perceptron when we determined that the weights and bias of a multiple cue perceptron were the same type of odds ratios as are the coefficients in logistic regression (Equations 4-14, 4-15, and 4-16). Schumacher et al. also proved that, in principle, if we train such a perceptron using gradient descent, then its weights should be identical to the coefficients of a logistic regression of the same data. However, they also noted that, in practice, gradient descent training may not provide this result, particularly if it uses a constant learning rate. Furthermore, Schumacher et al.'s proof defined the error gradient over the sum of a network's responses to all of the patterns in the training set (so-called batch training). Given that the perceptrons described in this chapter use a fixed learning rate and that they also learn via stochastic training which updates weights after every single pattern presentation, whether these perceptrons estimate probabilities by converging on the same solution as logistic regression is an open question.

To answer this question, we performed a logistic regression for each of the training sets, using the *glm* function in R. Each logistic regression fit a model derived from the 2×2×2 contingency table for each training set. We then used the SSE and R^2 to assess the relationship between the four coefficients of the logistic regression and the four characteristics of network structure (i.e., the bias of the output unit and the connection weight associated with each cue). The coefficients obtained from logistic regression, and the average bias and weights of a perceptron, are provided in Table 5-3. The bottom row of this table provides the quantitative comparison of each pair of columns. An examination of Table 5-3 reveals that the structure of each perceptron is very similar to the set of coefficients of a logistic regression performed on the same data. In short, each perceptron matches training set probabilities by adopting a structure that implements a logistic regression that maps cue signals into expected reward probabilities.

5.1.6 Discussion

In Chapter 4, we conducted a formal analysis of perceptrons that learn to assert probabilities using two cues. We established that such perceptrons are naïve Bayesian mechanisms and that their weights and bias reflect the natural logarithm of odds ratios, where each of these weights uses this value to represent the effect of a cue while the effects of all other cues are held constant. We also established that these perceptrons are capable of estimating probabilities perfectly when the probabilities asserted by cues are conditionally independent but would be less accurate in estimating probabilities when conditional independence between cues was not true.

The formal analysis presented in Chapter 4 could be extended for any number of cues, leading to identical conclusions. Rather than proceed with such proofs, we conducted a simulation study in which perceptrons are trained on a three-cue task, where each cue signals a different probability of reward, and the three cues were conditionally independent. The purpose of collecting these data is to permit comparisons to different simulations to be reported in following sections and comparisons to human performance later in this monograph. However, an examination of these simulation results should confirm the predictions that would be derived had a formal analysis of a three-cue perceptron been conducted.

The results of the simulations do indeed support these predictions. First, if the three-cue perceptron is a naïve Bayesian mechanism, and learns about the probability of reward signaled by conditionally independent cues, then we would expect it to estimate probabilities successfully. This is in accordance with our previous analysis. Our results support this prediction (Table 5-2). After 2,500 epochs of training, each of the 100 perceptrons that we examined achieved a high degree of accuracy for its probability estimations.

Table 5-4. The probability of reward for each of the eight possible cue states of a training set in which the probability of reward is signaled by the presence of Cue A or by the AND of Cues B and C. See text for details.

Input Pattern	Probability	Expected	Set 1	Set 2	Set 3	Set 4	Set 5
(0,0,0)	P(R|~A~B~C)	0.00	0.00	0.00	0.00	0.00	0.00
(0,0,1)	P(R|~A~BC)	0.00	0.00	0.00	0.00	0.00	0.00
(0,1,0)	P(R|~AB~C)	0.00	0.00	0.00	0.00	0.00	0.00
(0,1,1)	P(R|~ABC)	0.48	0.475	0.45	0.45	0.425	0.525
(1,0,0)	P(R|A~B~C)	0.48	0.5	0.45	0.425	0.425	0.475
(1,0,1)	P(R|A~BC)	0.48	0.55	0.4	0.425	0.475	0.50
(1,1,0)	P(R|AB~C)	0.48	0.45	0.45	0.45	0.50	0.50
(1,1,1)	P(R|ABC)	0.7296	0.725	0.75	0.725	0.75	0.70

Second, if the three-cue perceptron is a naïve Bayesian mechanism, and learns about the probability of reward signaled by conditionally independent cues, then we would expect that the values of its bias and connection weights would equal the natural logarithm of particular odds ratios. This prediction was confirmed by comparing the weights and biases of trained perceptrons to the coefficients of logistic regressions performed on the same training sets. As the coefficients of logistic regression are the same natural logarithms of odds ratios that are predicted to be found in the network structure, we should find a strong relationship between regression coefficients and the values of network components. This prediction was also confirmed in the simulation study (Table 5-3).

Other conclusions that were established formally for two-cue perceptrons should also be true when three cues are used to signal reward. In particular, performance of these perceptrons should decrease when conditional dependence between cues is introduced. The next section explores this prediction by examining the performance and structure of perceptrons for which the probability of reward is signaled (in part) by an interaction between cues: the logical AND of cues B and C.

5.2 Probability Estimation With Interactions: High-Reward AND

The simulations described in Section 5.1 demonstrated that perceptrons accurately estimate probabilities when the likelihood of reward is signaled by three conditionally independent cues. We now begin an exploration of perceptron limitations by considering perceptron performance when an interaction between cues is present. One aspect of the interactions that we study is their logical structure: We create an interaction between two cues that is defined by a logical operation that is either linearly separable (AND) or linearly nonseparable (XOR). A second aspect of these interactions is the amount of conditional dependence they contribute, which is manipulated by varying the probability of reward associated with the interaction.

To begin, let us consider a task in which the probability of reward is signaled by the presence of Cue A, or by the logical AND of Cues B and C. This latter signal involves an interaction between Cues B and C. It therefore should pose some difficulty for a perceptron that attempts to match probabilities using these signals. Furthermore, in this first set of simulations the probability of reward associated with the AND of Cues B and C is set high, producing a higher amount of conditional dependence than is present in a later simulation involving this interaction.

5.2.1 Network Architecture and Training Set

The perceptrons described in Section 5.2 have the identical structure as those discussed in Section 5.1. The coding of stimuli is identical to the coding described earlier in Section 5.1.2, because the same set of stimuli is used to train the perceptrons. The difference between the current perceptrons and those presented earlier is the set of rules used to determine probability of reward signaled by the cues. That is, although these perceptrons are provided the same stimuli as were those described in Section 5.1, these stimuli are associated with different probabilities of reward.

In this new set of simulations, the presence of Cue A indicated a reward probability of 0.48. In addition, the logical AND of Cues B and C signaled a reward probability of 0.48. These reward probabilities were selected to ensure that the overall likelihood of reward—the probability of reward across all of the training patterns—was approximately 0.33. This overall likelihood of reward was the same as that for the Section 5.1 simulations. This is called the High-Reward AND condition because the reward associated with the AND of the two cues is higher than is the case in another condition (Low-Reward AND) that is described later in the chapter.

Table 5-5. The probability of reward (P(R)) and the mean responses of perceptrons after being trained on the three-cue training sets that include the AND relation of Cues B and C.

	Training Set 1		**Training Set 2**		**Training Set 3**		**Training Set 4**		**Training Set 5**	
Input	P(R)	Mean	P(R)	Mean	P(R)	Mean	P(R)	Mean Response	P(R)	Mean
0,0,0	0.000	0.061	0.000	0.050	0.000	0.049	0.000	0.046	0.000	0.062
0,0,1	0.000	0.157	0.000	0.122	0.000	0.125	0.000	0.117	0.000	0.153
0,1,0	0.000	0.122	0.000	0.138	0.000	0.133	0.000	0.126	0.000	0.152
0,1,1	0.475	0.285	0.450	0.298	0.450	0.298	0.425	0.282	0.525	0.328
1,0,0	0.500	0.338	0.450	0.263	0.425	0.257	0.425	0.288	0.475	0.309
1,0,1	0.550	0.593	0.400	0.485	0.425	0.487	0.475	0.525	0.500	0.548
1,1,0	0.450	0.521	0.450	0.521	0.450	0.506	0.500	0.544	0.500	0.548
1,1,1	0.725	0.757	0.750	0.742	0.725	0.738	0.750	0.766	0.700	0.767
	R^2	**0.815**	R^2	**0.811**	R^2	**0.826**	R^2	**0.873**	R^2	**0.791**

Note. Each mean summarizes the performance of 20 perceptrons. The final row indicates the squared correlation (R^2) between the eight perceptron responses and the eight probabilities of reward for each of the training sets.

Table 5-6. The comparison between components of a logistic regression (Logit) and the average structure of a perceptron (Network) for each of the five three-cue training sets that involve an AND interaction between Cue B and C.

	Training Set 1		**Training Set 2**		**Training Set 3**		**Training Set 4**		**Training Set 5**	
Source	Logit	Network	Logit	Network	Logit	Network	Logit	Network	Logit	Network
θ	-3.318	-2.731	-3.525	-2.950	-3.523	-2.957	-3.630	-3.025	-3.264	-2.714
w	2.484	2.057	2.388	1.918	2.357	1.894	2.585	2.119	2.322	1.909
w	0.942	0.759	1.271	1.118	1.233	1.085	1.261	1.084	1.144	0.998
w	1.242	1.049	1.120	0.973	1.157	1.012	1.183	1.006	1.144	0.999
	R^2	**1.000**	R^2	**0.999**	R^2	**0.999**	R^2	**1.000**	R^2	**1.000**

Note. The final row indicates the squared correlation (R^2) between perceptron structure and logistic regression coefficients.

We constructed five training sets. Each training set consisted of 320 different input patterns; we replicated each of the eight possible cue configurations 40 times in a training set. Each input pattern was either rewarded (desired output activity = 1) or not (desired output activity = 0). The expected probability of reward for each type of stimulus is provided in Table 5-4. Note that the final expected probability in the table is computed using the addition rule ($P(R|A) + P(R|B \cap C) - P(R|A) \bullet P(R|B \cap C)$).

For each of the five training sets, we used the Section 5.1.2 stochastic procedure to determine whether a pattern was rewarded, but this time with the new set of expected probabilities provided in Table 5-4. The final five columns in Table 5-4 provide the actual probability of reward for each of the eight possible input patterns in each training set. Although these columns are not exact matches to the expected column, each is highly correlated with it, with each correlation being over 0.98. We used chi-square tests to compare the probability of reward for each type of pattern in a training set to the expected probability generated by the addition rule for multiple cues. None of the training sets differed significantly from the expected values.

5.2.2 Perceptron Training

Once again, we trained 20 different perceptrons on each of the five training sets using a gradient descent rule with a learning rate of 0.05, with connection weights randomly set in the range from −0.1 to 0.1 prior to training, and with the bias θ of the logistic activation function initialized to 0. During one epoch of training, we presented a network each of the 320 patterns: the learning rule modified connection weights and the bias after each pattern presentation. We randomized the order of input pattern presentations every epoch. Training proceeded for 2,500 epochs; we then recorded network responses to each of the eight possible input patterns, as well as the structure of the perceptron.

5.2.3 Perceptron Performance

An examination of the 20 perceptrons trained on the same training set revealed very high degrees of similarity between them, in terms of both their responses to each type of stimulus and their internal structure. Because of this similarity, Table 5-5 provides the mean response of the 20 perceptrons to each type of stimulus for each of the five training sets. When Table 5-5 is compared to Table 5-2 in Section 5.1, it becomes apparent that the Table 5-5 perceptrons did not match probabilities as well. The bottom row of Table 5-5 reports the same measure of goodness of fit (squared correlation) that was also reported in Table 5-2. This measure indicates quite clearly that when probability of reward is signaled in part by an interaction between cues, performance is poorer. For instance, the squared correlations in Table 5-5 indicate that the responses of these perceptrons typically capture only about 81% of the variance of the actual probabilities of reward. In contrast, most of the perceptrons reported in Table 5-2 captured well over 90% of the variance in actual probabilities. The worst perceptron reported in Section 5.2 (trained on Training Set 2) performed better than all but one of the perceptrons (the one trained on Training Set 4) reported in Table 5-5. A detailed discussion of the relative differences in performance between perceptrons is presented later in in this chapter.

5.2.4 Perceptron Structure

Table 5-5 demonstrates that the perceptrons faced with estimating reward probabilities when the AND of Cues B and C signals reward have poorer probability estimation performance than when this type of interaction is not present (i.e., the simulations reported in Section 5.1). Important to note, this is not because the structure of these perceptrons departs from expectations. To investigate perceptron structure, we again performed a logistic regression for each of the training sets, using the *glm* function in *R*. Each logistic regression fit a model derived from the 2×2×2 contingency table for each training set. We then used R^2 to assess the relationship between the four coefficients of the logistic regression and the four characteristics of network structure (i.e., the bias of the output unit and the connection weight associated with each cue). The coefficients obtained from logistic regression, and the average bias and weights of a perceptron, are provided in Table 5-6. The bottom row of this table provides the quantitative comparison of each pair of columns. These comparisons reveal that the structure of each perceptron is very similar to the set of coefficients of a logistic regression performed on the same data.

5.2.5 Discussion

The main results of this second set of simulations are again consistent with expectations derived from our formal analyses in Chapter 4. First, if the three-cue perceptron is a naïve Bayesian mechanism and receives signals from conditionally dependent cues, then we expect that it will not match probabilities correctly. This is because a naïve Bayesian mechanism is blind to interactions between input cues. Our results support this prediction (Table 5-5). After 2,500 epochs of training, each of the 100 perceptrons that we examined achieved a reasonably high degree of accuracy in its probability estimation performance, but this performance was not perfect.

Second, even when some cues are conditionally dependent, we expect that the perceptron will still be able to produce the best naïve Bayesian fit to the data, even if this type of prediction cannot capture all of the variability of the training set. This is confirmed by the strong relationship between network structure (biases and weights) and the coefficients of logistic regression. These logistic

Table 5-7. The probability of reward for each of the eight possible cue states of a training set in which the probability of reward is signaled by the presence of Cue A or by the XOR of Cues B and C. See text for details.

Input Pattern	Probability	Expected	Set 1	Set 2	Set 3	Set 4	Set 5
(0,0,0)	P(R|~A~B~C)	0.00	0.00	0.00	0.00	0.00	0.00
(0,0,1)	P(R|~A~BC)	0.36	0.4	0.35	0.35	0.325	0.325
(0,1,0)	P(R|~AB~C)	0.36	0.35	0.35	0.45	0.35	0.375
(0,1,1)	P(R|~ABC)	0.00	0.00	0.00	0.00	0.00	0.00
(1,0,0)	P(R|A~B~C)	0.37	0.4	0.325	0.425	0.35	0.4
(1,0,1)	P(R|A~BC)	0.5968	0.575	0.55	0.65	0.675	0.6
(1,1,0)	P(R|AB~C)	0.5968	0.65	0.625	0.625	0.65	0.6
(1,1,1)	P(R|ABC)	0.37	0.375	0.325	0.425	0.4	0.45

Table 5-8. The probability of reward (P(R)) and the mean responses of perceptrons after being trained on the three-cue training sets that include the XOR relation of Cues B and C.

	Training Set 1		**Training Set 2**		**Training Set 3**		**Training Set 4**		**Training Set 5**	
Input	P(R)	Mean	P(R)	Mean	P(R)	Mean	P(R)	Mean Response	P(R)	Mean
0,0,0	0.000	0.189	0.000	0.172	0.000	0.199	0.000	0.159	0.000	0.166
0,0,1	0.400	0.171	0.350	0.151	0.350	0.188	0.325	0.171	0.325	0.167
0,1,0	0.350	0.197	0.350	0.193	0.450	0.210	0.350	0.165	0.375	0.182
0,1,1	0.000	0.179	0.000	0.170	0.000	0.198	0.000	0.178	0.000	0.184
1,0,0	0.400	0.505	0.325	0.452	0.425	0.530	0.350	0.504	0.400	0.498
1,0,1	0.575	0.475	0.550	0.414	0.650	0.512	0.675	0.526	0.600	0.501
1,1,0	0.650	0.518	0.625	0.488	0.625	0.546	0.650	0.515	0.600	0.527
1,1,1	0.375	0.487	0.325	0.449	0.425	0.528	0.400	0.537	0.450	0.530
	R^2	**0.502**	***R^2***	**0.457**	***R^2***	**0.512**	***R^2***	**0.554**	***R^2***	**0.597**

Note. Each mean summarizes the performance of 20 perceptrons. The row indicates the squared correlation (R^2) between the eight perceptron responses and the eight probabilities of reward for each of the training sets.

regressions provide the best fit that they can to the training sets but are also blind to interactions between cues. The logistic regressions fit the data to the best of their ability by using main effects, the size of which are reflected in the natural logarithms of odds ratios. We expect that the perceptrons will perform in essentially the same manner and that their structure will reflect the same odds ratios. The results of Table 5-6 confirm this expectation.

Although it is clear that the perceptrons described in this section have problems with the interaction, their performance is still satisfactory. It would appear that these perceptrons are capable of estimating, to a level just above 80%, the actual probabilities of reward reflected in the training set. In Section 5.3, we next turn to perceptrons faced with a different interaction between Cues B and C—the linearly nonseparable XOR of these two cues. Will these perceptrons have similar performance to the ones that we have just discussed?

5.3 Probability Estimation With Interactions: High-Reward XOR

5.3.1 Network Architecture and Training Set

This next set of perceptrons has the identical structure as those discussed in the previous two sections: They are modern perceptrons comprising a single output unit and three input units. The stimulus encoding is also identical to the coding that described earlier because the same set of stimuli is used to train these perceptrons. The difference between this simulation and the two described in Sections 5.1 and 5.2 is with respect to the probability of reward associated with each stimulus. In these simulations, the presence of Cue A indicated a reward probability of 0.36. In addition, the logical XOR of Cues B and C signaled a reward probability of 0.37. These reward probabilities were selected to ensure that the overall likelihood of reward—the probability of reward across all of the training patterns—was approximately 0.33. This overall likelihood of reward was the same as that for the previous simulations described in this chapter. This is called the High-Reward XOR condition because the reward associated with the XOR of the two cues is higher than is the case in another condition (Low-Reward XOR) that is described later in the chapter.

We constructed five training sets. Each training set consisted of 320 input patterns; we replicated each of the eight possible cue configurations 40 times in a training set. Each input pattern was either rewarded (desired output activity = 1) or not (desired output activity = 0). The expected probability of reward for each type of stimulus is provided in Table 5-7. Note that when both Cue A is present and the XOR of Cues B and C is true the expected probability in the table is computed using the addition rule ($P(R|A) + P(R|XOR(B,C)) - P(R|A) \bullet P(R|XOR(B,C))$).

For each of the five training sets that were constructed, we used the same stochastic procedure to determine whether a pattern was rewarded that was described in Section 5.1.2. However, this time we used a new set of expected probabilities provided in Table 5-7. The final five columns in Table 5-7 provide the actual probability of reward for each of the eight possible input patterns in each training set. Although these columns are not exact

Table 5-9. The comparison between components of a logistic regression (Logit) and the average structure of a perceptron (Network) for each of the five three-cue training sets that involve an XOR interaction between Cue B and C.

	Training Set 1		**Training Set 2**		**Training Set 3**		**Training Set 4**		**Training Set 5**	
Source	Logit	Network	Logit	Network	Logit	Network	Logit	Network	Logit	Network
θ	-1.436	-1.455	-1.552	-1.574	-1.388	-1.392	-1.659	-1.667	-1.615	-1.618
w	1.467	1.475	1.376	1.382	1.513	1.511	1.670	1.682	1.602	1.611
w	0.000	0.050	0.096	0.144	0.092	0.066	0.064	0.046	0.160	0.117
w	-0.062	-0.121	-0.096	-0.157	-0.092	-0.072	0.064	0.088	0.000	0.010
	R^2	**0.999**	***R^2***	**0.999**	***R^2***	**1.000**	***R^2***	**1.000**	***R^2***	**1.000**

Note. The final row indicates the squared correlation (R^2) that between perceptron structure and logistic regression coefficients.

matches to the expected column, each is highly correlated with it, with each correlation being greater than 0.98. We used chi-square tests to compare the probability of reward for each type of pattern in a training set to the expected probability generated by the addition rule for multiple cues. None of the training sets differed significantly from the expected values.

5.3.2 Perceptron Training and Performance

Once again, we trained 20 perceptrons on each of the five training sets using the same procedure that was described in the previous two sections. An examination of the 20 perceptrons trained on the same training set revealed very high degrees of similarity between them, both in terms of their responses to each type of stimulus and in terms of their internal structure. Because of this similarity, Table 5-8 provides the mean response of the 20 perceptrons to each type of stimulus for each of the five training sets. When Table 5-8 is compared to the earlier Tables 5-3 or 5-6, it becomes apparent that the Table 5-8 perceptrons did not estimate probabilities as well. The bottom three rows of Table 5-8 report the same measure of goodness of fit (squared correlation) that was also reported in those previous tables. These measures indicate quite clearly that when probability of reward is signaled in part by the XOR interaction between cues, performance is poorer. For instance, the squared correlations in Table 5-8 indicate that the responses of these perceptrons typically capture only between 46% and 60% of the variance of the actual probabilities of reward. The probability estimation ability of each of these perceptrons is poorer than the worst performance of any reported in Table 5-2 or in Table 5-5.

5.3.3 Perceptron Structure

Table 5-8 demonstrates that the perceptrons faced with estimating reward probabilities when the XOR of Cues B and C signals reward have poorer performance than do the perceptrons described earlier in this chapter. Important to note, this is once again not because the structure of these perceptrons departs from expectations. To investigate perceptron structure, we again performed a logistic regression for each of the training sets, using the *glm* function in R. Each logistic regression fit a model derived from the 2×2×2 contingency table for each training set. We then used R^2 to assess the relationship between the four coefficients of the logistic regression and the four characteristics of network structure (i.e., the bias of the output unit and the connection weight associated with each cue). The coefficients obtained from logistic regression, and the average bias and weights of a perceptron, are provided in Table 5-9. The bottom row of this table provides the quantitative comparison of each pair of columns. These comparisons reveal that there are nearly perfect correlations between logistic coefficients and network structures. An examination of Table 5-9 indicates, in fact, that the structure of these perceptrons is more similar to the logistic regression coefficients than was the structure of the perceptrons confronted with the AND of Cues B and C.

5.3.4 Discussion

The results of this set of simulations are again consistent with expectations derived from our formal analyses in Chapter 4. First, if the three-cue perceptron is a naïve Bayesian mechanism, and receives signals from conditionally dependent cues, then its probability estimates will not be accurate. Our results support this prediction (Table 5-8).

Second, even when some cues are conditionally dependent, we expect that the perceptron will still be able to produce the best naïve Bayesian fit to the data, even if this type of prediction cannot capture all of the variability of the training set. This is confirmed by the strong relationship between network structure (biases and weights) and the coefficients of logistic regression. These logistic regressions provide the best fit that they can to the training sets but are also blind to interactions between cues. The

Table 5-10. The probability of reward for each of the eight possible cue states of a training set in which the probability of reward is signaled by the presence of Cue A or by the XOR of Cues B and C. See text for details.

Input Pattern	Probability	Expected	Set 1	Set 2	Set 3	Set 4	Set 5
(0,0,0)	P(R|~A~B~C)	0	0	0	0	0	0
(0,0,1)	P(R|~A~BC)	0	0	0	0	0	0
(0,1,0)	P(R|~AB~C)	0	0	0	0	0	0
(0,1,1)	P(R|~ABC)	0.18	0.175	0.2	0.175	0.225	0.25
(1,0,0)	P(R|A~B~C)	0.6	0.5	0.6	0.65	0.675	0.6
(1,0,1)	P(R|A~BC)	0.6	0.525	0.525	0.625	0.575	0.6
(1,1,0)	P(R|AB~C)	0.6	0.6	0.575	0.65	0.675	0.575
(1,1,1)	P(R|ABC)	0.672	0.65	0.65	0.8	0.725	0.65

logistic regressions fit the data to the best of their ability by using main effects, the size of which are reflected in the natural logarithms of odds ratios. We expect that the perceptrons will perform in essentially the same manner and that their structure will reflect the same odds ratios. The results of Table 5-9 confirm this expectation.

Third, it is apparent that the performance of perceptrons that estimate probability in the face of the XOR interaction is much poorer than that of those that estimate probability in the face of the AND interaction. Given the traditional view of perceptrons as pattern classifiers, this result may seem unsurprising at first glance. This is because perceptrons cannot solve linearly separable problems (Minsky & Papert, 1969), AND is linearly separable, whereas XOR is not. Thus, it may not be surprising that the perceptrons faced with XOR are poorer performers.

However, there is a more fruitful interpretation of differences between perceptron performances. First, given that AND is linearly separable, why do the perceptrons faced with this interaction perform more poorly than do those faced with three conditionally independent cues? Second, given that both AND and XOR involve interactions between Cues B and C, why is one interaction more detrimental than the other? A better approach to considering differences between perceptron performances is to turn away from the logical structure of the interactions and instead consider the interactions in terms of degrees of conditional dependence (Dawson & Gupta, 2017). The next section provides a brief comparison of the different results provided by the three sets of simulations that we have been considering to this point and begins to tease out the different contributions of linear nonseparability and conditional dependence.

5.4 Manipulating Conditional Dependence: Low-Reward AND

In Section 5.4, we saw that perceptron performance—the ability of a perceptron to estimate the actual probability of reward in a training set—was significantly affected by the presence of an interaction between signals of reward probability. When an interaction was present, performance decreased. We also saw that the nature of this interaction was important: Perceptrons faced with an AND interaction performed significantly better than did perceptrons faced with an XOR interaction.

At face value, the effect of the type of interaction agrees with standard accounts of perceptron performance: Section 5.4 demonstrated that perceptrons performed better when faced with a linearly separable interaction than when faced with a linearly nonseparable interaction. The point of the remaining simulations reported in Chapter 5 is to show that linearly separability is not the most appropriate influence on perceptron performance when signals of reward are uncertain. Dawson and Gupta (2017) showed that one could manipulate the degree of conditional dependence in training sets like those discussed in Sections 5.1 and 5.2 by varying the probability of reward associated with the logical combination of cues. In other words, one can alter conditional dependence while holding the logical structure of an interaction constant. The simulations in this section and the next reveal that perceptron performance changes when such a manipulation is explored.

In this section, we describe another set of simulations. Each of these simulations involves an interaction between two cues (the AND of Cues B and C). However, in these simulations the likelihood of reward associated with the interaction is reduced, which decreases the amount of conditional dependence between the interacting cues. At issue is whether this manipulation also affects perceptron performance.

5.4.1 Network Architecture and Training Set

The next set of perceptrons has the identical structure as those discussed in the earlier sections: They are perceptrons comprising a single output unit and three input units. The stimulus encoding is also identical to the coding that described earlier because the same set of stimuli is used to train these perceptrons. The difference between the current perceptrons and those presented in Section 5.2 is the set of rules used to determine probability of reward signaled by the cues. In this study, the presence of Cue A indicated a reward probability of 0.6. In addition, the logical AND of Cues B and C signaled a reward probability of 0.18. These reward probabilities were selected to (a) lower the reward associated with Cues B and C while (b) ensuring that the overall likelihood of reward—the probability of reward across all of the training patterns—was approximately 0.33. This overall likelihood of reward is the same as that for the previous simulations. In other words, in comparison to the simulations reported in Section 5.2, there is a higher reward associated with Cue A and a lower reward associated with the interaction between Cues B and C.

We used the same stochastic procedure employed earlier to construct five training sets. Each training set consisted of 320 input patterns; we replicated each of the eight possible cue configurations 40 times in a training set. Each input pattern was either rewarded (desired output activity = 1) or not (desired output activity = 0). The expected probability of reward for each type of stimulus is provided in Table 5-10. Note that when both Cue A is present and the AND of Cues B and C is true the expected probability in the table is computed using the addition rule $(P(R|A) + P(R|B \cap C) - P(R|A) \bullet P(R|B \cap C))$.

The final five columns in Table 5-10 provide the actual probability of reward for each of the eight possible input patterns in each training set. Although these columns are not exact matches to the expected column, each is highly correlated with it, with each correlation being greater than 0.99. We used chi-square tests to compare the probability of reward for each type of pattern in a training set to the expected probability generated by the addition rule for multiple cues. None of the training sets differed significantly from the expected values.

The purpose of using the new set of reward probabilities (Table 5-10) was to decrease the conditional dependence in the training sets that is supplied by Cues B and C interacting. Table 5-11 shows that a decrease in the reward associated with the AND of these two cues accomplishes this goal. It uses the log-likelihood ratio test (i.e., *G*, as described in Section 4.3.1) to measure conditional dependence (Woolf, 1957).

Table 5-11. The degree of conditional dependence between Cues B and C in the training sets with a low reward associated with the AND of Cues B and C (P(R|BC) = 0.18) compared to the degree of conditional dependence between Cues B and C in the training sets with a high reward associated with the AND of Cues B and C (P(R|BC)=0.48).

	Low-Reward AND	High-Reward AND
Training Set 1	0.39	5.69
Training Set 2	1.40	8.35
Training Set 3	1.96	6.32
Training Set 4	1.59	4.60
Training Set 5	1.01	5.82
Mean	**1.27**	**6.16**
SE	**0.27**	**0.62**

Note. The G statistic, described in Section 4.3.1, is used to measure conditional dependence. The bottom two rows provide the mean and the standard error of each column.

To calculate *G*, a training set was split into four 2×2 contingency tables, where each of these tables crossed the state of Cue B (absent vs. present) with the state of Cue C (absent vs. present). The four contingency tables were created by considering Cues B and C in the context of two other variables: the state of Cue A (absent or present) and the state of the reward (presented or not). *G* was calculated by using Woolf's equation for each of these contingency tables and summing all of the computed values. Table 5-11 shows that this measure is lower for each of the low AND reward training sets in comparison to the high AND reward training sets that were described in Section 5.2. The last two rows of the table provide the mean value of *G* for each condition, along with the standard error of the mean. These values indicate that the average value of *G* is significantly lower in the Low-Reward AND condition than in the High-Reward AND condition, indicating that there is significantly less conditional dependence.

5.4.2 Perceptron Training and Performance

Once again, we trained 20 perceptrons on each of the five training sets using the same procedure used for the simulations described earlier in this chapter. An examination of the 20 perceptrons trained on the same training set revealed very high degrees of similarity between them, in terms of both their responses to each type of stimulus and their internal structure. Because of this similarity, Table 5-12 provides the mean response of the 20 perceptrons

Table 5-12. The probability of reward (P(R)) and the mean responses of perceptrons after being trained on the three-cue training sets that include the Low

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Input	P(R)	Mean	P(R)	Mean	P(R)	Mean	P(R)	Mean Response	P(R)	Mean
0,0,0	0.000	0.037	0.000	0.048	0.000	0.037	0.000	0.051	0.000	0.058
0,0,1	0.000	0.046	0.000	0.052	0.000	0.052	0.000	0.052	0.000	0.077
0,1,0	0.000	0.061	0.000	0.063	0.000	0.057	0.000	0.078	0.000	0.070
0,1,1	0.175	0.076	0.200	0.068	0.175	0.080	0.225	0.080	0.250	0.093
1,0,0	0.500	0.475	0.600	0.545	0.650	0.591	0.675	0.613	0.600	0.550
1,0,1	0.525	0.534	0.525	0.568	0.625	0.674	0.575	0.620	0.600	0.622
1,1,0	0.600	0.605	0.575	0.616	0.650	0.698	0.675	0.712	0.575	0.600
1,1,1	0.650	0.660	0.650	0.637	0.800	0.767	0.725	0.718	0.650	0.669
	R^2	**0.970**	R^2	**0.947**	R^2	**0.971**	R^2	**0.950**	R^2	**0.934**

Note. Each mean summarizes the performance of 20 perceptrons. The final row indicates the squared correlation (R^2) between the eight perceptron responses and the eight probabilities of reward for each of the training sets.

Table 5-13. The comparison between components of a logistic regression (Logit) and the average structure of a perceptron (Network) for each of the five three-cue training sets that involve a Low

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Source	Logit	Network	Logit	Network	Logit	Network	Logit	Network	Logit	Network
θ	-3.747	-3.263	-3.427	-2.997	-3.829	-3.269	-3.393	-2.924	-3.282	-2.788
w	3.453	3.161	3.351	3.177	3.977	3.637	3.582	3.384	3.212	2.987
w	0.715	0.527	0.525	0.291	0.699	0.470	0.693	0.447	0.473	0.206
w	0.450	0.240	0.351	0.094	0.600	0.356	0.325	0.029	0.558	0.298
	R^2	**0.998**	R^2	**0.994**	R^2	**0.997**	R^2	**0.993**	R^2	**0.993**

Note. The final two rows indicate the sum of squared error and squared correlation that compares perceptron structure perceptrons to logistic regression coefficients.

to each type of stimulus for each of the five training sets. When Table 5-12 is compared to the results for the High-Reward AND simulation, it is apparent that the Table 5-12 perceptrons were better probability matches. The bottom row of Table 5-12 reports the same measure of goodness of fit (squared correlation) that were also reported in that previous table. These measures indicate quite clearly that when probability of reward is signaled in part by the XOR interaction between cues, performance is better, because these numbers are higher. A detailed discussion of the relative differences in performance between perceptrons is presented later in Section 5.6.

5.4.3 Perceptron Structure

Table 5-12 demonstrates that the perceptrons faced with estimating reward probabilities when the reward associated with the AND of Cues B and C is lowered has better performance than the perceptrons trained on a higher reward version of the same problem (Section 5.2). Once again, this performance is the result of perceptron structure approximating the optimal odds ratios. We again performed a logistic regression for each of the training sets, using the *glm* function in R. Each logistic regression fit a model derived from the 2×2×2 contingency table for each training set. We then used R^2 to assess the relationship between the four coefficients of the logistic regression and the four characteristics of network structure (i.e., the bias of the output unit and the connection weight associated with each cue). The coefficients obtained from logistic regression, and the average bias and weights of a perceptron, are provided in Table 5-13. The bottom row of this table provides the quantitative comparison of each pair of columns. These comparisons reveal that the structure of each perceptron is very similar to the set of coefficients of a logistic regression performed on the same data. For instance, for each training set there are nearly perfect squared correlations between logistic coefficients and network structures.

Table 5-14. The probability of reward for each of the eight possible cue states of a training set in which the probability of reward is signaled by the presence of Cue A or by the XOR of Cues B and C. See text for details.

Input Pattern	Probability	Expected	Set 1	Set 2	Set 3	Set 4	Set 5
(0,0,0)	P(R|~A~B~C)	0	0	0	0	0	0
(0,0,1)	P(R|~A~BC)	0.18	0.175	0.175	0.225	0.125	0.175
(0,1,0)	P(R|~AB~C)	0.18	0.2	0.2	0.175	0.175	0.225
(0,1,1)	P(R|~ABC)	0	0	0	0	0	0
(1,0,0)	P(R|A~B~C)	0.53	0.55	0.6	0.525	0.525	0.45
(1,0,1)	P(R|A~BC)	0.6146	0.65	0.65	0.625	0.625	0.625
(1,1,0)	P(R|AB~C)	0.6146	0.675	0.6	0.725	0.6	0.625
(1,1,1)	P(R|ABC)	0.53	0.525	0.55	0.525	0.5	0.5

5.4.4 Discussion

The results of this set of simulations are again consistent with expectations derived from our formal analyses in Chapter 4. First, if the three-cue perceptron is a naïve Bayesian mechanism, and receives signals from conditionally dependent cues, then it will not be able to match probabilities perfectly. Our results support this prediction (Table 5-12). After 2,500 epochs of training, each of the 100 perceptrons that we examined achieved a high degree of accuracy in its probability estimation performance, but performance is poorer than that which is achieved when no conditional dependence is present.

Second, even when some cues are conditionally dependent, we expect that the perceptron will still be able to produce the best naïve Bayesian fit to the data, even if this type of prediction cannot capture all of the variability of the training set. This is confirmed by the strong relationship between network structure (biases and weights) and the coefficients of logistic regression. These logistic regressions provide the best fit that they can to the training sets but are also blind to interactions between cues. The logistic regressions fit the data to the best of their ability by using main effects, the size of which are reflected in the natural logarithms of odds ratios. We expect that the perceptrons will perform in essentially the same manner and that their structure will reflect the same odds ratios. The results of Table 5-13 confirm this expectation.

Third, it is apparent that the performance of perceptrons that probability match in the face of the lower reward associated with the AND of Cues B and C is better than that observed for perceptrons trained on the high-reward version of the same training set. A more detailed comparison of perceptron performance, one that examines the different influences of conditional dependence and linear separability, is provided later in Section 5.6.

5.5 Manipulating Conditional Dependence: Low-Reward XOR

5.5.1 Network Architecture and Training Set

This final set of perceptrons described has the identical structure as those discussed in the previous sections: They are modern perceptrons comprising a single output unit and three input units. The stimulus encoding is also identical to the coding that described earlier, because the same set of stimuli is used to train these perceptrons. The difference between the current perceptrons and those presented in Section 5.3 is the set of rules used to determine probability of reward signaled by the cues. In particular, there is a lower probability of reward signaled by the XOR of Cues B and C. In this last simulation, the presence of Cue A indicated a reward probability of 0.53. In addition, the logical XOR of Cues B and C signaled a reward probability of 0.18. These reward probabilities were selected to (a) lower the reward associated with Cues B and C while (b) ensuring that the overall likelihood of reward—the probability of reward across all of the training patterns—was approximately 0.33. This overall likelihood of reward was the same as that for the earlier simulations. In other words, in comparison to the simulations reported in Section 5.3, there is a higher reward associated with Cue A and a lower reward associated with the interaction between Cues B and C.

We used the same stochastic procedure described earlier to construct five training sets. Each training set consisted of 320 input patterns; we replicated each of the eight possible cue configurations 40 times in a training set. Each input pattern was either rewarded (desired output activity = 1) or not (desired output activity = 0). The expected probability of reward for each type of stimulus is provided in Table 5-14. Note that when both Cue A is present and the XOR of Cues B and C is true, the expected probability in the table is computed using the addition rule ($P(R|A) + P(R|XOR(B,C)) - P(R|A) \bullet P(R|\ XOR(B,C))$).

The final five columns in Table 5-14 provide the actual probability of reward for each of the eight possible input patterns in each training set. Although these columns are not exact matches to the expected column, each is highly correlated with it, with each correlation being over 0.99. We used chi-square tests to compare the probability of reward for each type of pattern in a training set to the expected probability generated by the addition rule for multiple cues. None of the training sets differed significantly from the expected values.

The purpose of using the new set of reward probabilities (Table 5-14) was to decrease the conditional dependence in the training sets that is supplied by Cues B and C interacting. Table 5-15 shows that a decrease in the reward associated with the XOR of these two cues accomplishes this goal.

It operationalizes the conditional dependence due to the interaction of Cues B and C as the G statistic following the same procedure that was described in Section 5.5.1. The means (and their standard errors) at the bottom of Table 5-15 indicate that there is significantly lower conditional dependence in the Low-Reward XOR condition than in the High-Reward XOR condition.

5.5.2 Perceptron Training and Performance

Once again, we trained 20 perceptrons on each of the five training sets using the same procedure described for the simulations reported earlier in this chapter. An examination of the 20 perceptrons trained on the same training set revealed very high degrees of similarity between them, in terms of both their responses to each type of stimulus and their internal structure. Because of this similarity, Table 5-16 provides the mean response of the 20 perceptrons to each type of stimulus for each of the five training sets. When Table 5-16 is compared to the results for the High-Reward XOR simulation, it is apparent that the Table 5-16 perceptrons are better probability estimators. The bottom row of Table 5-16 reports the same measure of goodness of fit (squared correlation) that was also reported in that earlier table. This measure indicates quite clearly that when probability of reward due to the XOR interaction between cues is lowered, performance is better, because these numbers are higher. A detailed dis-

Table 5-15. The degree of conditional dependence between Cues B and C in the training sets with a low reward associated with the XOR of Cues B and C (P(R|BC) = 0.18) compared to the degree of conditional dependence between Cues B and C in the training sets with a high reward associated with the XOR of Cues B and C (P(R|BC)=0.48).

	Low-Reward XOR	High-Reward XOR
Training Set 1	24.90	56.66
Training Set 2	22.70	56.06
Training Set 3	27.57	59.34
Training Set 4	18.91	56.37
Training Set 5	27.37	49.61
Mean	**24.29**	**55.61**
SE	**1.61**	**1.61**

Note. Conditional dependence is measured using G. The final two rows provide the mean and the standard error of the mean of each column.

Table 5-16. The probability of reward (P(R)) and the mean responses of perceptrons after being trained on the three-cue training sets that include the Low

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Input	P(R)	Mean	P(R)	Mean	P(R)	Mean	P(R)	Mean Response	P(R)	Mean
0,0,0	0.000	0.098	0.000	0.099	0.000	0.096	0.000	0.078	0.000	0.092
0,0,1	0.175	0.089	0.175	0.099	0.225	0.083	0.125	0.076	0.175	0.096
0,1,0	0.200	0.100	0.200	0.085	0.175	0.111	0.175	0.073	0.225	0.102
0,1,1	0.000	0.091	0.000	0.084	0.000	0.096	0.000	0.072	0.000	0.107
1,0,0	0.550	0.614	0.600	0.619	0.525	0.598	0.525	0.576	0.450	0.529
1,0,1	0.650	0.589	0.650	0.617	0.625	0.559	0.625	0.570	0.625	0.541
1,1,0	0.675	0.621	0.600	0.577	0.725	0.636	0.600	0.560	0.625	0.559
1,1,1	0.525	0.595	0.550	0.576	0.525	0.598	0.500	0.554	0.500	0.571
	R^2	**0.910**	R^2	**0.930**	R^2	**0.885**	R^2	**0.933**	R^2	**0.864**

Note. Each mean summarizes the performance of 20 perceptrons. The final row indicates squared correlation (R^2) between the eight perceptron responses and the eight probabilities of reward for each of the training sets.

cussion of the relative differences in performance between perceptrons is presented in Section 5.6.

5.5.3 Perceptron Structure

To compare the structure of perceptrons to the optimal odds ratios revealed by logistic regression, we again performed a logistic regression for each of the training sets, using the *glm* function in *R*. Each logistic regression fit a model derived from the 2×2×2 contingency table for each training set. We then used R^2 to assess the relationship between the four coefficients of the logistic regression and the four characteristics of network structure (i.e., the bias of the output unit and the connection weight associated with each cue). The coefficients obtained from logistic regression, and the average bias and weights of a perceptron, are provided in Table 5-17. The bottom row of this table provides the quantitative comparison of each pair of columns. These comparisons reveal that the structure of each perceptron is very similar to the set of coefficients of a logistic regression performed on the same data. For instance, for each training set there are nearly perfect correlations between logistic coefficients and network structures.

5.5.4 Discussion

The results of this final set of simulations are once again consistent with expectations derived from our formal analyses in Chapter 4. First, if the three-cue perceptron is a naïve Bayesian mechanism, and receives signals from cues that are conditionally dependent, then it will not be able to match probabilities perfectly. Our results support this prediction (Table 5-16). After 2,500 epochs of training, each of the 100 perceptrons that we examined achieved a reasonable degree of accuracy in its probability estimation performance, but performance is poorer than that which is achieved when no conditional dependence is present.

Second, even when some cues are conditionally dependent, we expect that the perceptron will still be able to produce the best naïve Bayesian fit to the data, even if this type of prediction cannot capture all of the variability of the training set. This is confirmed by the strong relationship between network structure (biases and weights) and the coefficients of logistic regression. These logistic regressions provide the best fit that they can to the training sets but are also blind to interactions between cues. The logistic regressions fit the data to the best of their ability by using main effects, the size of which are reflected in the natural logarithms of odds ratios. We expect that the perceptrons will perform in essentially the same manner and that their structure will reflect the same odds ratios. The results of Table 5-17 confirm this expectation.

Third, it is apparent that the performance of perceptrons that probability match in the face of the lower reward associated with the XOR of Cues B and C is better than that observed for perceptrons trained on the high-reward version of the same training set. A more detailed comparison of perceptron performance, one that examines the different influences of conditional dependence and linear separability, is provided in Section 5.6.

5.6 Linear Separability versus Conditional Dependence

Sections 5.2, 5.3, 5.4, and 5.5 presented the results of four different simulations that investigated the effects of conditional dependence on the ability of perceptrons to match probabilities. When considered together, these four simulations represent the four conditions of a two-factor experiment. One factor concerns the linear separability of the interaction being used to create conditional dependence. For some simulations (those that used the AND of Cues B and C), this was linearly separable. For other simulations (those that used the XOR of Cues B and C), this was linearly nonseparable. The other factor concerns the level of reward that was used to manipulate conditional

Table 5-17. The comparison between components of a logistic regression (Logit) and the average structure of a perceptron (Network) for each of the five three-cue training sets that involve a Low

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Source	Logit	Network	Logit	Network	Logit	Network	Logit	Network	Logit	Network
θ	-2.232	-2.224	-2.193	-2.208	-2.198	-2.239	-2.473	-2.474	-2.274	-2.290
w	2.676	2.690	2.676	2.692	2.604	2.638	2.764	2.779	2.400	2.407
w	0.038	0.027	-0.115	-0.174	0.076	0.158	0.000	-0.065	0.148	0.119
w	-0.115	-0.106	-0.038	-0.005	-0.076	-0.161	-0.079	-0.022	0.000	0.048
	R^2	**1.000**	R^2	**1.000**	R^2	**0.999**	R^2	**0.999**	R^2	**1.000**

Note. The final two rows indicate the sum of squared error and squared correlation that compares perceptron structure perceptrons to logistic regression coefficients.

dependence. For some simulations, a high level of reward was used to introduce higher conditional dependence. For other simulations, a low level of reward was used to introduce lower conditional dependence.

The purpose of this section is to take advantage of this factorial design to investigate the relative importance of linear separability and conditional dependence in terms of their effects on the ability of perceptrons to match probabilities when learning about the likelihood of reward signaled by uncertain cues.

5.6.1 Analysis of Variance

To begin, let us consider the four simulations from the basic perspective of a two-factor analysis of variance (ANOVA). Table 5-18 provides the basic data for this by presenting the average measure of probability estimation for the 100 different perceptrons in each of the four conditions defined by this two-factor design.

Table 5-18. The mean probability estimation performance (standard deviations in parentheses) of perceptrons as a function of problem type and level of reward.

	High-Reward	Low-Reward
AND of Cues B and C	0.82	0.95
XOR of Cues B and C	0.52	0.90

Note. Probability estimation performance is operationalized as the squared correlation between the perceptrons responses for each of eight stimuli and the actual reward probabilities for these same stimuli. Each mean is based on the performance of 100 perceptrons.

An inspection of Table 5-18 reveals several interesting findings. First, on average, networks perform better when there is an AND relationship between Cues B and C than when there is an XOR relationship between these cues. Second, on average, networks perform better when Cues B and C combine to signal a lower probability of reward than when they combine to signal a higher probability of reward. Third, changing the probability of reward has a much larger effect on network performance for the XOR versions of the training sets than for AND versions of the training sets.

An ANOVA of the data used to produce Table 5-18 confirms these general observations. We conducted an ANOVA on the *R*2 scores for the networks in different conditions to test the effects of interaction type and size of reward. This analysis revealed a significant main effect of type, $F(1, 396) = 3046$, $p < 2\text{e-}16$, $\eta 2 = 0.264$; a significant main effect of reward, $F(1, 396) = 6543$, $p < 2\text{e-}16$, $\eta 2 = 0.567$; and a significant Type × Reward interaction, $F(1, 396) = 1548$, $p < 2\text{e-}16$, $\eta 2 = 0.134$. Post hoc tests conducted using the Tukey honestly significant difference (HSD) statistic revealed that perceptrons trained on the AND interaction learned to be more accurate probability estimators than did perceptrons trained on the XOR interaction ($p < 1.0\text{e-}16$). As well, perceptrons trained in conditions in which the interacting cues signaled low reward were more accurate probability estimators than network trained in conditions in which the interacting cues signaled high reward ($p < 1.0\text{e-}16$). The significant interaction emerged because the Low-Reward XOR networks performed significantly better than did the High-Reward XOR networks (difference between conditions = 0.441), but there was a smaller effect of different reward levels on the AND networks (difference between conditions = 0.120). However, both of these differences were statistically significant ($p < 1.0\text{e-}16$). All of these effects replicate those reported by Dawson and Gupta (2017) when they trained perceptrons on stimuli defined using four cues.

5.6.2 Predicting Performance From G

One problem with the ANOVA just reported is that the logical structure of the relationship between Cues B and C is confounded with conditional dependence. This is because when conditional dependence is operationalized using *G*, on average this metric is higher for the training sets based on XOR than it is for the training sets based on AND. For instance, the average value of *G* for the 10 AND training sets in Table 5-10 is 3.715, whereas the average value of *G* for the 10 XOR training sets in Table 5-10 is 39.95. We can carry out two alternative analyses in light of this problem.

First, we can simply predict the R^2 fit between network responses and actual training set probabilities from *G* for each training set presented to each of the 400 perceptrons. This predicts network performance from degree of conditional dependence and ignores the logical relationship between Cues B and C. When we perform this analysis, we find that the degree of conditional dependence accounts for more than 77% of the variance in the fit of network responses ($R^2 = .772$), $F(1, 398) = 1351.05$, $p < 5.2061\text{E-}130$.

The logical structure of the interaction is also a significant predictor of network performance. A regression equation that predicts the R^2 fit between network responses and actual training set probabilities from the type of interaction (AND vs. XOR) is also statistically significant and accounts for more than 26% of the variance in the fit of network responses ($R^2 = .264$), $F(1, 398) = 142.83$, $p < 2.43051\text{E-}28$. However, this predictive relationship is weaker than the relationship between *G* and performance.

If one predicts performance from G, one captures nearly three times the variance of performance that is captured when one predicts performance from interaction type. This indicates that conditional dependence is a more powerful predictor of network performance than is the logical structure of the interaction.

Important to note, the level of reward (high vs. low) is a better predictor of network performance than is the type of interaction between Cues B and C. A regression equation that predicts the R^2 fit between network responses and actual training set probabilities from reward level accounts for more than 56% of the variance in the fit of network responses ($R^2 = .567$), $F(1, 398) = 521.85$, $p <$ 2.09513E-74. As level of reward manipulates response probabilities while holding the logical structure of the interaction between cues constant, and as this variable accounts for more than twice the amount of variance in performance as is captured by interaction type, this provides additional evidence that the logical structure of the training set does not provide the best account of network performance.

These three variables do appear to capture different components of performance variance. We used multiple regression to predict network performance from three variables: G, the logical relationship between Cues B and C, and level of reward. This equation accounted for almost all of the variance in network performance ($R^2 = .981$), $F(3, 396) = 6640.83$, $p < 0$, and all three predictors are statistically significant. In this regression equation, the bulk of the variance is being captured by G, but the other two predictors capture a statistically significant amount of variance that G does not predict.

5.6.3 Comparison to Conditionally Independent Cues

The statistical analyses just reported proceeded by taking four of the simulations to represent cells in a 2×2 factorial design. Another approach to examining the simulation results is to compare the results of the four simulation studies that involve an interaction between Cues B and C to a fifth simulation, the initial study in which all three cues were conditionally independent (Section 5.1). The means of the five conditions with their standard deviations are presented in Table 5-19. Each of these means summarizes the performance of 100 perceptrons; this performance is measured using R^2 between the responses of a perceptron to each of the eight stimuli and the actual probability of reward associated with each.

Table 5-19 reveals that each mean is associated with an extremely small standard deviation, indicating that very similar responses were generated by different perceptrons trained in the same condition. Table 5-20 provides the results of using Welch two-sample t tests to compare network performance in each of these four conditions to network performance in the independent cue conditions. Each of these tests reveals a significant difference.

Table 5-19. The mean probability estimation performance (with standard deviations) of perceptrons in each of the five simulations described in Chapter 5.

	Independent Cues	High-Reward	High-Reward	Low-Reward	Low-Reward
Mean	0.93	0.82	0.52	0.95	0.90
SD	0.04	0.03	0.05	0.01	0.03

Note. Probability estimation performance is operationalized as the squared correlation between the response generated by a perceptron to each of eight possible stimuli and the actual probability of reward associated with each. Each mean is based on the performance of 100 perceptrons.

Table 5-20. The values of t obtained from independent t

High-Reward AND	High-Reward XOR	Low-Reward AND	Low-Reward XOR
20.271	62.779	-6.6047	4.1703
DF = 171.23	DF = 194.55	DF = 121.78	DF = 167.79
p = 2.2e-16	p = 2.2e-16	p = 1.102e-09	p = 4.8

Note. Each t was evaluated at 198 degrees of freedom.

The statistical comparisons in Table 5-20 reveal one surprise: Perceptrons were better probability matchers in the Low-Reward AND condition than in the conditionally independent condition. The reason is that the stochastic method used to create training sets is not perfect: It introduces a certain amount of noise when rewards are assigned to stimuli. As a result, there is a small degree of conditional dependence in the conditionally independent training set. When conditional dependence is operationalized as G, the average value of this measure for the conditionally independent training set is 9.79. This value is higher than the mean G for the Low-Reward AND condition, which was equal to 1.27, which explains the results in Table 5-20.

This reinforces a key thread running through the simulation results presented in the current chapter: The best predictor of perceptron performance is the conditional dependence between cues in the environment and not the logical relationship between the cues. More confirmation for this conclusions comes from computing the correlation between the R^2 measure of each perceptron's performance (i.e., each of the 500 perceptrons trained in the five simulations reported in this chapter) to G for the training set to which the perceptron was exposed. This correlation is −0.881, indicating that perceptron probability estimation is strongly (and inversely) related to conditional dependence. The square of this correlation is 0.776, indicating that more than 77% of the variance in perceptron probability estimation can be explained by a single variable, one that measures conditional dependence.

5.7 Summary and Implications

5.7.1 Summary

Chapter 3 provided a formal analysis of what a perceptron learns when it is confronted by a single cue that signals a reward with uncertainty. It proved that the perceptron would learn to generate a conditional probability (i.e., the likelihood of reward given the presence of a cue) that is defined by Bayes's theorem. Chapter 4 extended this formal analysis to the situation in which two uncertain cues are encountered in the environment and proved that in this situation a perceptron will learn to generate conditional probabilities consistent with the naïve Bayes rule. It also argued that this proof could be extended to situations in which the probability of reward was signaled by more than two cues. Chapter 5 presented a number of simulations directly related to this last point.

Chapter 5 has provided the results of five simulations in which perceptrons learn the probabilities of reward signaled by three cues. One purpose of these simulations was to confirm empirically the formal prediction made in Chapter 4: that these perceptrons would act as naïve Bayesian mechanisms. A second purpose of these simulations was to collect data that can be compared to data collected from human participants when they explore a particular three-cue environment, the one provided by the card-choice task that was introduced in Chapter 1 (Section 1.2.3). A third purpose of these simulations was to explore the limitations of perceptrons in the context of their being limited to the naïve Bayes rule. In particular, these simulations explored performance when a traditional approach to perceptron limitations (linear nonseparability) is contrasted to a different approach that emerges from probability theory (conditional dependence).

The results of the various simulations reported in Chapter 5 lead to two general conclusions. First, all of the simulations are consistent with the expectation that perceptrons confronted with three uncertain signals of reward can be described as naïve Bayesian mechanisms. That is, all of the perceptrons learn the conditional probabilities associated with the various cues under the restriction that the perceptrons are blind to any interactions between cues. Second, all of the simulations point to the conclusion that, when perceptrons learn about uncertain environments signaled by multiple cues, the limitations of perceptrons are best explained by appealing to the amount of conditional dependence between different environmental cues. The traditional use of linear nonseparability to describe perceptron limitations is far less useful in the context of uncertain environments.

5.7.2 Implications

At this point in the monograph, we have established a detailed formal and empirical understanding of how perceptrons adapt to uncertain environments. In particular, when trained in an environment whose uncertainty is consistent with a probabilistic discrimination task, perceptrons learn to match probabilities. That is, the response of a perceptron to a stimulus is its estimate of the probability that the stimulus signals a reward. However, these estimates are constrained by the fact that perceptrons are naïve Bayesians. This means as more conditional dependence exists among the cues that signal rewards, perceptron estimates become less accurate.

That perceptrons estimate that probabilities in this fashion is an interesting and useful phenomenon. However, this ability by itself is not sufficient to model human probability matching in probability discrimination paradigms like the three card-choice task. In Chapter 1, when human probability matching was introduced, it was described as a relationship between probability of reward and frequency of choice. That is, the frequency with which human participants choose particular entities is the variable that is related to the reward probabilities associated with these entities. That perceptrons are able to estimate such reward probabilities is an important element of this task. However, the perceptrons that we have been discussing are able *only* to make such estimates. If perceptrons are to be used to model something like human performance in the three card-choice task, then they must include an additional mechanism that permits them to convert their probability estimates into choices.

The purpose of the next chapter is to describe a straightforward addition to the gradient descent learning rule that permits perceptrons to choose (and then to learn about) different stimuli. We have seen that the gradient descent rule is an example of incorporating negative feedback into artificial neural networks. In contrast, Chapter 6 makes the case that to make perceptrons choose an additional concept from cybernetics must be included: positive feedback.

Chapter 6: Choice and Positive Feedback

Up to this point, our formal understanding of perceptron learning, and our demonstrations of this learning via simulations, has focused on negative feedback: using the difference between expectations and reality to adjust associations and improve predictions. Chapter 6 explores the implications of a different source of information—positive feedback. Positive feedback is a tendency to amplify existing expectations. It can be modeled by guiding choice behavior: having a system decide to choose a stimulus because of an expectation about its reward probability. In other words, the more likely we expect to receive a reward from a stimulus, the more likely we are to choose it. This type of positive feedback is introduced to perceptron learning by implementing operant learning. In operant learning, when a perceptron detects a stimulus, it generates a response—its estimate of the probability of reward. It then uses this estimate to make a decision about whether to choose the stimulus and learn about it. This means that learning does not occur with each stimulus presentation. Two very different operant learning rules are explored. One is an increasing returns rule, where networks are more likely to learn about stimuli with high-reward probabilities than about stimuli with low-reward probabilities. In this rule, the likelihood of learning about a stimulus is equal to the estimated probability of reward. The other rule models the gambler's fallacy, where participants feel that the luck has run out for a high-reward stimulus and therefore are more likely to learn about a low-reward stimulus. In this rule, the likelihood of learning about a stimulus is equal to 1 minus the estimated probability of reward. This chapter reports the results of a number of simulations that train operant networks in the five stimulus conditions that were detailed in Chapter 5. We discover that the two operant learning rules lead to very different patterns of choice behavior. However, we also discover that either of these patterns produces perceptrons that have essentially the same models of the probabilities of reward associated with each cue.

6.1 Choice and Positive Feedback

6.1.1 From Probability to Choice

Before this section, we have considered two interrelated topics. The first is an uncertain world in which environmental cues provide signals about the probability of reward. The second is a simple system, an artificial neural network called a perceptron, which learns what these cues signal. Previous chapters have described extensive formal and empirical explorations of what perceptrons learn when exposed to an uncertain environment. A variety of results support one general conclusion: When perceptrons learn about an uncertain environment, they behave as naïve Bayesian mechanisms. When presented a stimulus—a configuration of cues—a perceptron outputs an estimate of the probability of reward that the cues signal. It then uses the outcome associated with the stimulus—whether it actually is rewarded or not—to modify its structure to improve future estimates. However, these estimates are constrained by the assumption that the cues are independent signals of reward probability.

We have spent a great deal of effort acquiring this technical understanding of perceptrons because our intention to use these networks to shed light on probability learning in biological agents. Chapter 1 introduced the study of probability learning by experimental psychologists, highlighting one phenomenon of particular interest: probability matching. Probability matching occurs when the probability that participants make a particular choice closely corresponds to the probability that the chosen stimulus is rewarded (Estes, 1957a, 1964). In other words, psychologists view probability matching as a relationship between frequency of choice and likelihood of reward; psychological studies of probability learning do not focus on subjects' probability estimates per se.

This definition of probability matching raises one problem: The perceptrons that have been discussed to this point do not make choices. Instead, they merely estimate reward probabilities. The purpose of this chapter is to remove this obstacle. It describes operant learning in perceptrons; with such learning, perceptrons only adapt to patterns that they *choose* to learn about (Dawson et al., 2009). In the operant paradigms introduced next, a perceptron detects a pattern and estimates the likelihood that the pattern signals reward. It then uses this likelihood to decide whether to choose the stimulus. If it chooses the stimulus, then it learns about it. If it instead ignores the stimulus, then it does not learn about it. This parallels the choice-driven learning in the card-choice task that was introduced in Chapter 1: Participants learn about a card in this task only by choosing it to see if it will provide a reward.

Let us now introduce two different procedures for converting probability estimates into choices during operant learning before outlining the rationale for the simulations whose results make up the bulk of Chapter 6.

6.1.2 Increasing Returns Perceptrons

Instrumental or operant conditioning is a foundational concept in the psychology of learning (Skinner, 1938). In operant conditioning, an agent first emits a behavior or a response. Then, this response leads to particular consequences. The behaving agent learns from these consequences, and this learning affects the agent's choices of its future actions. For instance, consider Thorndike's (1901) famous studies of the behavior of animals escaping puzzle boxes. When placed in these homemade contraptions, the animals were intent on escaping. They could perform a variety of actions: squeezing, clawing, biting, and so on. Some of these actions (clawing a string or loop or button) result in a successful and rewarding escape. Thorndike found that with repeated experience in the puzzle box, animals were much more likely to perform behaviors that had led to escaping and were much less likely to perform behaviors that did not result in an escape. This led to a basic operant principle, Thorndike's law of effect, which expressed the notion that some consequences of action increase the likelihood of choosing a particular behavior, whereas other consequences decrease this likelihood.

The law of effect in an example of a phenomenon called *increasing returns*. Increasing returns is a principle that has had a controversial history in economic theory (Arthur, 1989, 2015). It is used to account for why one technology can successfully corner a market after starting out competing against other similar technologies. According to increasing returns, this technology might begin with some small advantage that causes it to be slightly more preferred by consumers than are its competitors. Because of this small advantage, there are more purchases of this technology than of the others. This in turn leads to more investment in the technology (e.g., increasing sales of a technology result in a company putting resources into adding improvements to it), which in turn makes it more attractive than are its competitors. This is because there is less likelihood of the competitors investing in products that are not selling as well. The increasing attractiveness of the product increases its sales further, and the cycle continues until the point that one product might corner the market while the original competitors to it are taken out of production.

Economic theory that includes increasing returns is controversial because this principle conflicts with the more established notion that an economy will settle into a stable equilibrium and that when the economy moves away from this equilibrium, its underlying dynamics will return it to a stable state (Arthur, 2015). This more traditional view is essentially the claim that the economy is governed by negative feedback. Increasing returns is in conflict with this view because it claims that important regularities in the economy involve positive feedback that moves the economy away from equilibrium.

Recall that Chapter 2 introduced the cybernetic notion of negative feedback (Ashby, 1956). When a system is governed by negative feedback, it compares its current state to a goal state by computing the difference between the two. It then performs an adjustment to reduce this difference, taking it closer to the goal state. In other words, a system that uses negative feedback does so to maintain a desired or stable state. Beginning in Section 2.5.2, it was argued that the error correcting rules used to train perceptrons are examples of employing negative feedback to achieve equilibrium (i.e., to achieve a state of minimal error of responses). Increasing returns is instead an example of *positive feedback*, which is the opposite of negative feedback, and which was of less interest to cyberneticists. Positive feedback is a signal that leads to changes that *increase* the difference between a system's current state and an equilibrium state. These changes amplify a system's move away from stability.

The principle of increasing returns can be added to the procedure used to train a perceptron to convert it into a system that learns via operant conditioning. For instance, consider the following learning paradigm (Dawson et al., 2009): Training begins by presenting a stimulus to a perceptron. This in turn causes the perceptron to respond; this response is the perceptron's estimate of the probability that this stimulus will lead to reward. In nonoperant learning (used in the Chapter 5 simulations), the next step would be to compute response error and update connection weights. The increasing returns paradigm does not do this. Instead, it first uses the probability estimate to guide a decision about whether to proceed and learn about the stimulus. This is accomplished by making this choice with the same probability as the estimate. This means a perceptron is very likely to learn when a stimulus causes a high response but is much less likely to learn when a stimulus causes a low response. If the perceptron chooses to learn about the stimulus, then learning proceeds in the standard way. However, if the perceptron does not choose to learn about the stimulus, then it is simply ignored—no learning occurs, and the perceptron moves on to process the next stimulus.

This operant procedure is an example of using increasing returns because as the perceptron alters its connection weights, the responses made to the various stimuli will change. One result of these changes is that some stimuli will lead to higher probability estimates than will others. As a result, these stimuli will become more and more likely to be chosen for learning, whereas the opposite will be true for stimuli that lead to weaker responses. However, increasing returns is not the only principle being employed. Whenever a stimulus is selected for learning, this learning is accomplished by an error-correcting rule, which is an example of negative feedback. In other words, the operant training of a perceptron using an increasing returns paradigm leads to an interesting situation in which perceptron learning is guided by two principles—positive feedback and negative feedback.

The increasing returns paradigm will necessarily change how a perceptron experiences a training set. Consider the simulations that were described in Chapter 5. Each of these simulations used a training set that consisted of 320 stimuli; it was constructed by taking eight stimuli (all possible configurations of three cues) and duplicating each of these stimuli 40 times. Every perceptron was trained on this training set for 2,500 epochs. In other words, each perceptron was trained on 100,000 stimuli. This will not be the case in the increasing returns paradigm. By definition, the use of this paradigm means that fewer stimuli will be involved in learning. This is because while in a training session, a perceptron might still be presented 100,000 stimuli, the operant procedure will ensure that the perceptron will alter its weights only to a subset of these patterns—those that it chooses to learn about.

Furthermore, the increasing returns procedure will implement a bias in the patterns that the perceptron chooses to learn about so that there will be unequal learning about each different type of stimulus. The perceptron will learn more about stimuli that lead to higher responses (i.e., higher predictions of reward probability) and will learn less about stimuli that lead to lower responses.

This raises the question of whether the increasing returns paradigm will produce different perceptron responses at the end of training in comparison to the responses of the nonoperant perceptrons that were detailed in Chapter 5. The main goal of Chapter 6 is to compare the performance of operant perceptrons to that of nonoperant perceptrons by using operant procedures to train networks on the same training sets that were introduced in Chapter 5.

6.1.3 Gambler’s Fallacy Perceptrons

One motivation for the increasing returns paradigm described in the previous section is that it provides a plausible approach to modeling the card-choice task that was introduced in Chapter 1. We can imagine a participant scanning these cards, which are presented in random order. Looking at one of the eight cards is analogous to presenting it to a perceptron. If the participant chooses the card and then sees whether it provides a reward, this is analogous to the perceptron deciding to learn about a stimulus. The increasing returns procedure assumes that if participants in the card-choice want to accumulate a large number of rewards, they will do so by probability matching. That is, they will be more likely to choose cards that they have found to be more likely to offer a reward, and they will be less likely to choose cards that have rewarded them less frequently.

However, human participants may use radically different strategies to select cards in the card-choice task. Chapter 1 briefly discussed some of the early results on probability discrimination learning, and noted (Section 1.3.1) that it revealed individual differences in behavior. In particular, researchers discovered that some participants exhibited a “gambler’s fallacy”; they were more likely to predict an event the longer that it had *not* been rewarded (Jarvik, 1951). This strategy is the assumption that an often-rewarded event might be due to stop rewarding and that a rarely rewarded event might be due to deliver a reward. This is called the *gambler’s fallacy* because it is based on a mistaken assumption of dependence between events. For instance, to think that an event will start to provide rewards because it has been on a long cold streak is to assume that the current likelihood of a reward depends on this history.

Realizing that perceptron responses are subjective probabilities of reward, it is quite straightforward to design alternative operant paradigms for their training. For example, let $P(x)$ be the response of a perceptron to some stimulus x. In the increasing returns paradigm, the probability of choosing to learn about stimulus x would also be $P(x)$. The opposite approach—which we call the gambler’s fallacy paradigm—is to make the probability of choosing to learn about stimulus x equal $1 - P(x)$. This paradigm uses a rule that is opposite to that used in increasing returns. According to this gambler’s fallacy rule, a perceptron is more likely to learn about stimuli that cause low responses than to learn about stimuli that cause high responses.

The radical difference between the increasing returns paradigm and the gambler’s fallacy paradigm introduces an important idea: The rule for choosing to learn about a stimulus is quite distinct from the mechanism used to estimate reward probability. Both of these procedures use the same method for estimating probability (perceptron responses to stimuli) but then use a very different approach for converting probability estimates into choices to learn. Of particular interest in the current chapter is whether such radically different choice strategies will lead to radically different probability estimates at the end of training.

6.2 Operant Learning: Three Independent Cues

To investigate the effects of operant learning, let us begin by using it on a task in which three different and independent cues are used to signal reward. For these new simulations, we use the identical training sets that were introduced earlier in Section 5.1. The effect of operant learning can be assessed by comparing the performance of these new operant perceptrons to the performance that was reported in Chapter 5.

6.2.1 Increasing Returns: Training

The first study examines learning with the increasing returns operant learning paradigm. All of its perceptrons are identical in structure to those described earlier in Chapter 5, having a single output unit and three input units. The input units use the same binary code to represent the presence or absence of three cues: Cue A, Cue B, and Cue C. The same five training sets that were described in Table 5-1 are used in this simulation, with 20 perceptrons being trained as individual "subjects" on each of these training sets for a total of 100 networks. Each of these training sets uses the three cues as independent signals of reward. The presence of Cue A indicates a reward probability of 0.14, the presence of Cue B indicates a reward probability of 0.20, and the presence of Cue C indicates a reward probability of 0.40. The methods used to construct five training sets with these (ideal) probabilities, and the nature of the resulting training sets, were described in detail in Section 5.2.1.

Each perceptron is trained with a gradient descent rule, using the same parameters that were described earlier in Chapter 5: with a learning rate of 0.05, with connection weights randomly set in the range from −0.1 to 0.1 prior to training, and with the bias θ of the logistic activation function initialized to 0. The only difference between these networks and those described in Chapter 5 is that an increasing returns operant procedure is employed.

To be more specific, during one epoch of training, we present a network each of the 320 patterns in the training set in random order. For each pattern, the perceptron generates a response that serves as the perceptron's prediction of the probability that this stimulus signals a reward. The perceptron then decides whether to learn about this stimulus or to ignore it; the probability that it decides to learn about the stimulus is equal to the predicted probability of reward (i.e., the output unit's response). This is accomplished by generating a random number. If the random number is greater than the unit's response then the pattern is ignored; otherwise, the perceptron updates its weights and bias for this pattern using the gradient descent rule. Thus, even though the perceptron is presented each of the 320 patterns during an epoch of training, it does not modify its weights for each of these stimuli. Instead, it modifies only weights for those patterns that it chooses to investigate.

Apart from including the increasing returns paradigm, network training proceeds in the identical manner described in Chapter 5. We randomize the order of input pattern presentations every epoch. Training proceeds for 2,500 epochs; we then record network responses to each of the eight possible input patterns, as well as the structure of the perceptron. As well, during training we record the number of times that each pattern is chosen for learning and save this information when training is complete.

Table 6-1. The average number of times that a perceptron selected each of the input patterns for learning using the increasing returns operant training paradigm.

	Set 1		**Set 2**		**Set 3**		**Set 4**		**Set 5**	
Probability	P(R)	Times	P(R)	Times	P(R)	Times	P(R)	Times	P(R)	Times
P(R|~A~B~C)	0.00	8758.3	0.00	14391.45	0.00	14024.95	0.00	11890.65	0.00	10421.65
P(R|~A~BC)	0.4	35932.8	0.35	34359.25	0.375	36117.55	0.4	41833.6	0.45	43976.75
P(R|~AB~C)	0.15	15475.75	0.175	20399.3	0.225	24505.45	0.275	19669.35	0.15	15206.6
P(R|~ABC)	0.5	51905.55	0.475	44361.85	0.55	53067.5	0.575	56626.25	0.55	54821.7
P(R|A~B~C)	0.15	14318.4	0.15	18639.5	0.1	17112.05	0.075	16075.9	0.075	11625.85
P(R|A~BC)	0.475	49671.5	0.45	41536.85	0.45	41718.1	0.55	50613.45	0.475	47035.45
P(R|AB~C)	0.25	24271.15	0.35	25830.55	0.375	29120	0.275	25874.75	0.225	16872.55
P(R|ABC)	0.675	65517.45	0.475	51991.7	0.55	58900.65	0.625	65070.6	0.575	57877.15
Total Choices		**265850.9**		**251510.5**		**274566.3**		**287654.6**		**257837.7**
Correlation		**0.988**		**0.930**		**0.952**		**0.965**		**0.982**

Note. Each average is the mean of 20 perceptrons. The table also provides the probability of reward for each of the patterns in each of the five training sets. The second-to-last row provides the average number of total choices made (from 100,000 presentations) for each training set. The bottom row provides the correlation between the probability of reward and the average number of times each pattern was chosen for each of the training sets.

6.2.2 Increasing Returns: Performance

We saw in Chapter 5 that when nonoperant networks learn about independent cue training sets, by the end of training perceptron, responses are accurate estimates of the reward probability associated with each stimulus. With operant training, there are now two measures relevant for demonstrating that perceptrons match probabilities. One is the response to each stimulus at the end of training. The other is the number of times that each stimulus is chosen for learning. With the increasing returns paradigm, there should be a very strong relationship between stimulus rewards and stimulus choices.

Table 6-1 confirms the existence of this strong relationship for each of the five training sets. Averaging across the 20 perceptrons trained on each of these sets, it reports the average number of times that each stimulus was selected for learning by the operant procedure. An inspection of Table 6-1 reveals that when a stimulus is associated with a higher probability of reward, it is more frequently selected for learning by this operant procedure. Table 6-1 also reports the actual probability of reward associated with each stimulus in a training set and provides the correlation between this probability and the average number of times that a stimulus is selected for learning. These correlations range from about 0.93 to 0.99, indicating that the operant perceptrons exhibit a very high degree of probability matching with respect to their choices. This is expected with this operant learning paradigm.

Table 6-1 also provides the average of the total number of patterns selected for learning by the perceptrons trained on each training set. In Chapter 5, perceptrons were trained for 2,500 epochs, and the stochastic nature of the training set was achieved by replicating each stimulus pattern 40 times. As a result, the Chapter 5 perceptrons learned about each of the eight stimulus patterns 100,000 times. In contrast, as can be seen in the Total Choices row of Table 6-1, these operant perceptrons receive much less training. Total training ranges from just over 25,000 total patterns to just under 29,000 total patterns. In other words, these operant perceptrons received only between 25% and 29% of the total amount of training that was received by the nonoperant perceptrons investigated in Chapter 5.

However, this reduction in training does not seem to have a negative impact on an operant perceptron's ability to estimate the reward probability associated with each stimulus. Table 6-2 provides the average response of a perceptron to each of the eight stimuli for each of the training sets. This average is the mean of the responses generated by the 20 perceptrons trained on each training set. As was the case in Chapter 5, Table 6-2 reveals that perceptron responses are highly accurate estimates of reward probabilities.

To assess the accuracy of perceptron responses, we computed the same measure (R^2) to compare actual probabilities to perceptron responses that was used previously in Chapter 5. This measure is reported for each training set in the final row of Table 6-2. It supports the conclusion that the perceptrons were once again excellent probability matchers. The worst performance, for Training Set 2, produces an R^2 of 0.860, indicating that perceptron responses capture almost 86% of the variance in the expected probabilities. The best performance, for Training Set 1, produces an R^2 of 0.987, indicating that perceptron responses capture

Table 6-2. The probability of reward (P(R)) and the mean responses of perceptrons after being trained on the different conditionally independent three-cue training sets using the increasing returns operant paradigm.

	Training Set 1		**Training Set 2**		**Training Set 3**		**Training Set 4**		**Training Set 5**	
Input	P(R)	Mean	P(R)	Mean	P(R)	Mean	P(R)	Mean Response	P(R)	Mean
0,0,0	0	0.084	0	0.141	0	0.134	0	0.117	0	0.099
0,0,1	0.4	0.362	0.35	0.341	0.375	0.353	0.4	0.428	0.45	0.441
0,1,0	0.15	0.149	0.175	0.199	0.225	0.240	0.275	0.197	0.15	0.149
0,1,1	0.5	0.521	0.475	0.440	0.55	0.526	0.575	0.580	0.55	0.555
1,0,0	0.15	0.141	0.15	0.180	0.1	0.165	0.075	0.164	0.075	0.110
1,0,1	0.475	0.505	0.45	0.409	0.45	0.409	0.55	0.526	0.475	0.469
1,1,0	0.25	0.239	0.35	0.250	0.375	0.286	0.275	0.267	0.225	0.164
1,1,1	0.675	0.661	0.475	0.512	0.55	0.585	0.625	0.672	0.575	0.583
	R^2	**0.975**	**R^2**	**0.860**	**R^2**	**0.906**	**R^2**	**0.931**	**R^2**	**0.963**

Note. Each mean summarizes the performance of 20 perceptrons. The final row indicates the squared correlation (R^2) between the eight perceptron responses and the eight probabilities of reward for each of the training sets.

Table 6-3. The comparison between components of a logistic regression (Logit) and the average structure of a perceptron (Network) for each of the five conditionally independent three-cue training sets.

	Training Set 1		**Training Set 2**		**Training Set 3**		**Training Set 4**		**Training Set 5**	
Source	Logit	Network	Logit	Network	Logit	Network	Logit	Network	Logit	Network
θ	-2.680	-2.397	-2.296	-1.812	-2.381	-1.864	-2.466	-2.026	-2.637	-2.208
w	0.710	0.588	0.569	0.292	0.437	0.239	0.381	0.396	0.295	0.113
w	0.780	0.649	0.701	0.417	1.024	0.710	0.989	0.616	0.730	0.461
w	1.988	1.831	1.397	1.153	1.569	1.258	1.942	1.734	2.177	1.969
	R	**0.999**	**R**	**0.994**	**R**	**0.998**	**R**	**0.992**	**R**	**0.994**

Note. All of these perceptrons were trained using the increasing returns operant paradigm. The final row indicates the squared correlation (R^2) between perceptron structure and logistic regression coefficients.

almost 99% of the variance in the expected probabilities. All of the values reported in Table 6-2 are extremely close in value to those reported in Table 5-2 for the nonoperant perceptrons. To quantify this observation, we performed a Welch two-sample *t* test to compare the 100 R^2 values used to create the Table 5-2 data for nonoperant perceptrons (*M* value = 0.925) to the compare the 100 R^2 values used to create the Table 6-2 data for operant perceptrons (*M* value = 0.924). This comparison revealed no statistically significant difference, $t(197.87) = 0.14636$, $p = .8838$.

6.2.3 Increasing Returns: Perceptron Structure

Previous work has shown that there exists formal equivalence between a perceptron that uses the logistic activation function and logistic regression (Schumacher et al., 1996). In particular, the weights and bias of a multiple cue perceptron represent the same type of odds ratios as the coefficients in logistic regression (e.g., Equations 4-14, 4-15, and 4-16). In Chapter 5, we took advantage of this relationship to confirm that the structure of trained perceptrons revealed the expected odds ratios. We did so by comparing the components of perceptron structure to the coefficients of logistic regressions performed on each of the five training sets (Section 5.2.4). We employ this same approach to examine the final structure of the operant perceptrons.

Table 6-3 presents the logistic regression coefficients for each training set that were presented earlier in Table 5-3. It also provides the average bias and weights of the 20 perceptrons that learned about each of the training sets using the increasing returns operant paradigm. Finally, Table 6-3 offers the squared correlation between the regression coefficients and the perceptron components for each training set. Each R^2 is over 0.99, demonstrating that the operant training led to the same perceptron structure predicted by our Chapter 4 formal analysis, and exhibited by the nonoperant perceptrons trained on the same stimuli in Chapter 5. In short, the reduced amount of training that emerges from the increasing returns learning rule leads to nearly identical perceptron structure by the end of training.

Of interest, a comparison of the structure of nonoperant and operant perceptrons revealed a statistically significant difference. We performed a Welch two-sample *t* test to compare the 100 R^2 values used to create the Table 5-3 data for nonoperant perceptrons (*M* value = 0.999) to the compare the 100 R^2 values used to create the Table 6-3 data for operant perceptrons (*M* value = 0.994). This comparison revealed a statistically significant difference, $t(103.06) = 11.091$, $p = 2.2e\text{-}16$, despite that the two means differ by only 0.005. This is because this slight decrease in fit was true for almost all the operant perceptrons, and the variance of both sets of fit values is very small.

6.2.4 Gambler's Fallacy: Training

The preceding material in Section 6.2 described simulations in which perceptrons used an increasing returns paradigm to learn about the probability of reward signaled by three independent cues. We now turn to a set of simulations that use a different operant paradigm, the one based on the gambler's fallacy. According to this rule, the likelihood of choosing to learn about a stimulus is equal to 1 minus the predicted probability of reward. As a result, the perceptron is more likely to choose to learn about patterns that are associated with low probabilities of reward. Apart from this change in the operant paradigm, the simulations described next were performed using exactly the same methodology as that described in Section 6.2.1.

6.2.5 Gambler's Fallacy: Performance

We saw earlier that operant paradigms provide two measures related to probability matching: the number of times that a stimulus is chosen for learning, and the perceptron's response to that stimulus at the end of training.

Table 6-4. The average number of times that a perceptron selected each of the input patterns for learning using the gambler's fallacy operant training paradigm.

	Set 1		Set 2		Set 3		Set 4		Set 5	
Probability	P(R)	Times	P(R)	Times	P(R)	Times	P(R)	Times	P(R)	Times
P(R|~A~B~C)	0.00	92373.95	0.00	89793.9	0.00	90714.1	0.00	91337.1	0.00	92096.05
P(R|~A~BC)	0.4	64449.7	0.35	69647.7	0.375	67663.6	0.4	60792.2	0.45	59126.1
P(R|~AB~C)	0.15	85815.95	0.175	81829.95	0.225	78046.1	0.275	79731.45	0.15	85827.6
P(R|~ABC)	0.5	47410.2	0.475	53923	0.55	43025.55	0.575	36493.7	0.55	42752.2
P(R|A~B~C)	0.15	86638.2	0.15	83572	0.1	86118.35	0.075	87893.05	0.075	90120.55
P(R|A~BC)	0.475	49151.75	0.45	56908.4	0.45	56877.95	0.55	51580.2	0.475	52928.4
P(R|AB~C)	0.25	76395.2	0.35	72201.3	0.375	69164.75	0.275	73021.45	0.225	82493.75
P(R|ABC)	0.675	32563.5	0.475	40206.9	0.55	32321.9	0.625	28319.1	0.575	36803.2
Total Choices		**534798.5**		**548083.2**		**523932.3**		**509168.3**		**542147.9**
Correlation		**-0.986**		**-0.923**		**-0.945**		**-0.962**		**0.982**

Note. Each average is the mean of 20 perceptrons. The table also provides the probability of reward for each of the patterns in each of the five training sets. The second-to-last row provides the average number of total choices made (from 100,000 presentations) for each training set. The bottom row provides the correlation between the probability of reward and the average number of times each pattern was chosen for each of the training sets.

Let us consider these two measures for perceptrons trained with the gambler's fallacy paradigm.

Table 6-4 reports the mean number of times that each stimulus was selected for learning using the operant procedure, averaging over the 20 perceptrons trained on each training set. An inspection of Table 6-4 reveals, as expected, that the gambler's fallacy paradigm produces a strong negative correlation between a stimulus' probability of reward and the number of times that it is selected for learning. For each of the training sets, the correlation between these two measures is extremely high but is in the opposite direction of the relationship observed for the other operant paradigm (Table 6-1). This shows that when a stimulus is associated with a lower probability of reward, it is more frequently selected for learning using this operant procedure. These correlations range from about −0.92 to −0.99, indicating that the operant perceptrons exhibit a very high degree of probability matching with respect to their choices. This is expected with this operant learning paradigm.

Table 6-4 clearly indicates that the perceptrons trained with the gambler's fallacy operant procedure had a very different experience than did those trained according to increasing returns. First, the two paradigms led to opposite preferences of pattern choice, as indicated by the opposite signed correlations when Table 6-4 is compared to Table 6-1. Second, the perceptrons trained with the gambler's fallacy operant paradigm chose to learn about approximately twice the number of patterns than did the perceptrons trained with the other operant procedure. Table 6-4 indicates that, on average, these perceptrons chose to learn about between 51,000 and 55,000 of the total of 100,000 stimuli that they were presented, as indicated in the Total Choices row. This is substantially larger than the range from 26,000 to 29,000 choices provided by Table 6-1. In general, this is expected because the training sets were designed so that the likelihood of not being rewarded (roughly 0.66) is twice the likelihood of being rewarded (roughly 0.33).

Although the different operant procedures led to very different choices about patterns to learn, these choices produced very similar probability estimates by the end of training. Table 6-5 provides the average response of a gambler's fallacy perceptron to each of the eight stimuli for each of the training sets. This average is the mean of the responses generated by the 20 perceptrons trained on each training set. As was the case in Table 6-2, these responses are highly accurate estimates of reward probabilities, as shown by the goodness of fit measure provided in the bottom row of Table 6-5. The worst performance, for Training Set 2, again produces an R^2 of 0.860. The best performance, for Training Set 1 produces an R^2 of 0.987. All the values reported in Table 6-5 are extremely close in value to those reported for the other operant procedure in Table 6-2.

We quantified this comparison by performing a Welch two-sample *t* test to compare the 100 R^2 values used to create the Table 5-2 data for nonoperant perceptrons (M value = 0.925) to the compare the 100 R^2 values used to create the Table 6-5 data for operant perceptrons (M

Table 6-5. The probability of reward (P(R)) and the mean responses of perceptrons after being trained on the different conditionally independent three-cue training sets using the gambler's fallacy operant paradigm.

	Training Set 1		**Training Set 2**		**Training Set 3**		**Training Set 4**		**Training Set 5**	
Input	P(R)	Mean Response	P(R)	Mean Response	P(R)	Mean Response	P(R)	Mean Response	P(R)	Mean Response
0,0,0	0	0.084	0	0.141	0	0.134	0	0.117	0	0.099
0,0,1	0.4	0.362	0.35	0.341	0.375	0.353	0.4	0.428	0.45	0.441
0,1,0	0.15	0.149	0.175	0.199	0.225	0.240	0.275	0.197	0.15	0.149
0,1,1	0.5	0.521	0.475	0.440	0.55	0.526	0.575	0.580	0.55	0.555
1,0,0	0.15	0.141	0.15	0.180	0.1	0.165	0.075	0.164	0.075	0.110
1,0,1	0.475	0.505	0.45	0.409	0.45	0.409	0.55	0.526	0.475	0.469
1,1,0	0.25	0.239	0.35	0.250	0.375	0.286	0.275	0.267	0.225	0.164
1,1,1	0.675	0.661	0.475	0.512	0.55	0.585	0.625	0.672	0.575	0.583
	R^2	**0.975**	R^2	**0.860**	R^2	**0.906**	R^2	**0.931**	R^2	**0.963**

Note. Each mean summarizes the performance of 20 perceptrons. The final row indicates the squared correlation (R^2) between the eight perceptron responses and the eight probabilities of reward for each of the training sets.

Table 6-6. The comparison between components of a logistic regression (Logit) and the average structure of a perceptron (Network) for each of the five conditionally independent three-cue training sets.

	Training Set 1		**Training Set 2**		**Training Set 3**		**Training Set 4**		**Training Set 5**	
Source	Logit	Network	Logit	Network	Logit	Network	Logit	Network	Logit	Network
θ	-2.680	-2.524	-2.296	-2.196	-2.381	-2.302	-2.466	-2.383	-2.637	-2.499
w	0.710	0.643	0.569	0.571	0.437	0.473	0.381	0.392	0.295	0.261
w	0.780	0.681	0.701	0.700	1.024	1.052	0.989	1.004	0.730	0.682
w	1.988	1.930	1.397	1.381	1.569	1.573	1.942	1.930	2.177	2.111
	R^2	**0.999**	R^2	**1.000**	R^2	**1.000**	R^2	**1.000**	R^2	**1.000**

Note. All these perceptrons were trained with the gambler's fallacy paradigm. The final row indicates the squared correlation (R^2) between perceptron structure and logistic regression coefficients.

value = 0.920). This comparison revealed no statistically significant difference, $t(197.54) = 0.86644$, $p = .3873$.

6.2.6 Gambler's Fallacy: Perceptron Structure

Given the high degree of similarity between the perceptron responses provided in Tables 6-2 and 6-5, it would be expected that both types of operant training led to perceptrons with very similar structures. To examine this prediction we compared the structure of each perceptron (its weights and bias) to the logistic regression coefficients for each training sets. The results of this comparison are provided in Table 6-6. It provides the average bias and weights of the 20 perceptrons that learned about each training set using the gambler's fallacy paradigm. The squared correlations at the bottom of Table 6-6 are each over 0.99, demonstrating that the operant training led to the same perceptron structure predicted by our Chapter 4 formal analysis, exhibited by the nonoperant perceptrons trained on the same stimuli in Chapter 5, and exhibited by the other operant perceptrons described earlier in Section 6.2. In short, although the gambler's fallacy operant paradigm led to a very different kind of training (in terms of pattern choices) than did the increasing returns paradigm, the two operant paradigms zroduced perceptrons of nearly identical structure, and which generated nearly identical probability estimates.

However, a comparison of the structure of nonoperant and operant perceptrons reveals a statistically significant difference. We performed a Welch two-sample *t* test to compare the 100 R^2 values used to create the Table 5-3 data for nonoperant perceptrons (*M* value = 0.999) to the compare the 100 R^2 values used to create the Table 6-6 data for operant perceptrons (*M* value = 0.9994). This comparison revealed a statistically significant difference, $t(189.38) = -5.9896$, $p = 1.038\text{e-}08$, despite that the two means differ by 0.0004. In this case, the operant perceptrons revealed a small but reliable (and statistically significant) improvement in fit in comparison to the nonoperant networks.

6.3 Operant Learning: High-Reward AND

Section 6.2 described how two operant paradigms, one based on increasing returns and another based on the gambler's fallacy, performed on a probabilistic discrimination task in which three independent cues signaled a probability of reward. The next four sections of Chapter 6 investigate how learning for the four conditions involving an interaction between cues is accomplished by the two operant paradigms. This is of particular interest because we saw in Chapter 5 that the ability of perceptrons to estimate probabilities decreased when the conditional dependence in the training set increased. As both operant paradigms rely on a perceptron's changing estimates of reward probability, it is important to determine whether either of these paradigms is useful when the signals from cues interact. Section 6.3 begins by considering operant learning when the training set uses the AND of Cues B and C to signal a probability of reward, and the probability of reward is high, leading to an increased amount of conditional dependence.

6.3.1 Architecture, Training Sets, and Training

The methodology used to collect the results detailed next parallels that described in Section 6.2. All the perceptrons consisted of three input units and one logistic output unit. Perceptrons were trained on one of the five High-Reward AND training sets that were originally introduced in Section 5.3.1. These training sets were stochastically constructed so that the presence of Cue A signaled a reward probability of 0.48, and the AND of Cues B and C signaled a reward probability of 0.48. The properties of the five training sets were introduced earlier in Table 5-4. All perceptrons were trained using the same parameters that were used throughout Chapter 5 as well as earlier in the current chapter. Connection weights were randomly initialized in the range from −0.1 to 0.1, the output unit bias was initialized to 0, and the learning rate was 0.05. Each network was trained for 2,500 epochs using the gradient descent rule, with each epoch involving the presentation of 320 stimuli in a random order. One hundred networks (20 for each training set) were trained using the increasing returns operant paradigm. Another 100 networks (20 for each training set) were trained using the gambler's fallacy operant paradigm.

6.3.2 Increasing Returns: Performance

Let us begin by discussing the performance of perceptrons that were trained on the High-Reward AND problem using the increasing returns operant paradigm. Averaging across the 20 perceptrons trained on each of the five training sets, Table 6-7 reports the mean number of times that each stimulus was selected for learning by this operant procedure. An inspection of Table 6-7 reveals that when a stimulus is associated with a higher probability of reward, it is more frequently selected for learning using this operant procedure. Table 6-7 also reports the actual

Table 6-7. The average number of times that a perceptron selected each of the input patterns for learning using the increasing returns operant training paradigm when learning the High

	Set 1		Set 2		Set 3		Set 4		Set 5	
Probability	**P(R)**	**Times**	**P(R)**	**Times**	**P(R)**	**Times**	**P(R)**	**Times**	**P(R)**	**Times**
P(R|~A~B~C)	0	9313.2	0	5832.45	0	6571.1	0	6822.3	0	9719.7
P(R|~A~BC)	0	20303.5	0	14916.8	0	15903.65	0	15368.6	0	19946.6
P(R|~AB~C)	0	16032.55	0	16817.1	0	16873.3	0	16375.95	0	19896.1
P(R|~ABC)	0.475	32229.5	0.45	36926.8	0.45	35677.4	0.425	33088.35	0.525	36585.3
P(R|A~B~C)	0.5	35567.2	0.45	21976.4	0.425	23956.05	0.425	29198.75	0.475	32462.6
P(R|A~BC)	0.55	58020.8	0.4	45025	0.425	46366.05	0.475	51080.6	0.500	52645.6
P(R|AB~C)	0.45	50812.15	0.45	48625.7	0.45	48064.2	0.5	52910.2	0.500	52647.75
P(R|ABC)	0.725	72011.6	0.75	73343.3	0.725	71723.05	0.75	73874.15	0.700	71969.65
Total Choices		**294290.5**		**263463.6**		**265134.8**		**278718.9**		**295873.3**
Correlation		**0.910**		**0.894**		**0.909**		**0.939**		**0.897**

Note. Each average is the mean of 20 perceptrons. The table also provides the probability of reward for each of the patterns in each of the five training sets. The second-to-last row provides the average number of total choices made (from 100,000 presentations) for each training set. The bottom row provides the correlation between the probability of reward and the average number of times each pattern was chosen for each of the training sets.

Table 6-8. The probability of reward (P(R)) and the mean responses of perceptrons after being trained on the different High

	Training Set 1		**Training Set 2**		**Training Set 3**		**Training Set 4**		**Training Set 5**	
Input	P(R)	Mean Response	P(R)	Mean Response	P(R)	Mean Response	P(R)	Mean Response	P(R)	Mean Response
0,0,0	0.000	0.089	0.000	0.053	0.000	0.061	0.000	0.065	0.000	0.095
0,0,1	0.000	0.198	0.000	0.147	0.000	0.158	0.000	0.155	0.000	0.201
0,1,0	0.000	0.156	0.000	0.162	0.000	0.164	0.000	0.164	0.000	0.203
0,1,1	0.475	0.318	0.450	0.374	0.450	0.361	0.425	0.343	0.525	0.377
1,0,0	0.500	0.353	0.450	0.211	0.425	0.234	0.425	0.287	0.475	0.322
1,0,1	0.550	0.578	0.400	0.453	0.425	0.468	0.475	0.515	0.500	0.531
1,1,0	0.450	0.507	0.450	0.481	0.450	0.479	0.500	0.532	0.500	0.535
1,1,1	0.725	0.720	0.750	0.741	0.725	0.725	0.750	0.750	0.700	0.732
	R^2	**0.828**	R^2	**0.797**	R^2	**0.824**	R^2	**0.881**	R^2	**0.804**

Note. Each mean summarizes the performance of 20 perceptrons. The final row indicates the squared correlation (R^2) between the eight perceptron responses and the eight probabilities of reward for each of the training sets.

probability of reward associated with each stimulus in a training set and provides the correlation between this probability and the average number of times that a stimulus is selected for learning. These correlations range from about 0.89 to 0.94, indicating that the operant perceptrons exhibit a very high degree of probability matching with respect to their choice behavior. This is as expected with this operant learning paradigm. However, the correlations in Table 6-7 are slightly lower than those reported in Table 6-1, indicating that for this training set, although the networks were accurate probability matchers, this accuracy was slightly less than was observed when all three cues were independent signals of reward.

Table 6-7 also provides the average of the total number of patterns selected for learning by the 20 perceptrons trained on each training set. As can be seen in the Total Choices row of Table 6-7, these operant perceptrons received less training than did the nonoperant perceptrons in Chapter 5, which were trained on 100,000 stimuli. Table 6-7 shows that the total amount of training ranged from just over 26,000 total patterns to just under 30,000 total patterns. This is a similar range to that reported for the independent cue perceptrons in Table 6-1.

Table 6-8 provides the average response (after training is complete) of perceptrons to each of the eight stimulus types for each training set. Each average is the mean of the responses generated by the 20 perceptrons trained on the same training set.

The results reported in Table 6-8 indicate, on one hand, strong relationships between perceptron responses and the actual probabilities of reward. For instance, the worst performance (Training Set 2) produces an R^2 of 0.797. The best performance, for Training Set 4, produces an R^2 of 0.938. The results in this table seem very similar to those for nonoperant perceptrons trained on the same problems (Table 5-5). We performed a Welch two-sample t test to compare the 100 R^2 values used to create the Table 5-5 data for nonoperant perceptrons (M value = 0.823) to the 100 R^2 values used to create the Table 6-8 data for operant perceptrons (M value = 0.824). This comparison revealed no statistically significant difference, $t(196.04) = -0.30199$, $p = .763$.

Given this result, the structure of these operant perceptrons should be similar to that of nonoperant perceptrons and should be strongly related to logistic regression coefficients. Table 6-9 presents the logistic regression coefficients for each training set, as well as the average bias and weights of the 20 perceptrons that learned about each of the training sets using the increasing returns operant paradigm. The final row of Table 6-9 provides the mean squared correlation between the regression coefficients and the perceptron components for each training set. Each R^2 is over 0.99, demonstrating that the operant training led to the same perceptron structure predicted by our Chapter 4 formal analysis and exhibited by the nonoperant perceptrons trained on the same stimuli in Chapter 5.

A comparison of the fit of the structure of nonoperant and operant perceptrons to regression coefficients revealed a statistically significant difference. We performed a Welch two-sample t test to compare the 100 R^2 values used to create the Table 5-6 data for nonoperant perceptrons (M value = 0.999) to the compare the 100 R^2 values used to create the Table 6-9 data for operant perceptrons (M value = 0.992). This comparison revealed a statistically significant

Table 6-9. The comparison between components of a logistic regression (Logit) and the average structure of a perceptron (Network) for each of the five High

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Source	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**
θ	-3.318	-2.323	-3.525	-2.888	-3.523	-2.735	-3.630	-2.670	-3.264	-2.250
w	2.484	1.715	2.388	1.567	2.357	1.552	2.585	1.755	2.322	1.507
w	0.942	0.634	1.271	1.243	1.233	1.108	1.261	1.043	1.144	0.882
w	1.242	0.922	1.120	1.130	1.157	1.060	1.183	0.973	1.144	0.866
	R^2	**1.000**	R^2	**0.983**	R^2	**0.990**	R^2	**0.996**	R^2	**0.997**

Note. All these perceptrons were trained using the increasing returns operant paradigm. The final row indicates the average squared correlation (R^2) between perceptron structure and logistic regression coefficients.

Table 6-10. The average number of times that a perceptron selected each of the input patterns for learning using the gambler's fallacy operant training paradigm.

	Set 1		Set 2		Set 3		Set 4		Set 5	
Probability	**Times**	**P(R)**	**Times**	**P(R)**	**Times**	**P(R)**	**Times**	**P(R)**	**Times**	**P(R)**
P(R|~A~B~C)	0	95632	0	95385.25	0	95639.95	0	96160.3	0	96020.35
P(R|~A~BC)	0	85966.4	0	88652.3	0	88450.8	0	89239.15	0	87100.15
P(R|~AB~C)	0	89209.9	0	87057.5	0	87616.65	0	88484.9	0	87116.35
P(R|~ABC)	0.475	69405.3	0.45	71486.75	0.45	70810.05	0.425	71347.3	0.525	64997.65
P(R|A~B~C)	0.5	65924	0.45	72065.2	0.425	73222.9	0.425	70652.85	0.475	68754.55
P(R|A~BC)	0.55	34559	0.4	48861.7	0.425	48164.65	0.475	43541.45	0.5	37516.2
P(R|AB~C)	0.45	41562.65	0.45	45172.5	0.45	46282.25	0.5	41784.85	0.5	37500.1
P(R|ABC)	0.725	16474.15	0.75	23519.15	0.725	22863.3	0.75	18883.8	0.7	14278.9
Total Choices		**498733.4**		**532200.4**		**533050.6**		**520094.6**		**493284.3**
Correlation		**-0.901**		**-0.898**		**-0.906**		**-0.932**		**-0.889**

Note. Each average is the mean of 20 perceptrons. The table also provides the probability of reward for each of the patterns in each of the five training sets. The second-to-last row provides the average number of total choices made (from 100,000 presentations) for each training set. The bottom row provides the correlation between the probability of reward and the average number of times each pattern was chosen for each of the training sets.

difference, $t(99.288) = 9.4554$, $p = 1.646e\text{-}15$, despite that the two means differ only by 0.007. It appears that the operant procedure leads reliably to a slightly poorer fit to the odds ratios, even though this fit still accounts for over 99% of the variance of the regression coefficients.

6.3.3 Gambler's Fallacy: Performance

Section 6.3.2 described the performance of perceptrons on the High-Reward AND problem when the increasing returns paradigm was employed. Let us now consider the performance of perceptrons that were trained on the same problem using the gambler's fallacy procedure.

Table 6-10 reports the mean number of times that each stimulus was selected for learning using the gambler's fallacy operant paradigm, averaging over the 20 perceptrons trained on each training set. An inspection of Table 6-10 reveals the expected strong negative correlation between a stimulus' probability of reward and the number of times that it is selected for learning. For each of the training sets, the correlation between these two measures is very high but is in the opposite direction of the relationship observed for the other operant paradigm (Table 6-7). This shows that when a stimulus is associated with a lower probability of reward, it is more frequently selected for learning using this operant procedure. These correlations range from about −0.89 to −0.93.

Table 6-10 indicates that the perceptrons trained with the gambler's fallacy operant paradigm once again chose to learn about approximately twice the number of patterns than did the perceptrons trained with the increasing returns procedure. Table 6-10 indicates that, on average, these perceptrons chose to learn about between 50,000 and 53,000 of the total of 100,000 stimuli that they were presented, as indicated in the Total Choices row.

Although the different operant procedures produced very different choice behavior, these choices led to very similar probability estimates by the end of training. Table 6-11 provides the average response of a perceptron to each of the eight stimulus types for each training set. This average is the mean of the responses generated by the 20 perceptrons trained on the same training set. As was the case in Table 6-8, these responses are highly accurate estimates of reward probabilities, as shown by the measure of goodness of fit provided in the bottom row of Table 6-11. The worst performance, for Training Set 5, produces an R^2 of 0.792. The best performance for Training Set 4 produces an R^2 of 0.871. All the values reported in Table 6-11 are extremely close in value to those reported for the other operant procedure in Table 6-8.

We quantified this comparison by performing a Welch two-sample *t* test to compare the 100 R^2 values used to create the Table 6-8 data for increasing returns perceptrons (*M* value = 0.824) to the compare the 100 R^2 values used to create the Table 6-11 data for gambler's fallacy perceptrons (*M* value = 0.819). This comparison revealed no statistically significant difference, $t(196.33) = 1.2073$, $p = .2288$. A similar test was performed to compare the performance of these perceptrons to the nonoperant perceptrons whose performance was summarized in Table 5-5 of Chapter 5. There was no statistically significant difference between the performance of the nonoperant and operant perceptrons, $t(197.99) = 0.95501$, $p = .3407$.

Finally, let us consider the structure of these perceptrons. Table 6-12 presents the logistic regression coefficients for each training set, as well as the average bias and weights of the 20 perceptrons that learned about each of the training sets using the gambler's fallacy paradigm. The final row of Table 6-12 provides the mean

Table 6-11. The probability of reward (P(R)) and the mean responses of perceptrons after being trained on the different conditionally independent three-cue training sets using the gambler's fallacy operant paradigm.

	Training Set 1		**Training Set 2**		**Training Set 3**		**Training Set 4**		**Training Set 5**	
Input	P(R)	Mean Response	P(R)	Mean Response	P(R)	Mean Response	P(R)	Mean Response	P(R)	Mean Response
0,0,0	0.000	0.040	0.000	0.044	0.000	0.041	0.000	0.036	0.000	0.037
0,0,1	0.000	0.137	0.000	0.113	0.000	0.113	0.000	0.106	0.000	0.128
0,1,0	0.000	0.104	0.000	0.126	0.000	0.120	0.000	0.115	0.000	0.128
0,1,1	0.475	0.305	0.450	0.287	0.450	0.289	0.425	0.292	0.525	0.359
1,0,0	0.500	0.339	0.450	0.277	0.425	0.268	0.425	0.300	0.475	0.313
1,0,1	0.550	0.659	0.400	0.515	0.425	0.521	0.475	0.575	0.500	0.632
1,1,0	0.450	0.585	0.450	0.547	0.450	0.539	0.500	0.598	0.500	0.634
1,1,1	0.725	0.841	0.750	0.770	0.725	0.776	0.750	0.824	0.700	0.867
	R^2	**0.811**	R^2	**0.806**	R^2	**0.821**	R^2	**0.871**	R^2	**0.792**

Note. Each mean summarizes the performance of 20 perceptrons. The final row indicates the squared correlation (R^2) between the eight perceptron responses and the eight probabilities of reward for each of the training sets.

Table 6-12. The comparison between components of a logistic regression (Logit) and the average structure of a perceptron (Network) for each of the five High

	Training Set 1		**Training Set 2**		**Training Set 3**		**Training Set 4**		**Training Set 5**	
Source	Logit	Network	Logit	Network	Logit	Network	Logit	Network	Logit	Network
θ	-3.318	-3.171	-3.525	-3.089	-3.523	-3.156	-3.630	-3.289	-3.264	-3.258
w	2.484	2.501	2.388	2.126	2.357	2.149	2.585	2.439	2.322	2.468
w	0.942	1.013	1.271	1.154	1.233	1.164	1.261	1.249	1.144	1.341
w	1.242	1.330	1.120	1.022	1.157	1.089	1.183	1.154	1.144	1.335
	R^2	**1.000**	R^2	**1.000**	R^2	**1.000**	R^2	**1.000**	R^2	**1.000**

Note. All these perceptrons were trained using the gambler's fallacy operant paradigm. The final row indicates the average squared correlation (R^2) between perceptron structure and logistic regression coefficients.

squared correlation between the regression coefficients and the perceptron components for each training set. Each R^2 is over 0.99, demonstrating that the operant training led to the same perceptron structure predicted by our Chapter 4 formal analysis, exhibited by the nonoperant perceptrons trained on the same stimuli in Chapter 5, and revealed in our analysis of the increasing returns perceptrons described earlier in this chapter.

A comparison of the fit of the structure of nonoperant and operant perceptrons to regression coefficients revealed a statistically significant difference. We performed a Welch two-sample *t* test to compare the 100 R^2 values used to create the Table 5-6 data for nonoperant perceptrons (*M* value = 0.999) to the compare the 100 R^2 values used to create the Table 6-12 data for operant perceptrons (*M* value = 0.999). This comparison revealed a no statistically significant difference, $t(196.04) = -0.53181$, $p = .5955$. A second comparison of the R^2 measure of fit for the gambler's fallacy networks to the same measure for the increasing returns perceptrons did reveal a statistically significant difference, $t(99.352) = -9.484$, $p = 1.417\text{e-}15$. This indicates that the fits provided in Table 6-12 are better than those provided in Table 6-9. However, there is little to be made of this difference, as both models provide near perfect fits, and the difference between these two means is a mere 0.007.

6.4 Operant Learning: High-Reward XOR

Section 5.8 described the four training sets involving interactions between Cues B and C as representing the four cells of a 2×2 factorial design that crossed the logical nature of the interaction (linearly separable AND versus linearly nonseparable XOR) with the degree of conditional dependence between the interacting cues (high reward vs. low reward). The results just presented in Section 6.3 are related to one cell in this design. The current section is concerned with another cell, involving training sets that have a high reward associated with the XOR of Cues B and C.

6.4.1 Architecture, Training Sets, and Training

The methodology used to collect the results detailed next parallels that described in Sections 6.2 and 6.3. All the perceptrons consisted of three input units and one logistic output unit. Perceptrons were trained on one of the five High-Reward XOR training sets that were originally introduced in Section 5.4.1. These training sets were stochastically constructed so that the presence of Cue A signaled a reward probability of 0.37, and the XOR of Cues B and C signaled a reward probability of 0.36. The properties of the five training sets that were constructed were introduced earlier in Table 5-7. All perceptrons were trained using the same parameters that were used throughout Chapter 5 as well as earlier in the current chapter. Connection weights were randomly initialized in the range from −0.1 to 0.1, the output unit bias was initialized to 0, and the learning rate was 0.05. Each network was trained for 2,500 epochs using the gradient descent rule, with each epoch involving the presentation of 320 stimuli in a random order. One hundred networks (20 for each training set) were trained using the increasing returns paradigm. Another 100 networks (20 for each training set) were trained using the gambler's fallacy paradigm.

6.4.2 Increasing Returns: Performance

Let us begin by discussing the performance of perceptrons that were trained on the High-Reward XOR problem using the increasing returns paradigm. Averaging across the 20 perceptrons trained on each of these sets, Table 6-13 reports the mean number of times that each stimulus was selected for learning using this operant procedure. An inspection of Table 6-13 reveals that when a stimulus is associated with a higher probability of reward it is more frequently selected for learning. Table 6-13 also reports the actual probability of reward associated with each stimulus in a training set and provides the correlation between this probability and the average number of times that a stimulus is selected for learning. These correlations range from about 0.67 to 0.77, indicating that the operant perceptrons exhibit a strong degree of probability matching with respect to their choices but not as high as was seen with this same operant technique in the previous two sections. This is not surprising, because the results in Chapter 5 lead us to expect that probability matching will not be as accurate for these particular training sets.

Table 6-13 also provides the average of the total number of patterns selected for learning by the 20 perceptrons trained on each training set. As can be seen in the Total Choices row of Table 6-13, these operant perceptrons received less training than did the nonoperant perceptrons in Chapter 5, which were trained on 100,000 stimuli. Table 6-13 shows that the total amount of training ranged from about 25,000 total patterns to about 29,000 total patterns. This is a similar range to that observed in Tables 6-1 and 6-7.

Table 6-14 provides the average response (after training is complete) of a perceptron to each of the eight stimulus types for each of the training sets. This average is the mean of the responses generated by the 20 perceptrons trained on each training set.

A comparison between different stimulus conditions in our 2×2 factorial design is presented later in this chapter once results for each condition have been presented. For

Table 6-13. The average number of times that a perceptron selected each of the input patterns for learning using the increasing returns operant training paradigm when learning the High

	Set 1		**Set 2**		**Set 3**		**Set 4**		**Set 5**	
Probability	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**
P(R|~A~B~C)	0	18807.75	0	17257.9	0	20166.8	0	16046.65	0	16704.15
P(R|~A~BC)	0.4	15581.75	0.35	13583.65	0.35	20094.35	0.325	17816.6	0.325	17135.45
P(R|~AB~C)	0.35	21344.3	0.35	21569.8	0.45	20163.4	0.35	16029.2	0.375	17999.65
P(R|~ABC)	0	17707	0	17106	0	20008.85	0	17749.65	0	18483
P(R|A~B~C)	0.4	50489.45	0.325	44991.15	0.425	53049.35	0.35	50080.95	0.4	49496.85
P(R|A~BC)	0.575	44788.95	0.55	37956.3	0.65	52801.25	0.675	53109.2	0.6	50230.25
P(R|AB~C)	0.65	54358.15	0.625	51844.4	0.625	52916.65	0.65	49949.35	0.6	51662.65
P(R|ABC)	0.375	48674.8	0.325	44738.45	0.425	52817.8	0.4	53066.15	0.45	52608.45
Total Choices		**271752.2**		**249047.7**		**292018.5**		**273847.8**		**274320.5**
Correlation		**0.707**		**0.671**		**0.715**		**0.744**		**0.772**

Note. Each average is the mean of 20 perceptrons. The table also provides the probability of reward for each of the patterns in each of the five training sets. The second-to-last row provides the average number of total choices made (from 100,000 different presentations) for each training set. The bottom row provides the correlation between the probability of reward and the average number of times each pattern was chosen for each of the training sets.

Table 6-14. The probability of reward (P(R)) and the mean responses of perceptrons after being trained on the different High

	Training Set 1		**Training Set 2**		**Training Set 3**		**Training Set 4**		**Training Set 5**	
Input	P(R)	Mean Response	P(R)	Mean Response	P(R)	Mean Response	P(R)	Mean Response	P(R)	Mean Response
0,0,0	0.000	0.189	0.000	0.171	0.000	0.198	0.000	0.158	0.000	0.162
0,0,1	0.400	0.156	0.350	0.135	0.350	0.198	0.325	0.177	0.325	0.164
0,1,0	0.350	0.216	0.350	0.212	0.450	0.193	0.350	0.158	0.375	0.174
0,1,1	0.000	0.179	0.000	0.170	0.000	0.192	0.000	0.176	0.000	0.176
1,0,0	0.400	0.507	0.325	0.453	0.425	0.527	0.350	0.503	0.400	0.491
1,0,1	0.575	0.448	0.550	0.386	0.650	0.526	0.675	0.536	0.600	0.494
1,1,0	0.650	0.548	0.625	0.519	0.625	0.518	0.650	0.502	0.600	0.512
1,1,1	0.375	0.489	0.325	0.451	0.425	0.518	0.400	0.535	0.450	0.516
	R^2	**0.498**	R^2	**0.452**	R^2	**0.509**	R^2	**0.554**	R^2	**0.597**

Note. Each mean summarizes the performance of 20 perceptrons. The final row indicates the squared correlation (R^2) between the eight perceptron responses and the eight probabilities of reward for each of the training sets.The results reported in Table 6-14 indicate much weaker relationships between perceptron responses and the actual probabilities of reward than were observed in Tables 6-2 or 6-8. For instance, the worst performance (Training Set 2) produces an R^2 of only 0.452. The best performance, for Training Set 5, produces a relatively small R^2 of 0.597. This level of performance is noticeably poorer than that observed for either the independent cue training sets or the High-Reward AND training sets.

the time being, let us simply compare the performance summarized in Table 6-14 to the nonoperant perceptrons trained on the same patterns, whose performance was summarized in the previous chapter in Table 5-8. We performed a Welch two-sample *t* test to compare the 100 R^2 values used to create the Table 5-8 data for nonoperant perceptrons (*M* value = 0.525) to the 100 R^2 values used to create the Table 6-14 data for increasing returns perceptrons (*M* value = 0.521). This comparison revealed no statistically significant difference, $t(197.63) = 0.57492$, $p = .566$.

Given this result, the structure of these operant perceptrons should comprise the same odds ratios used by logistic regression to map cue patterns onto the likelihood of reward, and discovered inside the perceptrons described in Chapter 5. Table 6-15 presents the logistic regression coefficients for each training set, as well as the average bias and weights of the 20 perceptrons that learned about each

Table 6-15. The comparison between components of a logistic regression (Logit) and the average structure of a perceptron (Network) for each of the five High

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Source	Logit	Network	Logit	Network	Logit	Network	Logit	Network	Logit	Network
θ	-1.436	-1.457	-1.552	-1.583	-1.388	-1.399	-1.659	-1.671	-1.615	-1.646
w	1.467	1.483	1.376	1.393	1.513	1.508	1.670	1.684	1.602	1.608
w	0.000	0.165	0.096	0.268	0.092	-0.035	0.064	-0.006	0.160	0.087
w	-0.062	-0.236	-0.096	-0.273	-0.092	-0.003	0.064	0.133	0.000	0.015
	R^2	**0.987**	***R^2***	**0.987**	***R^2***	**0.994**	***R^2***	**0.998**	***R^2***	**0.999**

Note. All these perceptrons were trained using the increasing returns operant paradigm. The final row indicates the average squared correlation (R^2) between perceptron structure and logistic regression coefficients.

of the training sets using the increasing returns operant paradigm. The final row of Table 6-9 provides the mean squared correlation between the regression coefficients and the perceptron components for each training set. Each R^2 is over 0.98, demonstrating that the operant training led to the same perceptron structure predicted by our Chapter 4 formal analysis, and exhibited by the nonoperant perceptrons trained on the same stimuli in Chapter 5.

A comparison of the fit of the structure of nonoperant and operant perceptrons to regression coefficients revealed a statistically significant difference. We performed a Welch two-sample *t* test to compare the 100 R^2 values used to create the Table 5-10 data for nonoperant perceptrons (*M* value = 0.999) to the compare the 100 R^2 values used to create the Table 6-9 data for operant perceptrons (*M* value = 0.992). This comparison revealed a statistically significant difference, $t(99.774) = 8.5717$, $p = 1.338\text{e-}13$, despite that the two means only differ by 0.007.

6.4.3 Gambler's Fallacy: Performance

Let us now consider the performance of the other perceptrons that were trained on the same problem using the gambler's fallacy procedure. Table 6-16 reports the mean number of times that each stimulus was selected for learning using the gambler's fallacy operant paradigm, averaging over the 20 perceptrons trained on each training set it. An inspection of Table 6-16 reveals the expected negative correlation between a stimulus's probability of reward and the number of times that it is selected for learning. For each of the training sets, the correlation between these two measures is moderately high but is in

Table 6-16. The average number of times that a perceptron selected each of the input patterns for learning using the gambler's fallacy operant training paradigm.

	Set 1		Set 2		Set 3		Set 4		Set 5	
Probability	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**
P(R|~A~B~C)	0	80752.7	0	82444.95	0	79877.9	0	83967.4	0	83270.45
P(R|~A~BC)	0.4	81786.05	0.35	84001.2	0.35	81594.05	0.325	83075.6	0.325	83461.15
P(R|~AB~C)	0.35	80710	0.35	80959.6	0.45	78189.55	0.35	83168.55	0.375	81452.35
P(R|~ABC)	0	81639.05	0	82516.25	0	79911.5	0	82203.7	0	81552.2
P(R|A~B~C)	0.4	49149.4	0.325	54219.55	0.425	46519.95	0.35	49447.45	0.4	50008.95
P(R|A~BC)	0.575	50741	0.55	57020.25	0.65	49250	0.675	47930.6	0.6	50281.5
P(R|AB~C)	0.65	49063.3	0.625	51684.6	0.625	43929.05	0.65	48096.25	0.6	46925.25
P(R|ABC)	0.375	50649.85	0.325	54359.6	0.425	46537.6	0.4	46330.8	0.45	47083.25
Total Choices		**524491.4**		**547206**		**505809.6**		**524220.4**		**524035.1**
Correlation		**-0.707**		**-0.674**		**-0.714**		**-0.743**		**-0.772**

Note. Each average is the mean of 20 perceptrons. The table also provides the probability of reward for each of the patterns in each of the five training sets. The second-to-last row provides the average number of total choices made (from 100,000 presentations) for each training set. The bottom row provides the correlation between the probability of reward and the average number of times each pattern was chosen for each of the training sets.

the opposite direction of the relationship observed for the other operant paradigm (Table 6-13).

The perceptrons trained with the gambler's fallacy paradigm once again chose to learn about approximately twice the number of patterns in comparison to the perceptrons trained with the other operant procedure. Table 6-16 indicates that on average these perceptrons chose to learn about between 52,000 and 55,000 of the total of 100,000 stimuli that they were presented, as indicated in the Total Choices row.

Table 6-17 provides the average response of a perceptron to each of the eight stimuli for each of the training sets. This average is the mean of the responses generated by the 20 perceptrons trained on each training set. As was the case in Table 6-14, these responses are less accurate estimates of reward probabilities, as shown by the bottom row of Table 6-17. The worst performance, for Training Set 2, produces an R^2 of .456, indicating that perceptron responses capture less than half of the variance in the expected probabilities. The best performance, for Training Set 5, produces an R^2 of .597, indicating that perceptron responses capture just under 60% of the variance in the expected probabilities. All of the values reported in Table 6-17 are extremely close in value to those reported for the other operant procedure in Table 6-14.

We performed a Welch two-sample t test to compare the 100 R^2 values used to create the Table 6-14 data for increasing returns perceptrons (M value = 0.521) to the 100 R^2 values used to create the Table 6-17 data for gambler's fallacy perceptrons (M value = 0.523). This comparison revealed no statistically significant difference, $t(197.76) = -0.41847$, $p = .6761$). A similar test was performed to compare the performance of the gambler's fallacy perceptrons to the nonoperant perceptrons whose performance was summarized in Table 5-8 of Chapter 5. This comparison also did not reveal a statistically significant difference, $t(197.99) = 0.15763$, $p = .8749$.

Finally, let us consider the structure of these perceptrons. Table 6-18 presents the logistic regression coefficients for each training set, as well as the average bias and weights of the 20 perceptrons that learned about each of the training sets using the increasing returns paradigm. The final row of Table 6-18 provides the mean squared correlation between the regression coefficients and the perceptron components for each training set. Each R^2 is 1.00, demonstrating that the operant training led to the same perceptron structure predicted by our Chapter 4 formal analysis, exhibited by the nonoperant perceptrons trained on the same stimuli in Chapter 5, and revealed in our analysis of the increasing returns perceptrons described earlier in this chapter.

A comparison of the fit of the structure of nonoperant and operant perceptrons to regression coefficients revealed a statistically significant difference. We performed a Welch two-sample t test to compare the 100 R^2 values used to create the Table 5-9 data for nonoperant perceptrons (M value = 0.999) to the compare the 100 R^2 values used to create the Table 6-18 data for operant perceptrons (M value = 0.998). This comparison revealed a statistically significant difference, $t(151.53) = 3.4378$, $p = .0007575$. A comparison of these fits to the regression model to those observed for the increasing returns perceptrons did reveal a statistically significant difference, $t(101.7) = -8.0786$, $p = 1.403e-12$, that indicates that the

Table 6-17. The probability of reward (P(R)) and the mean responses of perceptrons after being trained on the different High

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Input	P(R)	Mean Response	P(R)	Mean Response	P(R)	Mean Response	P(R)	Mean Response	P(R)	Mean Response
0,0,0	0.000	0.191	0.000	0.174	0.000	0.198	0.000	0.160	0.000	0.162
0,0,1	0.400	0.180	0.350	0.160	0.350	0.184	0.325	0.169	0.325	0.164
0,1,0	0.350	0.195	0.350	0.193	0.450	0.216	0.350	0.170	0.375	0.181
0,1,1	0.000	0.184	0.000	0.178	0.000	0.200	0.000	0.180	0.000	0.183
1,0,0	0.400	0.509	0.325	0.458	0.425	0.531	0.350	0.512	0.400	0.490
1,0,1	0.575	0.491	0.550	0.434	0.650	0.508	0.675	0.529	0.600	0.493
1,1,0	0.650	0.516	0.625	0.489	0.625	0.558	0.650	0.530	0.600	0.523
1,1,1	0.375	0.498	0.325	0.465	0.425	0.535	0.400	0.547	0.450	0.526
	R^2	**0.502**	**R^2**	**0.456**	**R^2**	**0.512**	**R^2**	**0.554**	**R^2**	**0.597**

Note. Each mean summarizes the performance of 20 perceptrons. The final row indicates the squared correlation (R^2) between the eight perceptron responses and the eight probabilities of reward for each of the training sets.

Table 6-18. The comparison between components of a logistic regression (Logit) and the average structure of a perceptron (Network) for each of the five High

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Source	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**
θ	-1.436	-1.446	-1.552	-1.559	-1.388	-1.398	-1.659	-1.659	-1.615	-1.646
w	1.467	1.484	1.376	1.391	1.513	1.525	1.670	1.707	1.602	1.606
w	0.000	0.027	0.096	0.125	0.092	0.105	0.064	0.072	0.160	0.133
w	-0.062	-0.073	-0.096	-0.098	-0.092	-0.093	0.064	0.068	0.000	0.014
	R^2	**1.000**	R^2	**1.000**	R^2	**1.000**	R^2	**1.000**	R^2	**1.000**

Note. All these perceptrons were trained using the gambler's fallacy operant paradigm. The final row indicates the average squared correlation (R2) between perceptron structure and logistic regression coefficients.

fits provided in Table 6-18 are better than those provided in Table 6-15. However, there is little to be made of this difference, as both models provide near perfect fits, and the difference between means is a mere 0.006.

6.5 Operant Learning: Low-Reward AND

We now turn to the third cell of our 2×2 factorial design by describing operant learning for training sets that have a low reward associated with the AND of Cues B and C.

6.5.1 Architecture, Training Sets, and Training

The methodology used to collect the results detailed next parallels that described in Sections 6.2 through 6.4. All the perceptrons were networks that consisted of three input units and one logistic output unit. Perceptrons were trained on one of the five LowReward AND training sets that were originally introduced in Section 5.6.1. These training sets were stochastically constructed so that the presence of Cue A signaled a reward probability of 0.6, and the AND of Cues B and C signaled a reward probability of 0.18. The properties of the five training sets that were constructed were introduced earlier in Table 5-10. All perceptrons were trained using the same parameters that were used throughout Chapter 5 as well as earlier in the current chapter. Connection weights were randomly initialized in the range from −0.1 to 0.1, the output unit bias was initialized to 0, and the learning rate was 0.05. Each network was trained for 2,500 epochs using the gradient descent rule, with each epoch involving the presentation of 320 stimuli in a random order. One hundred networks (20 for each training set) were trained using the increasing returns operant paradigm. Another 100 networks (20 for each training set) were trained using the gambler's fallacy operant paradigm.

6.5.2 Increasing Returns: Performance

Let us begin by discussing the performance of perceptrons that were trained on the Low-Reward AND problem using the increasing returns paradigm. Averaging across the 20 perceptrons trained on each of these sets, Table 6-19 reports the mean number of times that each stimulus was selected for learning using this operant procedure. An inspection of Table 6-19 reveals that when a stimulus is associated with a higher probability of reward it is more frequently selected for learning using this operant procedure. Table 6-19 also reports the actual probability of reward associated with each stimulus in a training set and provides the correlation between this probability and the average number of times that a stimulus is selected for learning. These correlations range from about 0.96 to 0.99, indicating that the operant perceptrons exhibit a very high degree of probability matching with respect to their choices. This is not surprising, because the results in Chapter 5 lead us to expect that probability matching will be accurate for these training sets.

Table 6-19 also provides the average of the total number of patterns selected for learning by the 20 perceptrons trained on each training set. As can be seen in the Total Choices row of Table 6-19, these operant perceptrons received less training than did the nonoperant perceptrons in Chapter 5, which were trained on 100,000 stimuli. Table 6-19 shows that the total amount of training ranged from about 25,000 total patterns to about 30,000 total patterns.

Next, Table 6-20 provides the average response (after training is complete) of a perceptron to each of the eight stimuli for each of the training sets. This average is the mean of the responses generated by the 20 perceptrons trained on each training set. The results reported in Table 6-20 indicate very strong relationships between perceptron responses and the actual probabilities of reward than were observed in Tables 6-2 or 6-8. For instance, the worst

Table 6-19. The average number of times that a perceptron selected each of the input patterns for learning using the increasing returns operant training paradigm when learning the Low

	Set 1		Set 2		Set 3		Set 4		Set 5	
Probability	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**
P(R|~A~B~C)	0	5756.2	0	6822.75	0	5709.5	0	7016.05	0	7792.1
P(R|~A~BC)	0	6654.05	0	6912.85	0	7486.2	0	6429.2	0	9100.85
P(R|~AB~C)	0	8685.2	0	8191.6	0	8269.8	0	9458.45	0	8361.25
P(R|~ABC)	0.175	10073.7	0.2	8358.7	0.175	10684.75	0.225	8722.5	0.25	9779.05
P(R|A~B~C)	0.5	48879.5	0.6	55759.8	0.65	60072.05	0.675	63412.5	0.6	57343.2
P(R|A~BC)	0.525	53083.15	0.525	56216.6	0.625	66922.65	0.575	61174.55	0.6	61634.5
P(R|AB~C)	0.6	60365.9	0.575	60846.45	0.65	69435.85	0.675	70917.75	0.575	59201.9
P(R|ABC)	0.65	64219.8	0.65	61243.6	0.8	75276.9	0.725	69019.7	0.65	63458.75
Total Choices		**257717.5**		**264352.4**		**303857.7**		**296150.7**		**276671.6**
Correlation		**0.986**		**0.973**		**0.986**		**0.974**		**0.966**

Note. Each average is the mean of 20 perceptrons. The table also provides the probability of reward for each of the patterns in each of the five training sets. The second-to-last row provides the average number of total choices made (from 100,000 presentations) for each training set. The bottom row provides the correlation between the probability of reward and the average number of times each pattern was chosen for each of the training sets.

Table 6-20. The probability of reward (P(R)) and the mean responses of perceptrons after being trained on the different Low

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Input	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**
0,0,0	0.000	0.046	0.000	0.054	0.000	0.047	0.000	0.058	0.000	0.067
0,0,1	0.000	0.055	0.000	0.054	0.000	0.063	0.000	0.055	0.000	0.080
0,1,0	0.000	0.072	0.000	0.068	0.000	0.070	0.000	0.079	0.000	0.072
0,1,1	0.175	0.085	0.200	0.068	0.175	0.092	0.225	0.074	0.250	0.086
1,0,0	0.500	0.488	0.600	0.554	0.650	0.606	0.675	0.636	0.600	0.576
1,0,1	0.525	0.534	0.525	0.554	0.625	0.675	0.575	0.620	0.600	0.622
1,1,0	0.600	0.604	0.575	0.611	0.650	0.699	0.675	0.708	0.575	0.592
1,1,1	0.650	0.647	0.650	0.612	0.800	0.758	0.725	0.694	0.650	0.637
	R^2	**0.972**	R^2	**0.946**	R^2	**0.972**	R^2	**0.948**	R^2	**0.932**

Note. Each mean summarizes the performance of 20 perceptrons. The final row indicates the squared correlation (R^2) between the eight perceptron responses and the eight probabilities of reward for each of the training sets.

performance (Training Set 5) produces an R^2 of 0.932, indicating that perceptron responses capture well over 90% of the variance in the expected probabilities. The best performance, for Training Sets 1 and 3, produces an R^2 of 0.972, indicating that perceptron responses capture just over 97% of the variance in the expected probabilities.

We performed a Welch two-sample *t* test to compare the 100 R^2 values used to create the Table 5-12 data for non-operant perceptrons (*M* value = 0.954) to the 100 R^2 values used to create the Table 6-20 data for operant perceptrons (*M* value = 0.954). This comparison revealed no statistically significant difference, $t(196.3) = 0.41419$, $p = .6792$.

Given this result, it would be expected that the structure of these operant perceptrons comprised the same odds ratios used by logistic regression to map cue patterns onto the likelihood of reward. Table 6-21 presents the logistic regression coefficients for each training set, as well as the average bias and weights of the 20 perceptrons that learned about each of the training sets using the increasing returns operant paradigm. The final row of Table 6-21 provides the mean squared correlation between the regression coefficients and the perceptron components for each training set. Each R^2 is greater than 0.98, demonstrating that the operant training led to the same perceptron structure predicted by our Chapter 4 formal analysis and exhibited by the nonoperant perceptrons trained on the same stimuli in Chapter 5.

A comparison of the structure of nonoperant and operant perceptrons to regression coefficients revealed a statistically significant difference. We performed a Welch two-sample *t* test to compare the 100 R^2 values used to create the Table 5-13 data for nonoperant perceptrons (*M* value = 0.995) to the 100 R^2 values used to create the Table 6-21 data for operant perceptrons (*M* value = 0.990). This comparison revealed a statistically significant difference, $t(124.79) = 8.8174$, $p = 8.556\text{e-}15$, despite that the two means only differ by 0.005.

6.5.3 Gambler's Fallacy: Performance

Section 6.5.2 described the performance of perceptrons on the Low-Reward AND problem when the increasing returns operant paradigm was employed. Let us now consider the performance of the other perceptrons that were trained on the same problem using the gambler's fallacy operant procedure.

Table 6-22 reports the mean number of times that each stimulus was selected for learning using the gambler's fallacy paradigm, averaging over the 20 perceptrons trained on each training set. An inspection of Table 6-22 reveals the expected negative correlation between a stimulus's probability of reward and the number of times that it is selected for learning. For each of the training sets, the correlation between these two measures is very high but is

Table 6-21. The comparison between components of a logistic regression (Logit) and the average structure of a perceptron (Network) for each of the five Low

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Source	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**
θ	-3.747	-3.028	-3.427	-2.861	-3.829	-3.005	-3.393	-2.786	-3.282	-2.631
w	3.453	2.980	3.351	3.078	3.977	3.436	3.582	3.346	3.212	2.936
w	0.715	0.470	0.525	0.236	0.699	0.411	0.693	0.329	0.473	0.067
w	0.450	0.186	0.351	0.002	0.600	0.300	0.325	-0.070	0.558	0.192
	R^2	**0.996**	**R^2**	**0.991**	**R^2**	**0.995**	**R^2**	**0.986**	**R^2**	**0.984**

Note. All these perceptrons were trained using the increasing returns operant paradigm. The final row indicates the average squared correlation (R^2) between perceptron structure and logistic regression coefficients.

Table 6-22. The average number of times that a perceptron selected each of the input patterns for learning using the gambler's fallacy operant training paradigm.

	Set 1		Set 2		Set 3		Set 4		Set 5	
Probability	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**
P(R|~A~B~C)	0	96298.45	0	95252.05	0	96372.7	0	95344.75	0	94811.1
P(R|~A~BC)	0	95033.8	0	94372.1	0	94565.65	0	94102.75	0	91910.95
P(R|~AB~C)	0	93607.5	0	93317.95	0	93969.7	0	91746.65	0	92508.2
P(R|~ABC)	0.175	91475.55	0.2	92049.65	0.175	90870.95	0.225	89577.9	0.25	88451.95
P(R|A~B~C)	0.5	53625.85	0.6	47461.3	0.65	41956.9	0.675	41884.95	0.6	48287.05
P(R|A~BC)	0.525	45458	0.525	42844.8	0.625	31384.1	0.575	35742.6	0.6	36416.3
P(R|AB~C)	0.6	38954.6	0.575	38291.75	0.65	29044.5	0.675	27754.15	0.575	38417.25
P(R|ABC)	0.65	31534.85	0.65	33929.65	0.8	20651.55	0.725	22895.05	0.65	27828
Total Choices		**545988.6**		**537519.3**		**498816.1**		**499048.8**		**518630.8**
Correlation		**-0.985**		**-0.973**		**-0.985**		**-0.973**		**-0.964**

Note. Each average is the mean of 20 perceptrons. The table also provides the probability of reward for each of the patterns in each of the five training sets. The second-to-last row provides the average number of total choices made (from 100,000 presentations) for each training set. The bottom row provides the correlation between the probability of reward and the average number of times each pattern was chosen for each of the training sets.

in the opposite direction of the relationship observed for the other operant paradigm (Table 6-19).

The perceptrons trained with the gambler's fallacy paradigm once again chose to learn about approximately twice the number of patterns than did the perceptrons trained with the other operant procedure. Table 6-22 indicates that, on average, these perceptrons chose to learn about between 50,000 and 55,000 of the total of 100,000 stimuli that they were presented, as indicated in the Total Choices row.

Next, Table 6-23 provides the average response of a perceptron to each of the eight stimulus types for each of the training sets. This average is the mean of the responses generated by the 20 perceptrons trained on the same training set. As was the case in Table 6-20, these responses are very accurate estimates of reward probabilities. The worst performance, for Training Sets 5, produces an R^2 of .929. The best performance, for Training Set 3, produces an R^2 of .970. All the values reported in Table 6-23 are close in value to those reported for the other operant procedure in Table 6-20.

We performed a Welch two-sample *t* test to compare the 100 R^2 values used to create the Table 6-20 data for increasing returns perceptrons (*M* value = 0.954) to the 100 R^2 values used to create the Table 6-23 data for gambler's fallacy perceptrons (*M* value = 0.951). This comparison revealed no statistically significant difference, $t(196.28) = 1.4471$, $p = .1495$. A similar test was performed to compare the performance of the gambler's fallacy perceptrons to the nonoperant perceptrons whose performance was summarized in Table 5-12 of Chapter 5. This comparison also did not reveal a statistically significant difference, $t(196.3) = 0.41419$, $p = .6792$.

Finally, let us consider the structure of these perceptrons. Table 6-24 presents the logistic regression coefficients for each training set, as well as the average bias and weights of the 20 perceptrons that learned about each of the training sets using the increasing returns operant paradigm. The final row of Table 6-24 provides the mean squared correlation between the regression coefficients and the perceptron components for each training set. Each

Table 6-23. The probability of reward (P(R)) and the mean responses of perceptrons after being trained on the different Low

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Input	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**
0,0,0	0.000	0.034	0.000	0.044	0.000	0.033	0.000	0.043	0.000	0.049
0,0,1	0.000	0.046	0.000	0.053	0.000	0.051	0.000	0.056	0.000	0.077
0,1,0	0.000	0.061	0.000	0.065	0.000	0.058	0.000	0.080	0.000	0.073
0,1,1	0.175	0.082	0.200	0.078	0.175	0.089	0.225	0.103	0.250	0.112
1,0,0	0.500	0.460	0.600	0.524	0.650	0.581	0.675	0.580	0.600	0.515
1,0,1	0.525	0.540	0.525	0.572	0.625	0.686	0.575	0.644	0.600	0.630
1,1,0	0.600	0.611	0.575	0.621	0.650	0.717	0.675	0.726	0.575	0.616
1,1,1	0.650	0.684	0.650	0.665	0.800	0.800	0.725	0.777	0.650	0.720
	R^2	**0.969**	**R^2**	**0.945**	**R^2**	**0.970**	**R^2**	**0.945**	**R^2**	**0.929**

Note. Each mean summarizes the performance of 20 perceptrons. The final row indicates the squared correlation (R^2) between the eight perceptron responses and the eight probabilities of reward for each of the training sets.

Table 6-24. The comparison between components of a logistic regression (Logit) and the average structure of a perceptron (Network) for each of the five Low

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Source	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**
θ	-3.747	-3.356	-3.427	-3.071	-3.829	-3.390	-3.393	-3.095	-3.282	-2.964
w	3.453	3.195	3.351	3.168	3.977	3.718	3.582	3.418	3.212	3.026
w	0.715	0.615	0.525	0.397	0.699	0.605	0.693	0.655	0.473	0.413
w	0.450	0.321	0.351	0.193	0.600	0.455	0.325	0.272	0.558	0.472
	R^2	**0.999**	**R^2**	**0.998**	**R^2**	**0.999**	**R^2**	**1.000**	**R^2**	**0.999**

Note. All these perceptrons were trained using the gambler's fallacy operant paradigm. The final row indicates the average squared correlation (R^2) between perceptron structure and logistic regression coefficients.

R^2 is greater than .99, demonstrating that the operant training led to the same perceptron structure predicted by our Chapter 4 formal analysis, exhibited by the nonoperant perceptrons trained on the same stimuli in Chapter 5, and revealed in our analysis of the increasing returns perceptrons described earlier in this chapter.

We performed a Welch two-sample *t* test to compare the 100 R^2 values used to create the Table 5-13 data for nonoperant perceptrons (*M* value = 0.995) to the compare the 100 R^2 values used to create the Table 6-24 data for operant perceptrons (*M* value = 0.998). This comparison revealed a statistically significant difference, $t(144.5) = -17.894$, $p = 2.2\text{e-}16$, indicating that the gambler's fallacy perceptrons reliably generated better fits to the coefficients, though the absolute value of this difference was quite small. A comparison of these fits to the regression model to those observed for the increasing returns perceptrons also revealed a statistically significant difference, $t(105.38) = -16.386$, $p = 2.2\text{e-}16$.

6.6 Operant Learning: Low-Reward XOR

Finally, we turn to the last cell of our 2×2 factorial design by considering operant learning for training sets that have a low reward associated with the XOR of Cues B and C.

6.6.1 Architecture, Training Sets, and Training

The methodology used to collect the results detailed next parallels that described in Sections 6.2 through 6.5. All the perceptrons consisted of three input units and one logistic output unit. Perceptrons were trained on one of the five Low-Reward XOR training sets that were originally introduced in Section 5.7.1. These training sets were stochastically constructed so that the presence of Cue A signaled a reward probability of 0.53, and the AND of Cues B and C signaled a reward probability of 0.18. The properties of the five training sets that were constructed were introduced earlier in Table 5-14. All perceptrons were trained using the same parameters that were used throughout Chapter 5 as well as earlier in the current chapter. Connection weights were randomly initialized in the range from −0.1 to 0.1, the output unit bias was initialized to 0, and the learning rate was 0.05. Each network was trained for 2,500 epochs using the gradient descent rule, with each epoch involving the presentation of 320 stimuli in a random order. One hundred networks (20 for each training set) were trained using the increasing returns operant paradigm. Another 100 networks (20 for each training set) were trained using the gambler's fallacy operant paradigm.

6.6.2 Increasing Returns: Performance

Let us begin by discussing the performance of perceptrons that were trained on the Low-Reward XOR problem using the increasing returns paradigm. Averaging across the 20 perceptrons trained on each of these sets, Table 6-25 reports the mean number of times that each stimulus was selected for learning using the operant procedure. An inspection of Table 6-25 reveals that when a stimulus is associated with a higher probability of reward, it is more frequently selected for learning using this operant procedure. Table 6-25 also reports the actual probability of reward associated with each stimulus in a training set and provides the correlation between this probability and the average number of times that a stimulus is selected for learning. These correlations range from about 0.96 to 0.99, indicating that the operant perceptrons exhibit a very high degree of probability matching with respect to their choices. This is not surprising, because the results in Chapter 5 lead us to expect that probability matching will be accurate for these training sets.

Table 6-25 also provides the average of the total number of patterns selected for learning by the 20 perceptrons trained on each training set. The total amount of training ranged from about 25,000 patterns to about 28,000 patterns. This range is similar to that observed for other perceptrons we have seen that were also trained with the increasing returns paradigm.

Table 6-26 provides the average response (after training is complete) of a perceptron to each of the eight stimuli for each of the training sets. This average is the mean of the responses generated by the 20 perceptrons trained on each training set. The results reported in Table 6-26 indicate strong relationships between perceptron responses and the actual probabilities of reward. For instance, the worst performance (Training Set 5) produces an R^2 of .864, whereas the best performance (Training Set 4) produces an R^2 of .933.

We performed a Welch two-sample *t* test to compare the 100 R^2 values used to create the Table 5-15 data for nonoperant perceptrons (*M* value = 0.904) to the 100 R^2 values used to create the Table 6-26 data for operant perceptrons (*M* value = 0.904). This comparison revealed no statistically significant difference, $t(198) = 0.21832$, $p = .8274$.

Given this result, it would be expected that the structure of these operant perceptrons is comprised of the same odds ratios used by logistic regression to map cue patterns onto the likelihood of reward and discovered inside the perceptrons described in Chapter 5. Table 6-27 presents the logistic regression coefficients for each training set, as well as the average bias and weights of the 20 perceptrons

Table 6-25. The average number of times that a perceptron selected each of the input patterns for learning using the increasing returns operant training paradigm when learning the Low

	Set 1		Set 2		Set 3		Set 4		Set 5	
Probability	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**
P(R|~A~B~C)	0	10682.4	0	10829.15	0	10236.4	0	8997.2	0	9934.3
P(R|~A~BC)	0.175	9691	0.175	10880.15	0.225	8373.8	0.125	8932.55	0.175	10562.5
P(R|~AB~C)	0.2	10748.7	0.2	9050.95	0.175	12345.6	0.175	8187.1	0.225	10770.55
P(R|~ABC)	0	9724.7	0	9049.7	0	10118.3	0	8141.15	0	11424.45
P(R|A~B~C)	0.55	61095.3	0.6	62223.95	0.525	59999.95	0.525	57343.65	0.45	52732.4
P(R|A~BC)	0.65	58328.6	0.65	62337.6	0.625	54328	0.625	57196.5	0.625	54478.55
P(R|AB~C)	0.675	61268.5	0.6	57309.85	0.725	64924.8	0.6	54813.7	0.625	55018.35
P(R|ABC)	0.525	58590.4	0.55	57363.15	0.525	59712.95	0.5	54695.3	0.5	56780.95
Total Choices		**280129.6**		**279044.5**		**280039.8**		**258307.2**		**261702.1**
Correlation		**0.954**		**0.964**		**0.940**		**0.966**		**0.930**

Note. Each average is the mean of 20 perceptrons. The table also provides the probability of reward for each of the patterns in each of the five training sets. The second-to-last row provides the average number of total choices made (from 100,000 presentations) for each training set. The bottom row provides the correlation between the probability of reward and the average number of times each pattern was chosen for each of the training sets.

Table 6-26. The probability of reward (P(R)) and the mean responses of perceptrons after being trained on the different Low

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Input	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**
0,0,0	0.000	0.097	0.000	0.098	0.000	0.093	0.000	0.077	0.000	0.092
0,0,1	0.175	0.086	0.175	0.097	0.225	0.075	0.125	0.077	0.175	0.097
0,1,0	0.200	0.098	0.200	0.082	0.175	0.113	0.175	0.070	0.225	0.101
0,1,1	0.000	0.087	0.000	0.080	0.000	0.092	0.000	0.071	0.000	0.106
1,0,0	0.550	0.608	0.600	0.619	0.525	0.595	0.525	0.578	0.450	0.536
1,0,1	0.650	0.577	0.650	0.614	0.625	0.539	0.625	0.579	0.625	0.551
1,1,0	0.675	0.611	0.600	0.569	0.725	0.645	0.600	0.554	0.625	0.561
1,1,1	0.525	0.580	0.550	0.564	0.525	0.592	0.500	0.555	0.500	0.575
	R^2	**0.910**	R^2	**0.930**	R^2	**0.884**	R^2	**0.933**	R^2	**0.864**

Note. Each mean summarizes the performance of 20 perceptrons. The row indicates the squared correlation (R^2) between the eight perceptron responses and the eight probabilities of reward for each of the training sets.

that learned about each of the training sets using the increasing returns operant paradigm. The final row of Table 6-27 provides the mean squared correlation between the regression coefficients and the perceptron components for each training set. Each R^2 is greater than .99.

We performed a Welch two-sample *t* test to compare the 100 R^2 values used to create the Table 5-16 data for non-operant perceptrons (*M* value = 0.999) to the 100 R^2 values used to create the Table 6-27 data for operant perceptrons (*M* value = 0.998). This comparison did not reveal a statistically significant difference, $t(190.48) = -1.0776$, $p = .2826$.

6.6.3 Gambler's Fallacy: Performance

Section 6.6.2 described the performance of perceptrons on the Low-Reward XOR problem when the increasing returns paradigm was employed. Let us now consider the performance of the other perceptrons that were trained on the same problem using the gambler's fallacy procedure.

Table 6-28 reports the mean number of times that each stimulus was selected for learning using the gam-

Table 6-27. The comparison between components of a logistic regression (Logit) and the average structure of a perceptron (Network) for each of the five Low

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Source	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**
θ	-2.232	-2.236	-2.193	-2.215	-2.198	-2.283	-2.473	-2.489	-2.274	-2.288
w	2.676	2.676	2.676	2.700	2.604	2.669	2.764	2.805	2.400	2.433
w	0.038	0.014	-0.115	-0.207	0.076	0.215	0.000	-0.099	0.148	0.101
w	-0.115	-0.130	-0.038	-0.019	-0.076	-0.227	-0.079	0.005	0.000	0.057
	R^2	**1.000**	R^2	**0.999**	R^2	**0.997**	R^2	**0.999**	R^2	**0.999**

Note. All these perceptrons were trained using the increasing returns operant paradigm. The final row indicates the average squared correlation (R2) between perceptron structure and logistic regression coefficients.

bler's fallacy paradigm, averaging over the 20 perceptrons trained on each training set. An inspection of Table 6-28 reveals the expected negative correlation between a stimulus's probability of reward and the number of times that it is selected for learning. For each of the training sets, the correlation between these two measures is very high but is in the opposite direction of the relationship observed for the other operant paradigm (Table 6-25).

Table 6-28 indicates that the perceptrons trained with the gambler's fallacy procedure learn about approximately twice the number of patterns than did the perceptrons trained with the other operant procedure. Table 6-28 indicates, on average, these perceptrons chose to learn about between 51,000 and 55,000 of the total of 100,000 stimuli that they were presented, as indicated in the Total Choices row.

Table 6-23 provides the average response of a perceptron to each of the eight stimuli for each of the training sets. This average is the mean of the responses generated by the 20 perceptrons trained on each training set. As was the case in Table 6-20, these responses are very accurate estimates of reward probabilities, as shown by the measure of goodness of fit provided in the bottom row of Table 6-23. The worst performance, for Training Set 5, produces an R^2 of .864, whereas the best performance (Training Set 2) produces an R^2 of .930.

We performed a Welch two-sample *t* test to compare the 100 R^2 values used to create the Table 6-26 data for increasing returns perceptrons (*M* value = 0.904) to the 100 R^2 values used to create the Table 6-29 data for gambler's fallacy perceptrons (*M* value = 0.903).

Table 6-28. The average number of times that a perceptron selected each of the input patterns for learning using the gambler's fallacy operant training paradigm.

	Set 1		Set 2		Set 3		Set 4		Set 5	
Probability	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**	**P(R)**	**Times Chosen**
P(R|~A~B~C)	0	90094.9	0	89889.65	0	89937.4	0	92008.55	0	90519.65
P(R|~A~BC)	0.175	91125.7	0.175	90083.35	0.225	90756.95	0.125	92381.45	0.175	90418.15
P(R|~AB~C)	0.2	89816.45	0.2	90944.2	0.175	89104.35	0.175	92232.6	0.225	89331.55
P(R|~ABC)	0	90809.75	0	91135.4	0	89989.05	0	92620.5	0	89123.2
P(R|A~B~C)	0.55	38792.65	0.6	38084.65	0.525	39811.4	0.525	42496.25	0.45	46768.95
P(R|A~BC)	0.65	41658.15	0.65	38584.5	0.625	42061.2	0.625	43743.6	0.625	46379.95
P(R|AB~C)	0.675	38065.95	0.6	41123.2	0.725	37771	0.6	43320.25	0.625	43359.85
P(R|ABC)	0.525	40791.8	0.55	41628.2	0.525	39890.15	0.5	44605.5	0.5	42990.3
Total Choices		**521155.4**		**521473.2**		**519321.5**		**543408.7**		**538891.6**
Correlation		**-0.954**		**-0.964**		**-0.940**		**-0.966**		**-0.929**

Note. Each average is the mean of 20 perceptrons. The table also provides the probability of reward for each of the patterns in each of the five training sets. The second-to-last row provides the average number of total choices made (from 100,000 presentations) for each training set. The bottom row provides the correlation between the probability of reward and the average number of times each pattern was chosen for each of the training sets.

This comparison revealed no statistically significant difference, *t*(197.98) = 0.027326, *p* = .9782. This comparison also did not reveal a statistically significant difference, *t*(197.99) = 0.24458, *p* = .807.

Finally, let us consider the structure of these perceptrons. Table 6-30 presents the logistic regression coefficients for each training set, as well as the average bias and weights of the 20 perceptrons that learned about each of the training sets using the increasing returns operant paradigm. The final row of Table 6-30 provides the mean squared correlation between the regression coefficients and the perceptron components for each training set. Each R^2 is 1.00, demonstrating that the operant training led to the same perceptron structure predicted by our Chapter 4 formal analysis, exhibited by the nonoperant perceptrons trained on the same stimuli in Chapter 5, and revealed in our analysis of the increasing returns perceptrons described earlier in this chapter.

A comparison of the fit of the structure of nonoperant and operant perceptrons to regression coefficients was again conducted. We performed a Welch two-sample *t* test to compare the 100 *R*2 values used to create the Table 5-17 data for nonoperant perceptrons (*M* value = 0.9995) to the compare the 100 *R*2 values used to create the Table 6-24 data for operant perceptrons (*M* value = 0.9995). This comparison did not reveal a statistically significant difference, *t*(190.48) = −1.0776, *p* = .2826. A comparison of these fits to the regression model to those observed for the increasing returns perceptrons did reveal a statistically significant difference, *t*(115.43) = −6.7085, *p* = 7.673e-10.

6.7 Operant Learning and Conditional Dependence

Sections 6.2 through 6.6 described a number of simulations that involved using two operant learning paradigms to train perceptrons on the same training sets that

Table 6-29. The probability of reward (P(R)) and the mean responses of perceptrons after being trained on the different Low

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Input	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**	**P(R)**	**Mean Response**
0,0,0	0.000	0.096	0.000	0.097	0.000	0.098	0.000	0.076	0.000	0.092
0,0,1	0.175	0.088	0.175	0.095	0.225	0.090	0.125	0.073	0.175	0.094
0,1,0	0.200	0.098	0.200	0.087	0.175	0.106	0.175	0.074	0.225	0.106
0,1,1	0.000	0.090	0.000	0.085	0.000	0.098	0.000	0.071	0.000	0.108
1,0,0	0.550	0.613	0.600	0.611	0.525	0.601	0.525	0.568	0.450	0.533
1,0,1	0.650	0.589	0.650	0.606	0.625	0.577	0.625	0.557	0.625	0.538
1,1,0	0.675	0.619	0.600	0.581	0.725	0.622	0.600	0.560	0.625	0.572
1,1,1	0.525	0.594	0.550	0.577	0.525	0.599	0.500	0.549	0.500	0.576
	R^2	**0.910**	R^2	**0.930**	R^2	**0.884**	R^2	**0.933**	R^2	**0.864**

Note. Each mean summarizes the performance of 20 perceptrons. The final row indicates the squared correlation (R^2) between the eight perceptron responses and the eight probabilities of reward for each of the training sets.

Table 6-30. The comparison between components of a logistic regression (Logit) and the average structure of a perceptron (Network) for each of the five Low

	Training Set 1		Training Set 2		Training Set 3		Training Set 4		Training Set 5	
Source	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**	**Logit**	**Network**
θ	-2.232	-2.244	-2.193	-2.235	-2.198	-2.220	-2.473	-2.498	-2.274	-2.289
w	2.676	2.705	2.676	2.686	2.604	2.629	2.764	2.771	2.400	2.424
w	0.038	0.024	-0.115	-0.122	0.076	0.090	0.000	-0.033	0.148	0.155
w	-0.115	-0.102	-0.038	-0.018	-0.076	-0.097	-0.079	-0.044	0.000	0.019
	R^2	**1.000**	R^2	**1.000**	R^2	**1.000**	R^2	**1.000**	R^2	**1.000**

Note. All these perceptrons were trained using the gambler's fallacy operant paradigm. The final row indicates the average squared correlation (R^2) between perceptron structure and logistic regression coefficients.

were explored in Chapter 5. The earlier sections of this chapter have established that in spite of the radical differences between the increasing demands procedure and the gambler's fallacy paradigm, differences that lead to very different experiences of the various training sets, both of these operant techniques produce nearly identical networks to the nonoperant perceptrons discussed in Chapter 5.

We now turn to exploring the effects of the various differences between training sets. In Chapter 5, we identified a number of different results based on such comparisons. First, ANOVAs concerned with the ability of perceptron responses to estimate probabilities revealed that these were significantly better for linearly separable interactions between Cues B and C than for linearly nonseparable interactions between these cues. We also saw that performance was significantly better when interactions between Cues B and C were associated with low rewards than with high rewards. We finally noted that there was a significant interaction between these two factors; probability estimations with high reward, linearly nonseparable interactions were significantly poorer than probability estimations in other conditions. Second, we discovered using multiple regression that the degree of conditional dependence related to the interaction between Cues B and C could predict almost all of the variance of network probability estimates, suggesting that this was the key predictor to consider when uncertain environments were of interest.

Given that each operant network that has been discussed earlier in the current chapter produces nearly identical behavior to a corresponding nonoperant perceptron from Chapter 5, we should expect to replicate these key results from Chapter 5. To begin, let us consider using ANOVA to explore the effects of the two factors, interaction type and amount of reward, that were crossed in the four sets of simulations presented in Sections 6.3 through 6.6.

6.7.1 Analysis of Variance

To begin, let us consider the performance of perceptrons trained using the increasing returns procedure. Table 6-31 summarizes the basic data for this by presenting the average measure of probability matching performance for the 100 perceptrons in each of the four conditions defined by this two-factor design.

An inspection of Table 6-31 reveals several interesting findings. First, it is nearly identical to Table 5-18, which summarized probability estimations learned by nonoperant perceptrons. Second, on average networks perform better when there is an AND relationship between Cues B and C than when there is an XOR relationship between these cues. Third, on average networks perform better when Cues B and C combine to signal a lower probability of reward than when they combine to signal a higher probability of reward. Fourth, changing the probability of reward has a much larger effect on network performance for the XOR versions of the training sets than for AND versions of the training sets.

Table 6-31. The mean probability matching performance (with standard deviations in parentheses) of perceptrons as a function of problem type and level of reward.

	High-Reward	Low-Reward
AND of Cues B and C	0.82 (0.03)	0.95 (0.02)
XOR of Cues B and C	0.52 (0.05)	0.90 (0.03)

Note. All the perceptrons summarized in this table were trained with the increasing returns procedure. Probability matching performance is operationalized as the squared correlation between the perceptrons responses for each of eight stimuli and the actual reward probabilities for these same stimuli. Each mean is based on the performance of 100 perceptrons.

An ANOVA of the data used to produce Table 6-31 confirms these general observations. We conducted this analysis on the $R2$ scores for the networks in different conditions to test the effects of interaction type and level of reward. This analysis revealed a significant main effect of type, $F(1, 396) = 2840$, $p < 2\text{e-}16$, $\eta2 = 0.267$; a significant main effect of reward, $F(1, 396) = 5948$, $p < 2\text{e-}16$, $\eta2 = 0.559$; and a significant Type × Reward interaction, $F(1, 396) = 1462$, $p < 2\text{e-}16$, $\eta2 = 0.134$. Post hoc tests conducted using the Tukey HSD statistic revealed that perceptrons trained on the AND interaction learned to be more accurate probability estimators than did perceptrons trained on the XOR interaction ($p < 1.0\text{e-}16$). As well, perceptrons trained in conditions in which the interacting cues signaled low reward were more accurate probability estimators than networks trained in conditions in which the interacting cues signaled high reward ($p < 1.0\text{e-}16$). The significant interaction emerged because the difference between Low-Reward XOR networks and High-Reward XOR networks was greater than the difference between the two reward conditions involving the AND interaction.

Table 6-32 provides a similar summary of probability estimation performance for the other set of perceptrons trained using the gambler's fallacy procedure. This table is nearly identical to Table 6-31. Thus it is not surprising that an ANOVA of the data used to produce Table 6-32 a significant main effect of type, $F(1, 396) = 2888$, $p < 2\text{e-}16$, $\eta2 = 0.258$; a significant main effect of reward, $F(1, 396) = 6399$, $p < 2\text{e-}16$, $\eta2 = 0.572$; and a significant Type × Reward interaction, $F(1, 396) = 1503$, $p < 2\text{e-}16$, $\eta2 = 0.134$.

Post hoc tests conducted using the Tukey HSD statistic revealed the same pattern of differences that were reported for Table 6-31. These results are also essentially the same as those reported for nonoperant perceptrons in Section 5.8

Table 6-32. The mean probability matching performance (with standard deviations in parentheses) of perceptrons as a function of problem type and level of reward.

	High-Reward	Low-Reward
AND of Cues B and C	0.82 (0.03)	0.95 (0.02)
XOR of Cues B and C	0.52 (0.05)	0.90 (0.03)

Note. All the perceptrons summarized in this table were trained with the gambler's fallacy procedure. Probability matching performance is operationalized as the squared correlation between the perceptrons responses for each of eight stimuli and the actual reward probabilities for these same stimuli. Each mean is based on the performance of 100 perceptrons.

6.7.2 Predicting Performance From G

As was noted earlier in Chapter 5, one problem with the ANOVAs just reported is that the logical structure of the relationship between Cues B and C is confounded with conditional dependence. This is because when conditional dependence is operationalized using *G*, on average this metric is higher for the training sets based on XOR than it is for the training sets based on AND (Dawson & Gupta, 2017). We can carry out two alternative analyses in light of this problem.

First, we can use our measure of conditional dependence for a training set (*G* as described in Section 5.5.1) to predict the R^2 fit between network responses and actual training set probabilities. This predicts network performance from degree of conditional dependence, and ignores the logical relationship between Cues B and C. When we perform this analysis for the increasing returns perceptrons, we find that the degree of conditional dependence accounts for just under 80% of the variance in the fit of network responses ($R^2 = 0.776$), $F(1, 398) = 1376.43$, p = 3e-131. We can contrast this result with a second analysis that predicts network performance using the dichotomous nature of the relationship between Cues B and C (i.e., XOR vs. AND). This result is significant as well but accounts for only one fourth of the variance in network performance accounted for by our measure of conditional dependence ($R^2 = 0.267$), $F(1, 398) = 144.8$, p = 1.17e-28. In short, the amount of conditional dependence provides a much better prediction of network performance than is provided by linear separability, though both are statistically significant predictors.

Similar results are obtained for the gambler's fallacy networks. Using *G* to predict the R^2 fit between network responses and actual training set probabilities also accounts for nearly 80% of the variance in the fit of network responses ($R^2 = .766$), $F(1, 398) = 1307.38$, p = 8e-128. Making the same prediction only on the basis of the relationship between Cues B and C (i.e., XOR vs AND) is significant as well but accounts for only fourth of the variance in network performance accounted for by our measure of conditional dependence ($R^2 = .258$), $F(1, 398) = 138.5$, p = 1.22E-27. In short, the amount of conditional dependence provides a much better prediction of network performance than is provided by linear separability, though both are statistically significant predictors

6.7.3 Comparison to Conditionally Independent Cues

The statistical analyses just reported proceeded by taking four of the simulations to represent cells in a 2×2 factorial design. In Chapter 5, another approach was also taken to examine the simulations: comparing the results of the four simulation studies that involve an interaction between Cues B and C to a fifth simulation, the initial study in which all three cues were conditionally independent (Section 6.2). Let us conduct the same kind of analysis for operant perceptrons, beginning with those trained using the increasing returns paradigms.

The means of the five conditions with their standard deviations are in Table 6-33. Each of these means summarizes the performance of 100 perceptrons; this performance is the R^2 between the responses of a perceptron to each of the eight stimulus types and the actual probability of reward associated with each.

The means in Table 6-33 are each associated with an extremely small standard deviation, indicating very similar probability estimates were generated by

Table 6-33. The mean probability matching performance (with standard deviations) of perceptrons trained using the increasing returns procedure.

	Independent Cues	High-Reward	High-Reward	Low-Reward	Low-Reward
Mean R^2	0.92	0.82	0.52	0.95	0.90
SD	0.04	0.03	0.05	0.02	0.03

Note. Probability matching is operationalized as the squared correlation between the response generated by a perceptron to each of eight possible stimuli and the actual probability of reward associated with each. Each mean is based upon the performance of 100 perceptrons.

different perceptrons trained in the same condition. Statistical comparisons between the independent cues condition and the other four conditions were performed by using Welch two-sample *t* tests. The results of these comparisons are presented below in Table 6-34. Every single *t* test indicated a significant difference between the means of the conditions being compared.

Table 6-34. The results of independent t tests used to compare the independent cues condition to the other four conditions from Table 6-33.

	High-Reward AND	High-Reward XOR	Low-Reward AND	Low-Reward XOR
Inde-pendent Cues	*18.894*	*61.05*	*−6.3789*	*4.0865*
	df = 178.5	*df = 193.6*	*df = 125.14*	*df = 165.01*
	p = 2.3e-16	*p = 2.2e-16*	*p = 3.142e-09*	*p = 6.822e-05*

Note. Each simulation used the increasing returns procedure to train perceptrons.

Let us now perform the same analysis for the gambler's fallacy perceptrons. The means of the five conditions with their standard deviations are in Table 6-35. Each of these means summarizes the performance of 100 perceptrons; this performance is the R^2 between the responses of a perceptron to each of the eight stimuli and the actual probability of reward associated with each.

Table 6-35. The mean probability matching performance (with standard deviations) of perceptrons trained using the gambler's fallacy procedure.

	Independent Cues	High-Reward	High-Reward	Low-Reward	Low-Reward
Mean R^2	0.92	0.82	0.52	0.95	0.90
SD	0.04	0.03	0.05	0.03	0.03

Note. Probability matching is operationalized as the squared correlation between the response generated by a perceptron to each of eight possible stimuli and the actual probability of reward associated with each. Each mean is based upon the performance of 100 perceptrons.

Statistical comparisons between the means presented in Table 6-35 were performed by using Welch two-sample *t* tests to compare the independent cue condition to the other four conditions. The results of these *t* tests are presented in Table 6-36.

Table 6-36. The results of independent t-tests used to compare the independent cues condition to the other four conditions from Table 6-33.

	High-Reward AND	High-Reward XOR	Low-Reward AND	Low-Reward XOR
Independent Cues	19.255	60.53	−6.7183	3.1779
	df = 167.23	*df = 196.28*	*df = 124.05*	*df = 163.71*
	p = 2.2e-16	*p = 2.2e-16*	*p = 5.933e-10*	*p = .001774*

Note. Each simulation used the gambler's fallacy procedure to train perceptrons.

The results provided in Tables 6-33 through 6-36 indicate that both operant learning procedures replicated the results that were observed for the nonoperant perceptrons that were detailed in Chapter 5.

6.8 Summary and Implications

6.8.1 Summary

Chapter 5 provided the results of five simulations in which perceptrons learn the probabilities of reward signaled by three cues. The results of the various simulations reported in Chapter 5 indicated that that perceptrons confronted with uncertain signals of reward behave as naïve Bayesian mechanisms. The limitations of perceptrons are best explained by appealing to the amount of conditional dependence between different cues.

Chapter 6 noted that one issue with the Chapter 5 perceptrons was that they were nonoperant. That is, they did not use any procedure to choose whether to learn about a stimulus. Chapter 6 began by describing two very different operant learning procedures that can be used to train perceptrons—one based on increasing returns, the other based on the gambler's fallacy. Chapter 6 then proceeded to use each of these techniques to train perceptrons on the same set of problems that were explored in Chapter 5.

Chapter 6 therefore considered five simulation conditions. For each of these conditions, the stimuli that were experienced by perceptrons trained with increasing returns were quite different from those experienced by perceptrons trained with the gambler's fallacy. For each training set, the choices made by increasing returns perceptrons had a very high positive correlation with the probabilities of stimulus in reward. In contrast, for each training set the choices made by gambler's fallacy perceptrons exhibited a very high negative correlation between the same variables. Furthermore, although increasing returns perceptrons chose to learn about one fourth to one third of the stimuli that were presented, the gambler's fallacy perceptrons chose to learn about approximately half of the presented stimuli.

In spite of these differences between training experiences, the main result of Chapter 6 was to replicate all of the Chapter 5 results. For each condition, and for either operant paradigm, the Chapter 6 perceptrons generated identical responses and network structures as were observed in the corresponding Chapter 5 networks. When comparisons were made between different conditions trained with the same operant paradigm, the same pattern of results was revealed as was discovered for the nonoperant perceptrons in Chapter 5.

6.8.2 Implications

One major implication of these results is that very different operant paradigms can produce identical results to those produced via nonoperant training. This is because both of the operant paradigms studied in this chapter led to sufficient exploration of all of the training stimuli. Looking back at any table in Chapter 6 that provides the number of times that each stimulus type was selected for training, one will see that each stimulus was chosen many times. This permitted negative feedback to lead the network to learn the optimal probability estimates, independent of which operant paradigm was used for stimulus selection.

Of course, this result will hold only for operant methods that permit sufficient training on each possible stimuli. One can imagine other operant rules that cause networks only to learn about a subset of stimuli; these rules would be expected to lead to very different probability estimates.

A second implication of these results is the surprising distinction between the procedure used to choose a stimulus for learning and the procedure used to modify probability estimates. If we were to look only at the choice behavior of the perceptrons, we would note striking differences between the selections made according to increasing returns and those made according to the gambler's fallacy. We might then be tempted to conclude that very different probability estimates underlie these very different patterns of behavior. However, our ability to examine the internal structure and the responses of perceptrons indicates that this conclusion is erroneous. Identical probability estimates can emerge from very different patterns of choice.

This has a further implication for the study of probability learning in biological agents. If different patterns of choice behavior can be mediated by identical probability estimates, then we should conduct experiments that not only record choice behavior but also collect responses that can be used to assess probability estimates. The preceding simulations have established that we cannot use different patterns of choices to conclude different probability models. In short, the probability matching behavior does not necessarily provide an accurate measure of what has been learned about reward probabilities.

We are now in a position to move away from simulation studies and to consider the results of an analogous experiment involving human participants. The next chapter describes the results of human performance on a probability discrimination task, the card-choice task that was introduced in Chapter 1. This task is one in which three cues signal the likelihood of reward associated with each of eight cards. Participants discover the probability structure of the task via operant learning: They choose a succession of cards and learn whether a card delivers a reward only after it has been chosen. The rules used to convert cues into reward probabilities can be manipulated, just as was the case with the three cue simulations that were described in Chapters 5 and 6. In fact, we can train human participants in versions of the card-choice task where the rules used to determine reward probability are identical to those used to create the various simulation conditions that we have been considering.

Chapter 7: Human Performance On The Card-Choice Task

The purpose of this chapter is to explore whether perceptrons can serve as potential models of human probability learning. Is the probability learning of humans similar to that of perceptrons with respect to the effects of conditional dependence? If not, then more sophisticated systems, such as multilayer networks, need to be considered. This chapter describes a study that presents human participants a tableau of eight different cards, each of which can be identified by a pattern of three different symbols. These patterns are analogous to the three-cue stimuli that were studied in the Chapter 5 and 6 simulations. Human participants took part in one of five conditions. In one, the three symbols were independent signals of reward. In the other four, there was one of two interactions (AND vs. XOR) between two of the symbols, and this interaction signaled one of two reward probabilities (high vs. low). In the first phase of the study, participants explored the environment by successively choosing cards and learning about the probability of reward associated with each card. Analysis of the choice behavior in this phase indicates that human choices are influenced by the manipulated variables in a similar fashion to that observed in perceptrons, which suggests that perceptrons offer potential models of human behavior. In the second phase of the study, participants were presented pairs of cards and chose the member of the pair that they believed was more likely to offer a reward based on their experiences in the first phase. Analyses of these preferences suggest that participants are affected by the type of interaction between variables in a manner similar to that observed in trained perceptrons. Such results suggest that when humans learn the probabilities of reward in the card-choice task, they behave as if they too are naïve Bayesians. This supports the hypothesis that perceptrons are potential models of human probability learning.

7.1 From Perceptrons to People

7.1.1 Perceptrons and Probabilities

In Chapter 4, we explored the ability of perceptrons to adapt to uncertain environments when reward probability is signaled by multiple cues. We did so by developing a formal description of perceptrons faced with this situation and then by using this formal description to make assertions about what perceptrons can and cannot learn. One key assertion was that perceptrons, because of their simple structure, are limited in their ability to estimate reward probabilities. Perceptrons are naïve Bayesian mechanisms. Constrained by their simple structure, perceptrons must treat cues as conditionally independent signals of reward probability. Perceptrons are blind to signals carried by interactions between different cues. As the probability of reward signaled by interacting cues increases, the ability of perceptrons to accurately estimate reward probabilities decreases. This was illustrated, for instance, via Chapter 4's discussion of perceptron performance on uncertain Boolean operators (e.g., Table 4-2).

The Chapter 4 proofs were developed for perceptrons that received signals from only two different cues. They revealed that the connection weights in such a perceptron represented a particular measure of an individual cue's influence, the natural logarithm of its odds ratio. These proofs were developed in such a way that they could be extended to more complex situations (i.e., situations in which more than two cues signal reward probability). Rather than continue this formal development, Chapters 5 and 6 used simulation studies to examine probability estimates when reward probabilities are signaled by three cues. It was shown that these simulations were consistent with the formal analyses of Chapter 4, because connection weights in trained networks attained values equal to the natural logarithms of odds ratios (i.e., these weights were strongly related to the coefficients of logistic regression equations fit to the same data). Previously published research established the same result for perceptrons that receive signals about reward probabilities from four different cues (Dawson & Gupta, 2017).

Dawson and Gupta (2017) were the first to study the probability estimates of perceptrons by manipulating the interactions between cues that signaled reward probability. Dawson and Gupta (Study 2) created four conditions by crossing two different factors that related to the interaction between these two cues. The first factor was the logical type of the interaction. Half of the training sets used the logical AND of two cues to signal reward probability. The other half of the training sets used the logical XOR of two cues to signal reward likelihood. The second factor was the amount of reward (more properly, the size of the probability of reward) signaled by the interacting cues regardless of the logical nature of the interaction. In half of the training sets, the interaction signaled a much higher probability of reward than was signaled by the interaction in the other half of the training sets.

Dawson and Gupta (2017) found a significant main effect of both manipulations. In general, perceptrons trained on stimuli that included an AND interaction were more accurate probability estimators than were perceptrons trained on stimuli that included an XOR interaction.

As well, perceptrons trained on stimuli for which the interaction signaled a lower probability of reward were more accurate probability estimators than were perceptrons trained on stimuli for which the interaction signaled a higher probability of reward. Furthermore, Dawson and Gupta discovered a significant interaction between their two factors: Decreasing the probability of reward associated with an XOR interaction led to a larger improvement in a perceptron's probability estimation performance than was produced by decreasing the probability of reward associated with an AND interaction.

Simulations reported in Chapters 5 and 6 replicate the results reported by Dawson and Gupta (2017) for perceptrons trained on three-cue stimuli. Chapter 5 revealed the same effects of interaction type and level of reward for nonoperant perceptrons, whereas Chapter 6 replicated these effects in perceptrons trained by two operant procedures. In short, we are currently armed with a substantial technical understanding of what perceptrons learn from uncertain environments, an understanding based on both formal analyses and simulation studies. We have established that perceptrons are naïve Bayesians and their estimates of the probabilities of reward signaled by different cues assume that each cue provides a signal that is conditionally independent from other signals.

7.1.2 Are People Naïve Bayesians?

An important question remains: Are these simple networks possible models of human probability learning? A primary goal of the current monograph is to use perceptrons to provide insights into the probability learning of humans and animals. To satisfy this goal, we must support the claim that perceptrons can provide these insights because they are potential models of probability learning in biological agents. The purpose of the current chapter is to provide such support. It does so by investigating whether human performance on the card-choice task is similar to the perceptrons described in Chapters 5 and 6.

How might we support the claim that perceptrons are possible models of human probability learning? One approach is to have human participants learn about reward probabilities signaled by cues in conditions that are analogous to those that we have explored in the computer simulations. In particular, imagine an environment in which different objects are identified by unique patterns of three cues and that these cues signal the probability that an object will provide a reward if it is selected. Imagine that in some conditions there exists an interaction between the signals of two of the cues. In this environment, we can manipulate the logical type of this interaction (AND vs. XOR), as well as the probability of reward associated with this interaction (high vs. low). How is human probability learning affected by these manipulations? Is it affected in a fashion similar to that of perceptrons?

One possibility is that human probability learning is not plausibly modeled by the perceptron. For instance, human probability learning might be better described by a more powerful model. If this is the case, then we might discover that human probability learning is not affected by the type of interaction between cues or by the probability of reward associated with this interaction. This is because if human probability learning is better modeled with a more powerful system, then it may not be limited by the constraints that we have explored in earlier chapters. In short, if human probability learning is accomplished by mechanisms that are more powerful than a perceptron, then human participants will not be naïve Bayesians, and their performance in the presence of interacting cues will be quite different from that of perceptrons. Modeling human performance in this situation will require using models that are more powerful, such as the multilayered networks that were briefly discussed in Section 1.7.

Another possibility is that human probability learning is similar to that of the perceptron, because human probability learners behave as if they are naïve Bayesians. If this is the case, then we should expect to see a pattern of results in human probability learning similar to what we have already observed in the computer simulations. In particular, human participants should be poorer probability learners when there is an XOR interaction between two cues than when there is an AND interaction. In addition, human performance should improve when the probability of reward associated with either type of interaction is lower in comparison to when this probability is higher. If these results were observed, it would suggest that perceptrons are plausible models of human performance in this task.

The purpose of this chapter is to describe the results of an experiment in which human participants learn probabilities of reward in different conditions that permit their performance to be compared to the performance of perceptrons. Before providing the results of this experiment, let us briefly discuss the different types of evidence that can be used to compare the performance of human participants to computer models in the context of the experiment's design.

7.1.3 Rationale for the Experiment

In cognitive science, there is a long history of comparing the performance of models to the performance of biological agents (Fodor, 1968; Pylyshyn, 1980, 1984). Pylyshyn described three general types of evidence for evaluating the relationship between models and participants.

The first is *relative complexity evidence*. Relative complexity evidence involves presenting a variety of stimuli to models and to human participants while measuring in some way the difficulty that each stimulus poses to the two types of systems (Pylyshyn, 1984). If one's model provides a valid account of one's participants, then the model and the human participants should exhibit the same relative difficulties with the problems.

The current chapter describes an experiment with the primary purpose of delivering relative complexity evidence. It explores human probability matching in the card-choice task that was briefly introduced in Chapter 1. In this task, eight stimuli are uniquely identified by a pattern of three different cues. These cues signal the probability that a stimulus will deliver a reward if it is chosen. Participants explore these stimuli for a while, making 320 different stimulus choices, with the goal of obtaining as many rewards as possible. It is assumed that this exploration gives participants an opportunity to learn the probability of reward associated with the various stimuli and that the better it is learned, the better will be a participant's performance on the task. It is also expected that participants will exhibit probability matching. That is, it is expected that there should be a strong relationship between the probability that a stimulus provides a reward and the probability that a participant chooses it during their exploration of the environment.

This experiment is designed to acquire relative complexity evidence by manipulating the rules that convert cue patterns into reward probabilities. Five conditions—the same five that were explored in the simulations reported in Chapters 5 and 6—are created. In one, the three cues are independent signals of reward likelihood. In the other four conditions, two of the cues signal reward probability by interacting. In parallel with Dawson and Gupta (2017), and with the simulations of Chapters 5 and 6, this interaction is one of two types (AND vs. XOR) and is associated with two probabilities of reward (high vs. low). If perceptrons plausibly model human performance in this task, then human performance will parallel perceptron performance. In particular, relative complexity evidence in favor of using perceptrons as models would be obtained if human participants were influenced in the same way by the two manipulations of cue interactions as were perceptrons.

Relative complexity evidence is not the only way that humans and models can be compared. A second type of information to use is *intermediate state evidence* (Pylyshyn, 1984). Intermediate state evidence is based on the assumption that an input/output mapping is not computed directly but instead requires a different stages of processing, with each stage representing an intermediate result in a different way. To collect intermediate state evidence, one attempts to determine the number and nature of these intermediate results. One example of this type of intermediate state evidence for probability matching would be examining rates of learning—that is, changes in probability matching to various types of stimuli over time. If both the model and the participant produce similar learning curves, then this would provide even more evidence of the strong relationship between the two, because of a correspondence between changes in information over time.

A third type of information to use to compare models and participants is *error evidence* (Pylyshyn, 1984). Imagine that perceptrons and humans provide relative complexity evidence indicating that both have difficulties learning probabilities in the same conditions. This relationship could be strengthened by examining the *kinds* of difficulties that the two systems have with more challenging tasks. For instance, what kinds of errors do the two systems make to particular stimuli? If we were to discover that the kinds of probability matching errors made to various stimuli were similar, this would strengthen our view that we had developed a valid model of our participants.

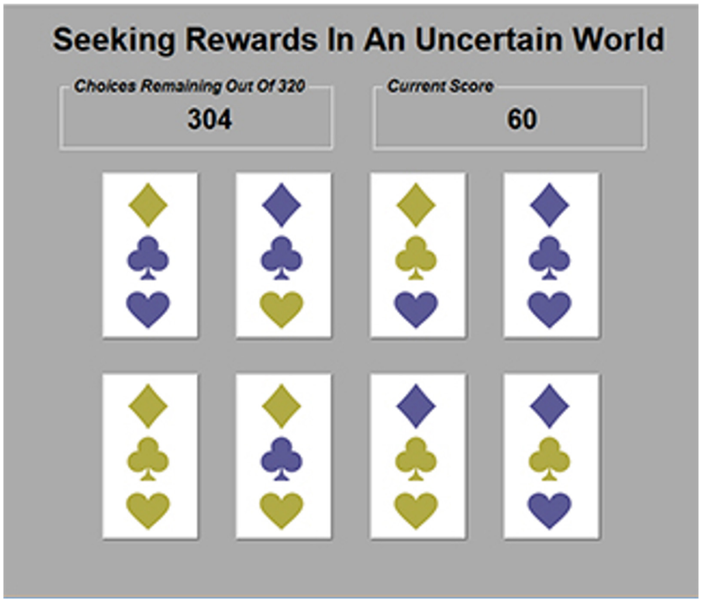
In summary, determining whether a particular model provides a valid or useful model of human behavior requires the systematic examination of this behavior, and its orderly comparison to that of the model. Before we use the limits of perceptrons to abandon them as models of human probability matching, we must first observe whether human probability matching exhibits similar limitations.

The remainder of Chapter 7 details an experiment that provides data that can be used to compare human probability matching to that of perceptrons via relative complexity evidence, intermediate state evidence, and error evidence. To foreshadow the major results, Chapter 7 shows that human probability matching exhibits similar limitations to those that have just been described. This suggests that perceptrons, in spite of their simplicity, are possible models of human probability matching, because humans—like perceptrons—behave like naïve Bayesians in the card-choice task.

7.2 Methodology for the Card-Choice Task

7.2.1 Overview of the Experiment

The card-choice task was briefly introduced in Section 1.2.3. It involves a computer environment that presents a tableau of eight different "cards" (Figure 7-1).



Seeking Rewards In An Uncertain World

Choices Remaining Out Of 320: 304

Current Score: 60

Figure 7-1. A view of the card tableau with which participants in the card-choice task interact. In this particular image, a participant has already made 16 card choices and has been rewarded for six of them.

Each card has three symbols on it (a diamond, a club, and a heart); each of these symbols can be either blue or green. When a participant uses the computer mouse to choose one of the eight cards, the person may receive a reward. Each card provides a reward with a certain probability, and participants are given the goal of obtaining as many rewards as possible. Thus, the card-choice task is an example of a probability discrimination task: To obtain many rewards, a participant must learn the probability structure of the environment and then use this knowledge to choose cards that are more likely to offer rewards. Good performance on this task requires participants to match probabilities.

The symbols on each card make the card-choice task analogous to the three-cue probability learning tasks that were studied in the Chapter 5 and Chapter 6 perceptron studies. The two possible colors of each symbol on a card are used to code two states of a particular cue. The probability that a particular card provides a reward is determined by a rule based on the states of the card's cues. However, it is clear from Figure 7-1 that all three cues are always present on each card; the only difference between "present" cues is their color. This might raise concerns that this task is not analogous to the simulations described in Chapters 5 and 6, because in those simulations cues were coded as being present (1) or not (0). To address these concerns, new sets of simulations that code present cues as being of one color (1) or not (2) are presented later in this chapter.

Table 7-1. The ideal probability of reward associated with individual cues in each of the five conditions of the card-choice task.

Cue	Indepen-dent	High-Reward-AND	High-Reward XOR	Low-Reward AND	Low-Reward XOR
A	0.14	0.48	0.37	0.60	0.53
B	0.2	-	-	-	-
C	0.4	-	-	-	-
AND (B, C)	-	0.48	-	0.18	-
XOR (B,C)	-	-	0.36	-	0.18

7.2.2 Human Participants

Two hundred introductory psychology students (135 female) served as participants in the experiment. Following approval from the Ethics Review Board at the University of Alberta, participants were recruited via the online system that manages the Introductory Psychology participant pool. Each participant was randomly assigned to one of the five conditions described in Section 7.2.3. Forty participants were randomly assigned to each condition.

7.2.3 Card-Choice Conditions

Each participant in the card-choice task took part in one experimental condition. Five conditions were created by using five different sets of rules to convert card cues into reward probabilities. These five conditions paralleled the five conditions used to study perceptron learning in previous chapters: one condition in which the three cues were conditionally independent signals of reward, and four conditions in which there was an interaction between two cues. These latter four conditions crossed type of interaction (AND vs. XOR) with probability of reward (high vs. low) in the same fashion as in the computer simulation studies considered earlier. Table 7-1 provides the probability of reward associated with each cue, or each cue interaction, in each of the five conditions of the card-choice task. These probabilities are identical to those used to create the training sets for each condition of the perceptron simulations described in previous chapters. These probability values were selected to ensure that the overall probability of reward in each condition was approximately 0.33.

As was the case in the earlier simulation studies, the additive probability rule was used to convert the individual probabilities of reward signaled by a card's cue states into an overall likelihood of reward. The resulting probability of reward associated with each card in each condition of the card-choice task is provided in Table 7-2. Note that the probabilities of reward in this table are identical to

Table 7-2. The ideal probability of reward in each condition for each of the eight cards in the card-choice task.

Cue Pattern	Probability	Independent	High-Reward AND	High-Reward XOR	Low-Reward AND	Low-Reward XOR
(0,0,0)	*P(R|~A~B~C)*	0.00	0.00	0.00	0.00	0.00
(0,0,1)	*P(R|~A~BC)*	0.4	0.00	0.36	0.00	0.18
(0,1,0)	*P(R|~AB~C)*	0.2	0.00	0.36	0.00	0.18
(0,1,1)	*P(R|~ABC)*	0.52	0.48	0.00	0.18	0.00
(1,0,0)	*P(R|A~B~C)*	0.14	0.48	0.37	0.60	0.53
(1,0,1)	*P(R|A~BC)*	0.484	0.48	0.5968	0.60	0.6146
(1,1,0)	*P(R|AB~C)*	0.312	0.48	0.5968	0.60	0.6146
(1,1,1)	*P(R|ABC)*	0.5872	0.7296	0.37	0.672	0.53

Note. In the Cue Pattern column, a 0 indicates that the cue is absent (i.e., is of one color) and a 1 indicates that the cue is present (i.e., is of the other color).

the probabilities of reward used to create the training sets in the various conditions of the simulation studies that were reported in Chapters 5 and 6.

The probabilities of reward provided in Table 7-2 were used to manage the rewards provided when a card was selected by a participant during the card-choice task. Participants saw all eight cards in a tableau at the same time (Figure 7-1) and used the mouse to choose a card. When selected, the probability that the card provided a reward was determined by the card's probability of reward as given in Table 7-2.

7.2.4 Card-Choice Task Instructions

Before beginning the card-choice task, groups of participants received instructions about the experiment. Participants saw a demonstration of the task, learned how they were to choose cards, and experienced what occurred when a selected card delivered a reward. They were told that each card generated a reward with a certain probability so that sometimes a card choice may be rewarded but that at other times the same card may not provide a reward. Participants were informed that the symbols on each card might provide information about reward probability, and were told that the location of a card was irrelevant. To emphasize this latter point, participants were informed that after each succession of 16 card choices the arrangement of the cards on their computer screen would be randomly shuffled. Participants were instructed that as they explored the environment they could learn about its reward structure and could use this knowledge to try to achieve a higher number of rewards.

After receiving these instructions, each participant moved to an individual testing room that separated them from all other participants and that contained a desktop computer running the Microsoft Windows operating system. The program that conducted the experiment was running when a participant entered this room; the participant initiated data collection by using the mouse to press a button labeled "Start the Study." When this was pressed, the computer program displayed the set of eight cards in a tableau (e.g., Figure 7-1). The locations of the cards in this tableau were randomly assigned. Participants then worked through their 320 card selections at their own pace. With each selection, the computer program recorded what card was selected, where it was located, and whether it provided a reward.

7.2.5 The Card-Choice Task Program

Data were collected using Microsoft Office Excel 2013 spreadsheet; the experiment was itself programed using Excel's Visual Basic for Applications. Before presenting the display to a participant, the computer program randomly selected which of the two symbol colors indicated that a cue was present. For the independent cues condition, the program randomly assigned reward probabilities (the independent cues column in Table 7-1) to symbols. For instance, for one participant, one symbol signaled a reward probability of 0.4, whereas a different symbol signaled this probability for another participant. For the other four conditions, the program randomly selected which cue was the independent signal of reward probability and which two cues interacted.

The computer program that conducted the study generated a reward for a selected card by using the card's reward probability given in Table 2. Whenever a card was selected by a participant, the program generated a random number between 0 and 1. If this number was greater than the card's Table 2 reward probability, then the card was not rewarded. Otherwise, the card was rewarded. A reward was signaled by generating a tone and by increasing the participant's displayed score by 10 points. After 16 card selections, the positions of the cards were randomly reassigned. This process was repeated until the participant had made 320 card choices.

7.2.6 Measuring Card Preferences

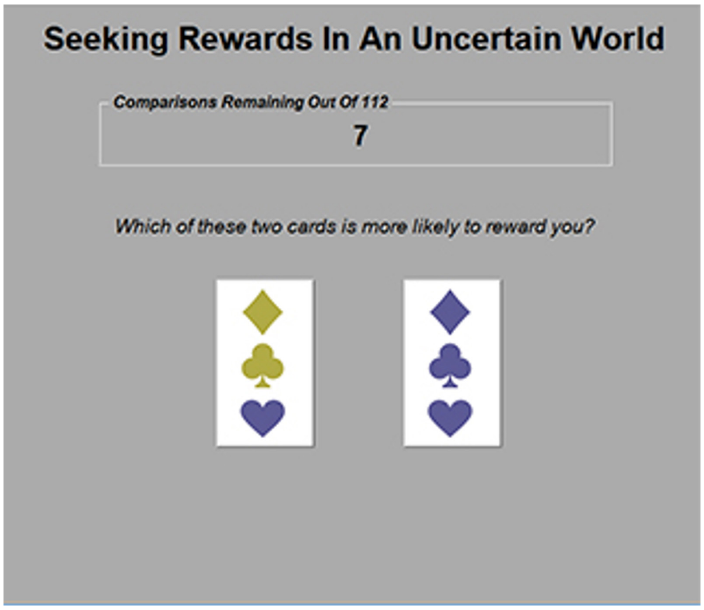
The card-choice task was the first phase in a two-phase experiment. First, participants explored the card-choice environment to learn the probability of reward associated with each card. It provides one kind of data that can be used to infer what participants learned: If participants are probability matchers, then the frequency with which they choose a card reflects their sense of its probability of reward.

However, the operant perceptron simulations that were presented in Chapter 6 demonstrated that very different patterns of choice behavior could be mediated by the same underlying knowledge of probability structure. For this reason, we sought an additional measure of what participants had learned about the environment and collected this information from participants in a second phase of the study that occurred immediately after they had completed choosing cards as just described.

Probability theory is often developed from an underlying theory of choice behavior (Savage, 1951, 1954, 1962). The basic idea is that if two events have different probabilities of reward, then a participant—if given the choice—will prefer the event with the higher probability of reward to the event with the lower probability of reward. This, of course, assumes that the participant values being rewarded. If the two events have identical reward probabilities, then the participant will not prefer one to the other.

The second phase of the card-choice experiment collected such preference data. When participants completed the card-choice task, the computer program presented a second button that enabled them to begin the next phase of the study when they were ready. In this phase, participants were presented a sequence of card pairs. Their task was to choose the card in each pair that they believed was more likely to provide a reward based on what they had learned about the cards in the first phase of the study. After choosing a card, the next pair of cards was presented. Unlike the card-choice task, participants never received any feedback about their choices. Participants had been instructed about this task and given a demonstration of how to proceed through it, as part of the instruction session at the beginning of the study. Figure 7-2 provides an example display to show what participants were interacting with in this second phase of the study.

Participants were presented 112 different pairs of cards in this card preference task. Each possible pair of cards was presented four times in this session. The two cards that were presented were always different from each other (i.e., a card was never compared to itself). In other words, participants were presented two cards (A and B) four times during this task; the position of the cards was counterbalanced (for some pair A and B, Card A appeared on the left in two of the pairs and on the right in the other two). The order of presentation of the 112 card pairs was randomized for each participant. With each choice, the computer recorded the pair of cards presented and the card that was preferred.


Seeking Rewards In An Uncertain World

Comparisons Remaining Out Of 112

7

Which of these two cards is more likely to reward you?

Figure 7-2. A view of the card display that participants interact with in the second phase of the study. In this phase, participants are presented a sequence of 112 card pairs like the one illustrated in the figure, and they choose the card they believe to be more likely to offer a reward.

7.3 Human Choice Behavior

We now turn to describing the results of the card-choice task. We begin by examining the results of the first phase of the study.

7.3.1 Combined Human Performance

For the participants in the first phase of the card-choice task, we do not have direct measures of probability estimates as we did have for the perceptron simulations reported earlier. Instead, we must rely on the phenomenon of probability matching: that the probability of choosing a stimulus reflects a participant's estimate of the probability that the stimulus will deliver a reward (Estes, 1964). Participants' data require preprocessing before this measure can be calculated because the program that conducted the experiment randomly assigned certain properties for

Table 7-3. The mean choice behavior of participants in each condition, taken over the number of times that each participant in each condition chose each of the eight stimuli.

Cue Pattern	Probability	High-Reward	High-Reward AND	Independent XOR	Low-Reward AND	Low-Reward XOR
(0,0,0)	$P(R\|{\sim}A{\sim}B{\sim}C)$	19.73	24.68	23.23	17.95	20.18
(0,0,1)	$P(R\|{\sim}A{\sim}BC)$	25.00	37.40	34.65	21.63	30.03
(0,1,0)	$P(R\|{\sim}AB{\sim}C)$	22.30	41.85	29.08	20.30	29.55
(0,1,1)	$P(R\|{\sim}ABC)$	43.78	30.00	50.05	34.68	29.73
(1,0,0)	$P(R\|A{\sim}B{\sim}C)$	37.68	39.85	28.23	43.45	44.73
(1,0,1)	$P(R\|A{\sim}BC)$	45.65	51.70	50.60	53.05	58.93
(1,1,0)	$P(R\|AB{\sim}C)$	46.40	56.13	39.83	54.33	55.73
(1,1,1)	$P(R\|ABC)$	79.48	38.40	64.35	74.63	51.15
	R	0.831	0.905	0.861	0.860	0.920

Note. The R^2 row provides the squared correlation between these columns of means and the appropriate column of reward probabilities from Table 7-2.

each participant, such as the color that signaled a cue was "present" (in terms of the probability rules of Table 7-2), and assigned which cue was the independent signal. The preprocessing involved relabeling each participant's cards so that different cards that served the same function for different participants were provided the same cue labels. For instance, for one participant the card with three blue cues might be the card for which all cues are present, whereas for a different participant the card with three green cues served this function instead. The preprocessing of the cards would give these two cards the same set of labels—that is, (1,1,1). In short, the purpose of preprocessing was simply to map each participant's cards to the appropriate row of Table 2.

After preprocessing, choice behavior can be determined for each participant in each condition. This simply involves counting the number of times that a participant chose each of the eight cards during the first phase of the study. Table 7-3 presents the mean number of times each card was selected by the 40 participants in each condition. When averaging over all of the participants in a condition, it can be seen that this measure is strongly related to the reward probabilities. The eight average choices in each column is the squared correlation between these choices and the appropriate probability of reward column from Table 7-2. The high values of R^2 in the bottom row of Table 7-3 indicate that the number of times that participants chose each of the eight stimuli is strongly related to the probability structure of the environment.

7.3.2 Individual Choice Behavior

The R^2 values at the bottom of Table 7-3 indicate a strong relationship between choice behavior and reward probability. Yet these values are also surprising. For instance, the values for the two XOR conditions are higher than the other values, which is quite different from the expectations generated from the simulation results reported in Chapters 5 and 6. However, the strong fit between the average choice performance in Table 7-3 and the reward probabilities in Table 7-2 hides the fact that within each condition there are substantial individual differences in the choice behaviors of participants, which is a common finding in the probability learning literature (Estes, 1964). Furthermore, individual participants in each condition tend to be far less accurate probability estimators than are the "average participants" represented by each column in Table 7-3.

To demonstrate this, we calculated the squared correlation between each participant's choices and the reward probabilities for his or her condition of the card-choice task. Figure 7-3 presents box plots of these results, with each box plot representing the distribution of the R^2 values computed for 40 participants. Figure 7-3 clearly shows that within each condition there is a great deal of variation between participants in terms of the relationship between their choice behavior and reward probabilities. This is reflected in the width of each box and the spread of each box's tails. In addition, the means of each distribution—the dark horizontal line through the middle of each box in Figure 7-3—exhibit a pattern that is much more consistent with the expectations generated by earlier simulations. Note, for instance, that the lowest mean is in the High-Reward XOR condition, whereas the highest mean is in the Low-Reward AND condition.

Table 7-4 presents the average R^2 value, with standard error, for each condition. Each average is the mean of the R^2 values computed for the 40 participants in a condi-

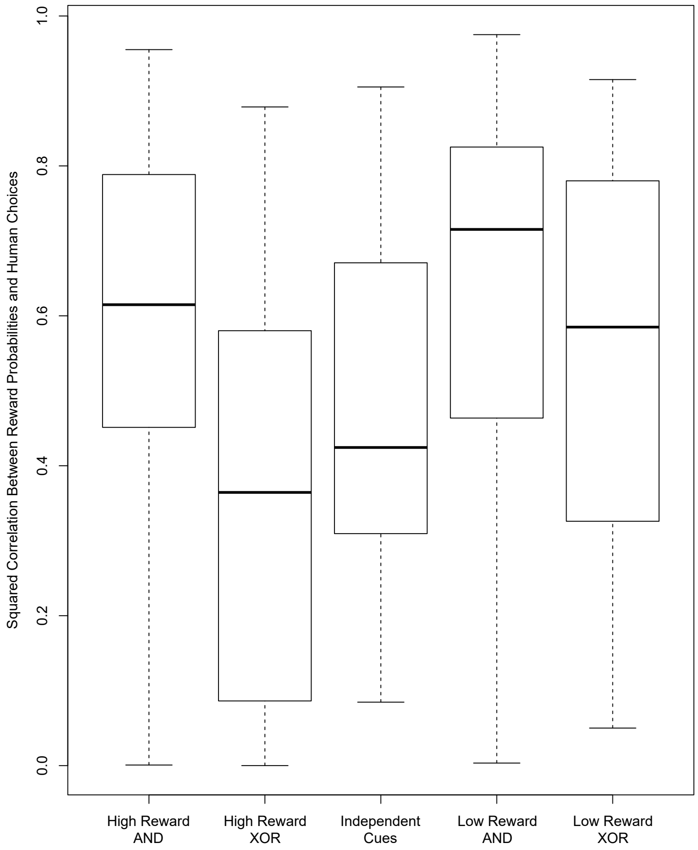


Figure 7-3. Box plots showing the distribution of R^2 values between the choice behavior of individuals and ideal reward probabilities for each of the conditions in the card-choice task. Each box plot summarizes the responses of 40 participants.

Table 7-4. The mean squared correlation between individual participant's choices of stimuli and the ideal reward probabilities.

	High-Reward AND	High-Reward XOR	Independent	Low-Reward AND	Low-Reward XOR
Mean R^2	0.569	0.380	0.480	0.626	0.550
SE	0.043	0.045	0.037	0.043	0.040

Note. The SE row provides the standard errors of these means, each of which is based on 40 participants.

tion. The values in this table are similar to those in earlier tables reporting the performance of perceptrons trained in analogous conditions, with one exception: Performance in the independent cues condition is poorer for human participants than for perceptrons. This result is discussed in more detail later in this chapter.

There is an interesting and important discrepancy between the results reported in Table 7-3 and those reported in Table 7-4. In essence, Table 7-3 shows that if one totals the stimulus choices made by each participant and then relates these choices to ideal reward probabilities, the choice behavior accurately reflects the reward probabilities in each condition. In contrast, Table 7-4 shows that if one does not combine the responses of all participants but instead relates the choices of each participant to ideal reward probabilities, then individual choice behavior is a much less accurate estimator of reward probabilities than is the choice behavior of the "average participants" presented in Table 7-3. What is the source of the differences between Tables 7-3 and 7-4?

Table 7-3 illustrates the phenomenon of *coarse coding*, which is well known in the artificial neural network literature (Hinton, McClelland, & Rumelhart, 1986; Van Gelder, 1991). In coarse coding, an accurate response of a network is achieved by combining a number of inaccurate estimates related to the response. These inaccurate estimates are typically produced by various hidden units in a multilayer perceptron. For example, networks can generate accurate judgments of the distance between points on a map, or the direction between these points, by combining multiple hidden unit responses, where each hidden unit represents a much more inaccurate measure of distance or direction (Dawson, Boechler, & Orsten, 2005; Dawson, Boechler, & Valsangkar-Smyth, 2000).

Coarse coding succeeds because each inaccurate component can be viewed as making a response that contains both signal and noise. If these components have a different perspective on the problem being solved, they will have a different source of noise but will be receiving similar information about the signal. When the components are combined, the different sources of noise tend to cancel one another out, leaving a better (i.e., less noisy) estimate of the signal than can be achieved by any component on its own. Figure 7-3 and Table 7-4 indicate that the choice behavior of each participant in the card-choice task may not be a highly accurate estimate of ideal reward probability. However, if the choice behavior of the participants is combined, there is a marked improvement between choice behavior and reward probability, presumably because there is a different pattern of error or noise in each participant's choices that is canceled out when responses are combined.

In the current chapter, our interest is in the choice behavior of individual participants, just as in earlier chapters we were concerned with the responses of individual perceptrons. In particular, how is the choice behavior of human participants affected by the different conditions in the card-choice task? Let us explore this question by studying the effects of different cue interactions on a variety of behaviors that were measured in the first phase of the study.

7.4 Cue Interactions and Relative Complexity Evidence

7.4.1 Exploring Relative Complexity

As noted in Section 7.1.3, one approach to comparing models and humans is to use relative complexity evidence. This type of evidence involves examining participant performance in different conditions, with the expectation that it will be poorer in some conditions than in others. To use relative complexity evidence to compare models to humans, one determines whether the same pattern of performance is exhibited by both models and human participants. If one has a plausible model of human performance, then the model and the human participants should similar differences of performance between conditions. That is, if the model performs worst in one condition, then this predicts that human participants should find this condition the most difficult. Similarly, if the model performs best in another condition, then this predicts that human participants should find this condition the easiest. Relative complexity evidence in essence examines human performance in light of these model predictions.

To explore relative complexity evidence in the card-choice task, let us focus on the four conditions that involve manipulating the type of interaction between cues as well as the probability of reward associated with this interaction. We saw in Chapters 5 and 6 that these manipulations affected the probability estimates made by perceptrons. In this section, our relative complexity evidence emerges from comparing the performance of human participants in the card-choice task to expectations generated from simulations reported earlier in Chapters 5 and 6.

7.4.2 Human Probability Matching

Section 7.3.2 described how we used the $R2$ statistic to measure the relationship between the frequency that each participant chose each of the eight cards and the probability of reward associated with each card. The mean values of this statistic for each condition were reported in Table 7-4. We now consider statistical differences between these means.

First, let us consider the four conditions of the card-choice task that involve an interaction between two of the three cues. These four conditions make up a 2×2 factorial design that crosses interaction type (XOR vs. AND) with probability of reward (high vs. low). We performed an ANOVA on this design to determine whether these two factors had similar effects on the human card choices as they did on perceptron responses.

This ANOVA reveals a significant main effect of type, $F(1, 156) = 9.573$, $p = .002$, $\eta2 = 0.549$, and a significant main effect of reward, $F(1, 156) = 7.088$, $p = .009$, $\eta2 = 0.041$. However, the Type × Reward interaction was not significant, $F(1, 156) = 0.129$, $p = .186$, $\eta2 = 0.01$. Post hoc tests conducted using the Tukey HSD statistic reveal that there was a significantly better fit between human choice frequencies and reward probabilities for participants who experienced the AND interaction than for those who experienced the XOR interaction ($p < .002$). As well, there was a significantly better relationship between choice frequencies and reward probabilities for human participants who experienced a low probability of reward condition than for participants who experienced a high probability of reward condition ($p = .009$). The two significant main effects were seen earlier in our computer simulation studies and are our first indications that human choices in the card-choice task are consistent with the view that participants are behaving as if they are naïve Bayesians.

7.4.3 Success in Obtaining Rewards

In earlier chapters, it was argued that one reason that probability learning is adaptive is because the better an agent understands the reward probabilities signaled by various cues, the more successful will the agent be in obtaining rewards. We can explore this in our human participants by examining whether they were more successful in obtaining rewards in some of the conditions of the card-choice task than in others. For the time being, we consider only the four conditions that involve an interaction between Cues B and C.

To perform this analysis, we simply counted the number of times that each participant in each condition was rewarded when a card was selected in the first phase of the card-choice task. Table 7-5 provides the mean number of successes for each condition (averaging over its 40 participants) along with the standard errors of these means.

We again used an ANOVA to examine the effects of type of interaction (AND vs. XOR) and amount of reward (high vs. low) on the number of rewards achieved. This analysis reveals a significant main effect of type, $F(1, 156) = 23.908$, $p = 2.49\text{e-}06$, $\eta2 = 0.128$, and a significant main effect of reward, $F(1, 156) = 6.192$, $p = .0139$, $\eta2 = 0.033$. The Type × Reward interaction was not significant, $F(1, 156) = 0.210$, $p = .647$, $\eta2 = .001$. Post hoc tests conducted using the Tukey HSD statistic reveal that human participants who explored the AND interaction obtained significantly more rewards than did those who explored the XOR interaction ($p < 2.5\text{e-}06$). As well, human participants who experienced a low reward condition obtained significantly more rewards than participants who did not (p = .014).

These results are consistent with the analyses of choice behavior in Section 7.4.2. The ANOVA reported in that section indicated that human choice behavior was more consistent with ideal reward probabilities for AND interactions than for XOR interactions, as well as when an interaction was associated with a low-reward probability in comparison a high-reward probability. The ANOVA of the data used to create Table 7-5 reveals this pattern of results. This agrees with our expectation that better learning of the reward contingencies in an environment will lead to a better ability to obtain rewards in that environment.

Table 7-5. The mean number of rewards (with standard deviations) obtained by each

	High-Reward AND	High-Reward XOR	Low-Reward AND	Low-Reward XOR
Mean Number of Rewards	141.10	122.73	148.08	132.85
SE	3.44	2.68	4.20	3.25

Note. Each mean is the average over the 20 participants in a condition.

7.5 Intermediate State Evidence: Independent Cues

7.5.1 Intermediate State Evidence

Section 7.3 provided relative complexity evidence in its examination of the performance of people and perceptrons in the four conditions of the card-choice task that involved an interaction between two of the three cues. This evidence indicated a similar pattern of behavior across participants and across conditions and suggests that human participants, like simple networks, behave as if they are naïve Bayesians. This, in turn, suggests that perceptrons are possible models of human performance in the card-choice task.

Let us now turn to a different kind of evidence, intermediate state evidence. This evidence comes from examining information processing at some intermediate point of time. We do not have direct access to such information in the card-choice task. However, we do have the ability to observe networks at any time during learning and have started to establish a relationship between perceptron behavior and human probability learning. Next we use this relationship to help explain one puzzling aspect of human performance in the card-choice task.

7.5.2 Poor Performance With Independent Cues

The previous section examined human performance in card-choice task conditions that crossed a manipulation of the type of interaction between Cues B and C with a manipulation of the amount of reward associated with this interaction. Let us now compare performance in these conditions with the performance of participants in the condition in which all three cues in the card-choice task were independent signals of reward.

We saw earlier (Figure 7-3 and Table 7-4) that human performance in the card-choice task revealed one surprise: The fit of participants' card choices to reward probabilities was lower than expected for the independent cues condition. The relationship between card choices and reward probabilities was lower in the independent cues condition than in all of the other conditions except for High-Reward XOR. This is surprising because the simulation results that we reported in Chapters 5 and 6, and the formal analyses reported in Chapter 4, lead us to expect that the independent cue condition should be conducive to very high probability matching performance in humans.

To examine this pattern in more detail, four Welch two-sample t tests were used to compare the performance of the 40 participants in the independent cue conditions to the performance of the 40 participants in each of the other four conditions. The results of these comparisons are provided in Table 7-6. These results show that performance in the independent cue condition was significantly poorer than performance in the Low-Reward AND condition, and not significantly different from the High-Reward AND condition. With respect to the other type of interaction, performance in the independent cues condition was not significantly different from performance in either the Low-Reward or the High-Reward XOR condition. However, the difference between the independent cues condition and the High-Reward XOR condition is approaching marginal statistical significance. Thus for human participants performance in the independent cues condition is intermediate, better than High-Reward XOR and poorer than Low-Reward AND but no different from the other two conditions.

Table 7-6. The results of using Welch two-sample t

	High-Reward AND	High-Reward XOR	Low-Reward AND	Low-Reward XOR
Independent Cues	t = -1.5605	t = 1.723	t = -2.565	t = -1.2866
	DF = 76.381	*DF = 75.291*	*DF = 76.38*	*DF = 77.581*
	p = 0.1228	*p = 0.08899*	*p = 0.01228*	*p = 0.2021*

7.5.3 Is More Learning Required?

Human choice behavior in the independent cues condition does seem to provide a poorer fit to reward probabilities than we might expect. Why might this be the case?

One possibility is that the human participants in the independent cue condition may have needed more training in this environment to improve their performance to expected levels. Although the cues in this condition are mutually independent signals of reward probability, in some respects it may still be a difficult condition. Each cue signals a different probability of reward, and some of the differences between signaled probabilities are not large (see Tables 7-1 and 7-2). Perhaps participants require more experience in this environment to tease out the subtle differences in probabilities associated with different cues.

To test the plausibility of this hypothesis, we conducted a set of perceptron simulations in which 20 perceptrons were trained in each of the five conditions. Each perceptron was trained on its own training set, where each training set was created using a method that is described in more detail later in this chapter. These training sets used the same encoding of inputs and outputs that were used in the Chapters 5 and 6 simulation studies. In this new set of simulations, perceptrons were trained for only 20 epochs, and perceptron performance was examined after every 2 epochs of training. Perceptron performance was measured by taking the squared correlation of a perceptron's responses to each stimulus type with the appropriate set ideal probabilities from Table 7-2.

Figure 7-4 provides the results of this simulation study. It graphs the mean performance of perceptrons in each condition (averaging more than 20 networks), along with the standard error of this performance. This figure shows that there is rapid learning for all five conditions. However, of particular interest in this figure is the shape of the solid black line that plots the performance of the perceptrons trained in the independent cue condition. After 2 epochs of training, these perceptrons are performing surprisingly poorly given that they face no interactions between cues: They are significantly poorer than all but the High-Reward XOR condition (i.e., standard error bars do not overlap), which is a pattern similar to that revealed in the Table 7-4 summary of human performance. However, over the next 4 to 8 epochs of additional training, the performance of these independent cue networks rapidly rises to achieve levels that are much more consistent with the simulations that were discussed in Chapters 5 and 6.

Figure 7-4 provides information about the intermediate states of networks (i.e., a measure of their knowledge about an environment's probabilities of rewards) over a period of training. This evidence indicates that even though in principle the independent cue condition should lead to highly accurate probability estimates, it takes some training to achieve this accuracy. This condition is clearly very difficult for networks early in their learning, as evidence by the low level of the line for this condition at 2 epochs of training. Why might this condition be difficult to learn? One reason is that in this condition each cue signals different amounts of information about reward probability (see Table 7-1), and the signals carried by two of the cues (A and B) are similar. It is not surprising that it might take a certain amount of learning for a network to sort out these subtleties in order to generate accurate probability estimates.

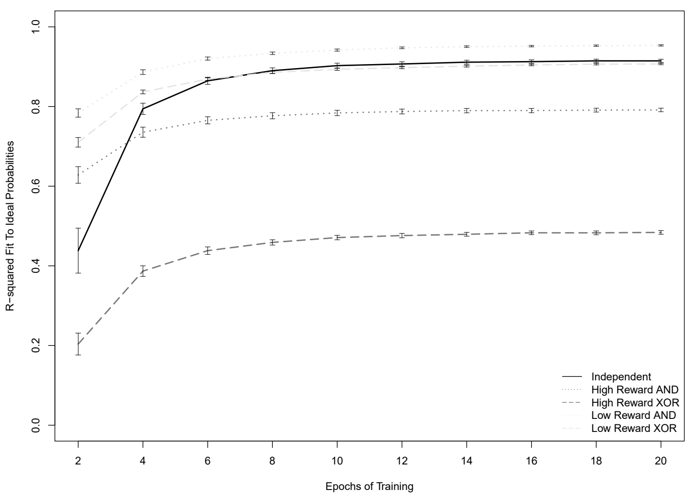


Figure 7-4. Mean probability learning performance (averaging more than 20 perceptrons) with standard errors for networks trained in the five conditions. Probability estimation is assessed as the R^2 between network responses and the probability of reward associated with each stimulus type. Performance is plotted as a function of the number of epochs of training received.

This might explain why human performance in the independent cue condition was poorer than expected. Human participants, like the networks, might need more experience with an environment to sort out the differences between signals that are being communicated by three independent cues. This suggests that if participants were offered more opportunity to explore this environment, their performance would likely improve to be more consistent with the expectations that were generated by our earlier simulation studies.

7.6 Training Perceptrons With an Alternative Input Code

7.6.1 The Need for Additional Simulations

Network simulations have played an important role in the narrative that has led to the human experiments described in this chapter. In particular, earlier chapters reported the effects of manipulating the type of interaction between cues that signal reward probability, as well as the amount of the probability signaled by an interaction. These effects have motivated the experimental design of the card-choice task described in Chapter 7. With this experimental design, we are in essence comparing these effects on human probability learning to the effects that were discovered in the computer simulations from Chapters 5 and 6.

However, some might be concerned that there are critical differences between the simulations that were reported earlier and the card-choice task described here. In particular, all of our previous work on perceptrons—both the formal work from Chapter 4 and the empirical studies from Chapters 5 and 6—use the values of 0 and 1 to represent the states of environmental cues. One reason is that this representation is very typical of perceptron research. Another reason is that it is with this encoding that an equivalence is established between perceptron structure and the coefficients of logistic regression.

However, the standard interpretation of this [0, 1] encoding is that the two codes represent the states of "absent" or "present." Indeed, this interpretation has been tacitly assumed throughout this book. However, this interpretation does not apply to the card-choice task. This is because in the card-choice task there are always three cues present on any of the eight stimulus cards. The cues differ in color; they do not differ in being present or absent. It could be argued that this difference between the card-choice task and the perceptron simulations is sufficient to invalidate comparing perceptrons to people in this chapter. This is because when a perceptron's input unit is activated with a value of 0, its connection weight is not updated by the learning rule (Dawson, 2008). When ai is 0 in Equation 1-3 or 1-6, the equation makes Δwij equal to 0. Studies of associative learning have shown that humans can learn from cues that are not present, and this has caused models to be changed to account for this finding (Ghirlanda, 2005; Van Hamme & Wasserman, 1994; Witnauer & Miller, 2011). In this monograph, we are faced with the opposite situation: The input representation for the perceptrons assumes that cues may be absent, but these cues are actually visible to human participants.

It should be stressed that as far as perceptron learning is concerned, the [0, 1] encoding for the perceptrons in Chapters 5 and 6 means that the logical nature of the problem that they learn about is identical to the structure of the card-choice task. Nonetheless, it is prudent to perform additional simulations that use an alternative encoding to eliminate this concern. In this section we report the results of simulations in which a [1, 2] encoding is employed. The interpretation of this encoding is that when a cue is rendered in one color, the input unit that represents it is activated with the value 1. When the cue is rendered in the other color, its input unit is activated with the value 2. With this representation, both cues are still represented as being in one of two states but are present to the network in either state—that is, the input unit's connection weight will be modified for either value of ai. We see that this change in encoding does not alter perceptron performance in any significant way. Furthermore, some interesting insights can be obtained from examining the structure of networks trained using this alternative input code.

7.6.2 Methodology

We now turn to describing an additional set of perceptron simulations that attempt to make a stronger analogy between perceptron training and human performance in the card-choice task. As was the case for the earlier simulations, each perceptron consisted of three input units and a single output unit that employed the logistic activation function. The primary difference between these simulations and those reported earlier is the use of [1, 2] encoding of cues as discussed in the previous section. In addition, all of these perceptrons were trained with the increasing returns operant rule that was discussed in Chapter 6. Operant training was adopted in these simulations to increase the similarity between perceptron training and human learning, because human participants in the card choice-task successively choose which cards to learn. We employed the same learning rate of 0.05 that was used in all of the previous simulations and used the same procedures as before to randomize networks prior to training. We did so to be better able to compare these new simulations to the earlier ones. In contrast to earlier simulations, these new perceptrons were trained for only 40 epochs. Pilot studies revealed that with this amount of operant training, the performance of the perceptrons in the two high reward conditions (measured by R^2 fit between perceptron responses and ideal probabilities) was similar to the probability matching of human participants in these two conditions. This amount of training was then used for the remaining conditions to permit valid comparisons between networks.

Our previous simulations (Chapters 5 and 6) trained many different networks on the same training set, under the assumption that random differences between the starting states of networks would introduce variability. However, those results demonstrated that there was a very high degree of similarity between different networks trained on the same stimuli. For these new simulations, we created a different training set for each perceptron that was studied. We were interested in seeing whether this approach would increase the within-condition variability of perceptron performance, making network behavior more similar to that of our human participants.

For these new simulation studies, we developed 40 training sets for each of the five conditions from Table 7-2. Each training set consisted of 320 stimuli. This was accomplished by duplicating each of the eight possible three-cue configurations (i.e., each row in Table 7-2) 40 times in a training set. Then, the same stochastic procedure that was used to determine rewards in the card-choice task (described in Section 7.2.5) was also used to determine whether a particular training pattern was rewarded (i.e., the perceptron was trained to turn on to the input pattern; Dawson, 2008) or not rewarded (i.e., the perceptron was trained to turn off to the pattern). That is, whether a particular stimulus was rewarded in a training set was determined by comparing a random number to the appropriate probability of reward from Table 7-2. After a training set was created, the probability of reward for each of the eight stimulus types was determined by dividing the number of times each type was rewarded by the number of times it appeared in the training set. The eight reward probabilities computed in this way were then correlated with the eight ideal probabilities from Table 7-2. For the 20 training sets created for each condition, these probabilities were very high and ranged from 0.93 to 0.99. This indicates that the stochastic procedure used to determine reward for networks (or for humans) provides reward probabilities that are strong approximations of the ideal values in Table 7-2.

Although these modifications to the simulation methodology were an attempt to make perceptrons more similar to people, it is important to recognize that there are still important differences between the two. For instance, even with 40 epochs of training perceptrons learn about many more stimuli than the mere 320 that were selected by human participants. Furthermore, all the perceptrons use the increasing returns operant rule; we cannot say whether human participants use this rule, and it is unlikely that every participant uses the same strategy to choose stimuli. However, these differences are acceptable because we are not interested in using the perceptrons as *models of* human performance. That is, our goal is not to fit perceptron performance to human behavior. Instead, we are exploring the possibility that perceptrons and people belong to the same class (naïve Bayesian mechanisms) and are using relative complexity evidence (instead of model fitting) to evaluate this possibility.

7.6.3 Perceptron Performance

As was the case in the earlier simulations, perceptron responses to the eight types of stimuli can be interpreted as estimates of the probability of reward associated with each. We are particularly interested in using these responses as relative complexity evidence: Do they vary in the same way from condition to condition as does human probability matching behavior?

To explore this possibility, for each condition and for each perceptron we correlated the eight network responses to the appropriate column of reward probabilities from Table 7-2. Table 7-7 reports the mean values of this measure of fit between perceptron probability estimates and ideal reward probabilities. The pattern of results in Table 7-7 is very similar to the pattern of results observed in previous simulations: Perceptron estimates are lower for XOR interactions, higher for low reward conditions, and very high for the independent cue condition. All of these results are consistent with the formal analyses of Chapter 4. The pattern of results in Table 7-7 is also similar to the human probability matching behavior summarized in Table 7-4. However, these simulations (like the earlier ones) produce much better performance for the independent cues conditions than is observed for human participants.

To compare the performance of these networks to human performance, we conducted analogous sets of statistical analyses of perceptron performance. In parallel to Section 7.4.2, we examined perceptron performance by conducting an ANOVA on the *R*2 values for networks in the four conditions involving an interaction between Cues B and C. This analysis revealed a significant main effect of type, *F*(1, 156) = 95.31, *p* = 2.0e-16, η2 = 0.135; a significant main effect of reward, *F*(1, 156) = 409.32, *p* =

Table 7-7. The mean squared correlation between individual network responses to each stimulus and the ideal reward probabilities.

	High-Reward AND	High-Reward XOR	Independent	Low-Reward AND	Low-Reward XOR
Mean	0.585	0.337	0.572	0.786	0.742
SE	0.022	0.013	0.029	0.011	0.010

Note. Each mean is based on 40 perceptrons. The SE row provides the standard errors of these means.

2.0e-16, η2 = 0.579; and a significant interaction between type and reward, *F*(1, 156) = 46.68, *p* = 1.77e-10, η2 = 0.066. Post hoc tests conducted using the Tukey HSD statistic demonstrated that participants exposed to the AND interaction learned to be more accurate probability estimators than did participants exposed to the XOR interaction (*p* = .009). As well, participants who experienced a low-reward condition performed better than did participants exposed to a high-reward condition (*p* = 1.0e-16). Finally, the significant interaction emerged because the difference between Low-Reward XOR and High-Reward XOR networks (*M* difference = 0.404) was significantly larger than the difference between Low-Reward AND and High-Reward AND networks (*M* difference = 0.200).

This pattern of results is very similar to the pattern observed in previous simulations (Chapters 5 and 6), indicating that changing input unit encoding does not change the effects of interaction type and probability of reward on perceptron behavior.

These results are also identical to those provided by the analysis of human probability matching behavior, with the exception that human participants did not exhibit a significant interaction between these two factors. However, the differences between mean R^2 in Table 7-4 are consistent with this interaction. The difference between the Low-Reward XOR condition and the High-Reward XOR condition is 0.186, whereas the difference between the Low-Reward AND condition and the High-Reward AND condition is only 0.054. One reason that the interaction is significant for networks, but not for humans, is that within-condition variability between networks is much lower than within-condition variability between humans (see Figure 7-3).

In parallel with Section 7.5.2, we also compared perceptron performance in the independent cues condition to perceptron performance in each of the other four conditions using Welch two-sample *t* tests. The results of these comparisons are provided in Table 7-8.

These results indicate that perceptrons in the independent cues condition generated responses that were significantly better fits to reward probabilities than was the case for perceptrons in the High-Reward XOR condition, and significantly poorer than was the case for perceptrons in either of the two low-reward conditions. For these perceptrons, independent cues performance was not significantly different from performance in the High-Reward AND condition. This pattern of results is consistent with the differences observed in human performance (Table 7-6) but is sharper. The key differences between the two tables is a stronger difference between the independent cues condition and both the High-Reward and Low-Reward XOR conditions for perceptrons than for humans. The strong performance for these networks in the Low-Reward XOR condition is reminiscent of network performance that was observed in Chapters 5 and 6.

Table 7-8. The results of using Welch two-sample t

	High-Reward AND	High-Reward XOR	Low-Reward AND	Low-Reward XOR
Independent Cues	*t* = -0.37793	*t* = 7.3053	*t*= -6.9102	*t* = -5.509
	df = 72.915	*df* = 54.815	*df* = 49.158	*df* = 48.785
	p = .7066	*p* = 1.214e-09	*p* = 8.997e-09	*p* = 1.338e-06

7.6.4 Perceptron Structure and Input Encoding

Before continuing with our examination of human performance in the card-choice task, let us briefly consider the effect of different input encodings (i.e., [1, 2] vs. [0, 1]) on the structure of perceptrons. This is because concerns about learning about absent cues have influenced proposals for models of associative learning (Ghirlanda, 2005; Van Hamme & Wasserman, 1994; Witnauer & Miller, 2011).

Let us start with the results of a simulation study that compares performance using the two encodings. In one condition, we repeated the simulation methodology described in 7.6.2, training 40 networks in each condition. In this case, however, each network was trained for 2,500 epochs to ensure that network performance was near a dynamic equilibrium of the sort discussed earlier in Section 2.6. This condition used exactly the same training sets for as were used in the earlier Section 7.6 simulations, so these simulations used the [1, 2] encoding and can illustrate changes in network performance (i.e., Table 7-7) when training is extended from 40 epochs to 2,500. In a second condition, 40 new training sets were created for each condition using the procedure described in Section 7.6.2, but these training sets used the [0, 1] encoding. What is the effect of the different encoding on the networks trained in the various conditions?

Table 7-9 summarizes the key results of this simulation study. For both encodings, this table reports the average network structure (the average bias and the average of each of the three weights), where each average is the mean of a network component taken over 40 perceptrons. The table also reports the average fit of networks to reward probabilities for each encoding in each of the five conditions, where this fit is the squared correlation between network responses and Table 7-2 probabilities.

Table 7-9. The comparison between components of networks trained on different input encodings ([1,2] vs [0,1]) in each of the five conditions of the analog card-choice task.

	High-Reward AND		High-Reward XOR		Independent		Low-Reward AND		Low-Reward XOR	
Source	[1, 2]	[0, 1]	[1, 2]	[0, 1]	[1, 2]	[0, 1]	[1, 2]	[0, 1]	[1, 2]	[0, 1]
θ	–	–	–	–	–	–	–	–	–	–
w	1.75	1.97	1.48	1.46	0.43	0.41	3.11	3.29	2.65	2.63
w	1.03	1.09	–	0.07	0.59	0.46	0.23	0.22	0.04	–
w	0.97	1.06	0.02	–	1.62	1.56	0.18	0.16	0.01	0.00
R	0.774	0.773	0.445	0.424	0.880	0.880	0.946	0.954	0.900	0.890
SE	0.007	0.009	0.012	0.016	0.010	0.010	0.004	0.002	0.003	0.006

Note. All networks were trained for 2,500 epochs using the increasing returns operant method. Each entry is the mean of a network component (bias or weight) taken over 40 networks. The bottom two rows provide the average R^2 fit, with standard error, of the network responses in each condition to the appropriate column of reward probabilities provided in Table 7-2.

Three key regularities are apparent in Table 7-9. First, for each condition there is no significant difference between the performances of networks trained with one encoding in comparison to networks trained with the other encoding. This is to be expected given that, as noted in Section 7.6.1, the change in encoding does not change the logical nature of the problem being presented to a perceptron.

Second, for each condition the three average network weights obtained using one encoding are nearly identical to the three average network weights obtained using the other encoding. Differences between weights are not significant; variations between weights in each stimulus condition are due to the random variations intrinsic to network training (e.g., random variations in initial states, random variations in operant choices). Again, this is not surprising. We proved in Chapter 4 that a perceptron's connection weight for a probabilistic task is a particular measure of a cue's effect on reward probability (i.e., the natural logarithm of the odds ratio). As in each stimulus condition all networks are trained on logically equivalent problems (the only difference is input encoding), regardless of encoding the relationship between a cue and reward probability is the same, so we expect the same connection weights (the same odds ratios) for networks trained in the same condition using different encodings.

Third, for each stimulus condition there appears to be a marked difference between the bias of networks trained with one encoding and the bias of networks trained with the other. For these simulations, the bias of networks trained using the [1, 2] encoding is approximately double the bias of networks trained using the [0, 1] encoding. A straightforward computational analysis shows that there is a very systematic relationship between bias values and input encoding.

The response of a network is the conversion of a net input value into activity using the logistic activation function, as was introduced in Chapter 1 in Equation 1-4. The net input for the output unit is the sum of weighted input signals plus the network's bias. Equation 7-1 provides a net input equation for a perceptron that uses a [0, 1] encoding. In this equation, the subscript [0, 1] is used to identify the network's input encoding to distinguish it from the other network. A, B, and C are all input signals that are either equal to 0 or 1.

$$net_{[0,1]} = \theta_{[0,1]} + w_{a[0,1]}A + w_{b[0,1]}B + w_{c[0,1]}C \tag{7-1}$$

Now consider the net input for a perceptron that is given the same input pattern as is represented in Equation 7-1, but with the [1, 2] encoding. The expression for this net input is provided in Equation 7-2. Note that the components of this equation are differentiated from those of Equation 7-1 with the use of the [1, 2] subscript. Note, too, that the input pattern in this is equation is represented as $(A + 1)$, $(B + 1)$, and $(C + 1)$. This is because if the same stimulus is being used to determine net input in Equation 7-1 and 7-2, then one converts the [0, 1] encoding for Equation 7-1 into the [1, 2] encoding for Equation 7-2 by adding 1 to each Equation 7-1 input value. The remainder of Equation 7-2 is produced by multiplying the weights through the equation and rearranging the resulting equation components.

$$\begin{aligned} & net_{[1,2]} \\ & = \theta_{[1,2]} + w_{a[1,2]}(A+1) + w_{b[1,2]}(B+1) + w_{c[1,2]}(C+1) \\ & = \theta_{[1,2]} + w_{a[1,2]}A + w_{a[1,2]} + w_{b[1,2]}B + w_{b[1,2]} + \\ & w_{c[1,2]}C + w_{c[1,2]} \\ & = (\theta_{[1,2]} + w_{a[1,2]} + w_{b[1,2]} w_{c[1,2]}C) + w_{a[1,2]}A + w_{b[1,2]}B + \\ & + w_{c[1,2]} \end{aligned} \tag{7-2}$$

Let us now make some simplifying assumptions using our logical understanding of the problem being learned by perceptrons under either encoding, assumptions that are also supported by the simulation results presented in Table 7-9. First, given that the logical structure of the problem is not affected by encoding—the mapping from stimuli to responses remains the same—when perceptrons trained with different encodings reach dynamic equilibrium they must generate the same response to the same stimulus. This is only possible if the net input produced by this stimulus under one encoding is equal to the net input produced by this stimulus under the other encoding. This assumption is provided in Equation 7-3 and leads to the conclusion that Equation 7-1 and 7-2 must generate the same values when presented the same stimulus.

$$net_{[0,1]} = net_{[1,2]} \tag{7-3}$$

Next, given that the logical structure of the task is not affected by encoding, the relationship between each cue and each reward probability must be the same regardless of encoding. In other words, the odds ratios achieved by a network faced with one encoding must be the same as those achieved by a network faced with the other encoding when dynamic equilibrium is achieved. This means that the weight associated with a cue in one network must be the same as the weight associated with the same cue in the other network. The relationships between network weights are provided in Equation 7-4, which also uses this relationship to simplify the notation for each weight by removing the subscript denoting encoding.

$$\begin{gathered} w_{a[0,1]} = w_{a[1,2]} = w_a \\ w_{b[0,1]} = w_{b[1,2]} = w_b \\ w_{c[0,1]} = w_{c[1,2]} = w_c \end{gathered} \tag{7-4}$$

The assumptions made explicit in Equations 7-3 and 7-4 permit us to equate the net input expression in Equation 7-1 to the net input expression in Equation 7-2. This relationship is made explicit in Equation 7-5, which also simplifies matters by removing the subscript denoting encoding where possible. Important to note, this subscript cannot be removed from either bias term, because we have no prior knowledge about the relationship between these two terms.

$$\begin{aligned} &\theta_{[0,1]} + w_a A + w_b B + w_c C \\ &= (\theta_{[1,2]} + w_a + w_b + w_c) + w_a A + w_b B + + w_c C \end{aligned} \tag{7-5}$$

However, we can derive an equation that converts the bias for one encoding into the bias for the other encoding using Equation 7-5. Equation 7-6 presents this expression. It is derived by subtracting the three terms that appear on both sides of Equation 7-5:

$$\theta_{[0,1]} = \theta_{[1,2]} + w_a + w_b + w_c \tag{7-6}$$

With Equation 7-6 in hand, let us return to the simulation results reported in Table 7-9. Equation 7-6 makes the prediction that if we sum the bias and the weights obtained from networks trained using the [1, 2] encoding, the result should be the bias for networks trained in the same condition using the [0, 1] encoding. Table 7-10 confirms this prediction, taking into account random variations between network performances in the different conditions.

Table 7-10. The use of the Table 7-9 simulation results to test the prediction made by Equation 7-6.

	High-Reward AND	High-Reward XOR	Independent Cues	Low-Reward AND	Low-Reward XOR
θ[0,1]	*−2.98*	*−1.66*	*−1.99*	*−2.98*	*−2.35*
θ[1, 2] + wA + wB+ wC	*−2.64*	*−1.67*	*−2.14*	*−2.89*	*−2.44*

Note. The first row provides the average bias of networks trained with the [0, 1] encoding, each of which is taken from Table 7-9. The second row presents the sum of the average bias and the average weights of networks trained with the [1, 2] encoding, also taken from Table 7-9.

In summary, the formal and empirical analyses presented in this section indicate that the choice of encoding used to represent inputs to networks will have only one effect if the logical structure of a problem—the mapping between stimuli and responses—is maintained. This effect is to change the bias of the output unit. In essence, the learning process adjusts the bias to take into effect the magnitude of the signals used to encode the state of the codes. One important implication of this result is that the formal and empirical results obtained in previous chapters still apply to the results of Chapter 7, because the logical nature of the card-choice task is not changed by encoding cue states as being absent versus present, or instead as being one color versus another.

7.7 Fitting Logistic Equations to Human Choices

7.7.1 Rationale

The purpose of the current chapter is to determine whether perceptrons are potential models of human performance in a probabilistic discrimination task. Previous sections in this chapter have presented analyses of human performance in the card-choice task and have compared this performance to that of perceptrons. These results suggest that human participants, like perceptrons, behave in this task as if they are naïve Bayesians. This is because human choice behavior is affected in a similar fashion to perceptron responses when interactions between cues signal reward probabilities. In particular, human performance is poorer when an XOR interaction between cues is present than when an AND interaction between cues is present. In addition, human performance is poorer when an interaction signals a higher probability of reward in comparison to when the interaction signals a lower probability of reward. These two results are consistent with the hypothesis that human participants have similar difficulties recognizing the signals of cue interactions as we have seen in simple artificial neural networks.

One consequence of the observed similarities between human and network performance is that we should be able to model human responses using an equation like the one that defines perceptron behavior. In this section we explore this possibility by employing a modeling approach that is more typical of experimental psychology—fitting a mathematical model to empirical data (Dawson, 2004). In particular, we use logistic regression to model the choice behavior of participants in each condition of the card-choice task.

7.7.2 Methodology

The goal was to discover the logistic equation that generated the best overall fit to the eight choice probabilities produced by the 40 participants in each condition in the card-choice task. This was accomplished by summarizing the choice behavior for each condition in a format that could permit logistic regression to be used to fit the data. This involved summing the card choice frequencies made by the 40 human participants in each condition, separating the choices that led to a reward from those that did not. Table 7-11 presents the summarized frequency data that was used for this analysis.

The *glm* function in the R statistical programming language was used to predict reward frequency from the states of the three cues using the binomial (logit) model. This was performed for each of the five columns of frequencies in Table 7-11. This analysis produces a logistic equation for each condition that predicts choice probability from the cue states of the eight stimuli.

7.7.3 Results

The results of these logistic regressions are presented in Table 7-12. Each column in this table provides the constant of the regression equation as well as the coefficient associated with each cue. These regression equation components are labeled to highlight their relationship between logistic regression and perceptron structure. Table 7-12 also presents the fit between the equation and human choice behavior. This fit was determined by generating the predicted probability of choice for each stimulus using a regression equation. The squared correlation between these eight predictions and the average number of times each of the eight cards was selected by the 40 participants in a condition (i.e., the appropriate column of Table 7-3) was then determined. The second row from the bottom of Table 7-12 presents this measure of the equation's fit to the human choices in each condition. The bottom row of the table presents the squared correlation between the probability estimates generated by the regression equation and the ideal reward probabilities from Table 7-2.

The results presented in Table 7-12 indicate that the logistic regression equations capture a great deal of the variance in human choice behavior. The worst fit is for the High-Reward XOR condition, but the equation still accounts for 46% of the variance in the human data. In all other conditions, the logistic equation accounts for 86% of the variance or higher. Of interest, the best fit is for the independent cues condition.

Although the logistic equations are being fit to human choice behavior, these choices are also being made in an environment governed by particular rules that convert cues into reward probabilities. How related are these models of choices to the ideal reward probabilities? The bottom row of Table 7-12 presents the relevant measures of fit, and the numbers in this row follow a similar pattern to that which was seen earlier for both human behavior and for network responses. In particular, these fit values seem very similar to those reported for the two sets of networks trained to dynamic equilibrium (Table 7-9).

In fact, there is a striking similarity between the logistic regression coefficients and the structure of these networks. Table 7-13 presents the regression coefficients from Table 7-12 along with the average network structure from the [0, 1] encoding previously seen in Table 7-9. This permits pairs of structures to be easily compared. It is apparent from Table 7-13 that there is a striking similarity between the regression equations and the network structures; the bottom row of the table provides the squared correlation between each pair of columns. This result is particularly interesting because the two structures arise from very different sources—one from being fit to human choices, the other from networks that are trained on an analogous task.

7.8 Exploring Human Choice Strategy

7.8.1 Choice Strategy

To this point, the examination of participant behavior in the card-choice task has focused on the relationship between the frequency that various cards were selected and the probability of reward associated with these cards. In this section, we briefly consider possible strategies that participants may have employed when deciding which card to choose next when they were participating in the first phase of the study. From one perspective, this is an exploration of another kind of intermediate state evidence, where the intermediate state is a transition from the current card to the next card.

One famous study of how human participants learned artificial concepts (Bruner, Goodnow, & Austin, 1956) emphasized the importance of strategy. In the operant version of this task, participants viewed a set of 81 cards. Each card depicted a unique pattern of four features; each feature could take on three values. Participants were given the task of learning a rule (e.g., "three red circles") that defined a concept. This was accomplished by participants choosing one of the cards and being informed whether the chosen card belonged to the target category. In this study, Bruner et al. were interested in the processes used by participants to choose the next card. They called these processes *strategies*: "A strategy refers to a pattern of decisions in the acquisition, retention, and utilization of information that serves to meet certain objectives, i.e., to insure certain forms of outcome and to insure against others" (Bruner et al., 1956, p. 54). It was discovered that changing task conditions, for instance to increase or decrease cognitive load, would alter the strategies adopted by participants.

The operant nature of the card-choice task permits participant strategy to come into play. This is because participants are free to choose any of the eight cards as they learn about the probability structure of the environment. The computer program that runs the card-choice task permits strategy to be examined because it records each card choice made by a participant, where the card was located, and whether the card was rewarded.

Many questions about strategy could be explored using this record of participant choices. The current section considers only one. To this point in Chapter 7, we have been considering human participants as probability matchers who learn about the probability of reward associated with each card. However, some might argue that these participants only *seem* to be probability matchers. Some have suggested that the behavior that we have been describing earlier in this chapter is not actually probability matching, but is instead an artifact of a simple strategy. Let us call this strategy "randomly choose but stay when rewarded." In this strategy, whenever a participant chooses a card, this choice is random. If the selected card is

Table 7-11. Total card choice frequencies for the 40 participants in each condition. Each frequency is associated with a reward state (1 = reward, 0 = no reward) and a [0, 1] encoding of the state of each cue on the chosen card.

Reward	Cue A	Cue B	Cue C	High-Reward AND	High-Reward XOR	Independent Cues	Low-Reward AND	Low-Reward XOR
1	0	0	0	0	0	0	0	0
1	0	0	1	0	503	560	0	225
1	0	1	0	0	631	229	0	229
1	0	1	1	823	0	1032	249	0
1	1	0	0	747	636	158	1056	923
1	1	0	1	874	1238	984	1269	1430
1	1	1	0	888	1344	501	1320	1387
1	1	1	1	2312	557	1490	2029	1120
0	0	0	0	789	987	929	718	807
0	0	0	1	1000	993	826	865	976
0	0	1	0	892	1043	934	812	953
0	0	1	1	928	1200	970	1138	1189
0	1	0	0	760	958	971	682	866
0	1	0	1	952	830	1040	853	927
0	1	1	0	968	901	1092	853	842
0	1	1	1	867	979	1084	956	926

Table 7-12. The structure of equations that result from using logistic regression to fit the frequency data from Table 7-11.

Logistic Regression Coefficient	High-Reward AND	High-Reward XOR	Independent Cues	Low-Reward AND	Low-Reward XOR
θ	-2.78	-1.23	-2.17	-3.07	-1.21
w_A	1.97	1.34	0.46	3.22	1.50
w_B	0.96	-0.02	0.65	0.35	0.04
w_C	0.94	-0.15	1.57	0.31	-0.01
R	0.890	0.461	0.929	0.859	0.876
R	0.811	0.498	0.941	0.967	0.921
R2 Fit To Average Human Choices	**0.890**	**0.461**	**0.929**	**0.859**	**0.876**
R2 Fit To Ideal Reward Probabilities	**0.811**	**0.498**	**0.941**	**0.967**	**0.921**

Note. The bottom row presents the R^2 fit between choice probabilities generated from each equation and the average choice behavior reported earlier in Table 7-3.

not rewarded, then participants randomly choose another card (which may include the card that was not rewarded with the previous choice). However, if the selected card is rewarded, then the participant chooses it again. In short, according to this strategy participants always continue to choose a card when it rewards and randomly choose a card when a reward is not received.

Why might the "randomly choose but stay when rewarded" strategy generate probability matching behavior without participants learning reward probabilities? The answer is that with this strategy, participants are simply more likely to choose cards associated with higher reward probabilities because they are more likely to be rewarded by these cards and are therefore more likely to continue to choose them. With this strategy, probability matching might emerge from the structure of the environment, instead of from what participants have learned from this environment.

This section explores this possibility. It does so by using the methodology of Bruner et al. (1956), who compared predictions generated by assuming that an ideal participant adopted a particular strategy to the behavior produced by actual participants. To preview our results, we do discover that the "randomly choose but stay when rewarded" strategy can predict a large amount of variance in our summary of human choices. However, significantly more variance is accounted for by exploiting a strategy that assumes that human participants are actually learning about the probability structure of the environment and are using this knowledge to guide their choice of the next card.

7.8.2 Summarizing Human Choices

Participant choices were summarized as follows: The first card selected by a participant is the starting point of the data summary. It is designated as being the "current card." After selecting the current card, the participant then selects a stimulus designated as being the "next card." After these two choices have been made, a frequency matrix that describes the participant's choice behavior can be updated. This is accomplished by adding 1 to the cell in the matrix with the row that corresponds to the current card and with the column that corresponds to the next card. For example,

Table 7-13. The comparison between logistic regression equation coefficients (from Table 7-12) and the structure of networks trained to equilibrium on the [0, 1] encoding of the analog to the card-choice task (from Table 7-9).

	High-Reward AND		High-Reward XOR		Independent Cues		Low-Reward AND		Low-Reward XOR	
Source	Logistic	Network	Logistic	Network	Logistic	Network	Logistic	Network	Logistic	Network
θ	–	–	–	–	–	–	–	–	–	–
w_A	1.97	1.97	1.34	1.46	0.46	0.41	3.22	3.29	1.50	2.63
w_B	0.96	1.09	–	0.07	0.65	0.46	0.35	0.22	0.04	–
w_C	0.94	1.06	–	–	1.57	1.56	0.31	0.16	–	0.00
R^2	**0.998**		**0.985**		**0.995**		**0.998**		**0.999**	

Note. The bottom row presents the R2 fit between choice probabilities generated from each equation and the average choice behavior reported earlier in Table 7-3.

if Card 5 is the current card and Card 3 is the next card, the cell (5,3) in the participant's data matrix is incremented by one. For the next choice, the "next card" from the previous step becomes the "current card," the new choice becomes the "next card" for the current step, and the matrix is updated accordingly. This process is continued until all 319 choices made by a participant have been summarized.

For the current section, our interest is in the sequences of card choices made by all of the participants in a condition, and not in the strategic behavior of individual participants. For this reason, we summed the 40 summary matrices for the participants in each condition. Table 7-14 provides one of these matrices, the one created for the 40 participants in the independent cues condition. Similar matrices were produced for the other four conditions of the card-choice task.

Table 7-14 provides the overall summary of participant choices—frequencies of transitions from the current card to the next card. Other summary matrices can also be constructed for the same experimental condition. For instance, Table 7-15 provides a summary matrix for the independent cues condition but only summarizes choices made after the current card has been rewarded.

A matrix that is complementary to Table 7-15 that only summarizes card transitions when the current card is not rewarded can also be created. Table 7-16 provides this matrix for the independent cues condition. Note that this table can be created by subtracting Table 7-15 from Table 7-14.

One approach to studying participants' choice strategy is to compute some simple statistics that summarize key properties of matrices like Tables 7-14, 7-15, and 7-16. For instance, if one sums the diagonal of one of these matrices and then divides this sum by the sum of the entire matrix the result is the overall probability of staying with the current card. The first three rows of Table 7-17 provide this probability for the overall summary matrix, the rewarded summary matrix, and the not rewarded summary matrix for each of the five conditions. These values indicate that the overall probability of staying with the current card is about 0.29, although it is slightly lower for the two XOR conditions. The probability of staying tends to be higher (about 0.38) when the current card is rewarded in comparison to the probability of staying when the current card is not rewarded (about 0.22). On one hand, this seems to be consistent with the "randomly choose but stay when rewarded" strategy, because across conditions participants are more likely to stick with the current card when it rewards. On the other hand, the probability of staying with the current card seems much lower than one might expect if this strategy was being employed. This is the first suggestion that the "randomly choose but stay when rewarded" strategy does not provide the best account of how the participants behave in the card-choice task.

Other simple summary statistics can be used to explore the utility of explaining participant choices via the "randomly choose but stay when rewarded" strategy. For instance, according to this strategy staying with the current card should be strongly related to the card's probability of reward. This can be assessed by relating the eight diagonal entries of a choice summary matrix with the eight ideal reward probabilities associated with the stimuli (i.e., the appropriate column of Table 7-2). The second-to-last row of Table 7-17 presents the squared correlations between these two variables for each of the conditions of the card-choice task and does show a strong relationship: across conditions, the average R^2 between the diagonal entries of the overall choice matrix and the set of reward probabilities is 0.722.

However, a similar analysis again suggests that participant choices are informed by learning reward probabilities and are not simply made using the "randomly choose by stay when rewarded" strategy. The sum of each column of a choice matrix indicates the number of times the stimulus associated with that column is selected overall. If stimulus choices are based on (learned) probabilities of reward, which is inconsistent with the "randomly choose but stay when rewarded" strategy, then these column sums should be strongly related to reward probability, because the column sums should provide evidence of probability matching. The final row of Table 7-17 indicates that this is indeed the case. It presents the R^2 between column sums and reward probabilities for each of the overall choice matrices. The average R^2 in this row is 0.864, indicating that these relationships are stronger than those between diagonal entries and reward probabilities. The diagonal entry relationship is stronger than the column sum relationship for only one condition, Low-Reward XOR.

7.8.3 Modeling Human Choices

The preceding discussion suggests that a simple heuristic does not provide the best account of human choices of cards—how participants choose the next stimulus after being rewarded (or not) for their current choice. A more appropriate test of a claim like this is to use a heuristic or strategy to simulate a set of stimulus choices and then to compare this simulated data to the choices generated by human participants (Bruner et al., 1956). In this section, we perform this test to examine the relationship of five strategies to human choice data for each of the five conditions in the card-choice task.

We conducted five simulation studies. In each study, 20 experiments were simulated. Each of these experiments used a strategy to simulate the choices of 40 participants, and therefore generated a data matrix that could be compared to human participant data such as Table 7-14. Each experiment started with a random choice of one of the eight stimulus cards. It then used a heuristic to generate the next 319 stimulus choices made by a single participant. This process was then repeated until the choices of 40 participants had been simulated. This method was used to simulate stimulus choices for each of the five conditions in the card-choice task. One condition was differentiated from another by the probability that a chosen stimulus was rewarded: The reward probabilities for a particular condition were always the ideal probabilities (i.e., the appropriate column of Table 7-2).

Once created, the data for a single simulation (i.e., one simulation of 40 participants in one condition of the card-choice task) were compared to the actual choice data obtained from human participants (i.e., to data like Table 7-14 but for the appropriate condition). This comparison was conducted by measuring the Euclidean distance between the simulated matrix and the participant matrix. This distance is simply the square root of the sums of the squared differences between the 64 corresponding cells in each of the two matrices. Higher distances between matrices indicate higher dissimilarity between simulated and actual choice data.

Three of the five strategies used to simulate data included a Stay rule: If the current stimulus was rewarded, then this stimulus was selected as the next card. One of these strategies (Random Choice) involved randomly choosing the next stimulus if the current stimulus was not rewarded. (This random choice could include choosing the current stimulus again.) A second strategy (Ideal Choice) involved choosing the next stimulus using the ideal probabilities of reward if the current stimulus was not rewarded. That is, the likelihood that the simulation chose a particular card was equal to the probability that the card is rewarded. A third strategy (Human Choice) was the same as the previous, but the probability of choosing a card when the current card was not rewarded was the probability that human participants actually selected the card in the appropriate condition of the card-choice task. This strategy was studied under the assumption (supported by results reported earlier in this chapter) that human participants did not learn the ideal reward probabilities in some of the task's conditions.

The remaining two strategies did not use a Stay rule but instead employed a Do Not Stay rule. For both strategies, the choice of the next card was not affected by whether the current card was rewarded. Instead, these Do Not Stay strategies always selected the next card using a probabilistic rule. One always chose the next card according to its ideal reward probability. The other always chose the next card according to the choice probability for the card observed for human participants.

Figure 7-5 presents the results of comparing the data generated by each of these different strategies to the appropriate choice data obtained from human participants. Each bar in Figure 7-5 presents the average Euclidean distance between simulated data and human choices, where this average is taken over the 20 simulations for a

Table 7-14. The overall summary of choices made by the 40 participants who experienced the independent cues condition of the card-choice task.

		Next Card								
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Sum**
Current Card	**1**	238	90	85	93	95	81	94	153	**929**
	2	124	397	123	128	122	143	181	167	**1385**
	3	79	143	285	123	131	164	98	135	**1158**
	4	91	134	132	624	146	303	220	346	**1996**
	5	92	145	111	144	268	125	108	133	**1126**
	6	90	161	163	309	111	606	219	362	**2021**
	7	75	162	120	229	114	238	397	253	**1588**
	8	133	152	136	349	137	357	275	1018	**2557**
	Sum	**922**	**1384**	**1155**	**1999**	**1124**	**2017**	**1592**	**2567**	**12760**

Note. The rows of the matrix correspond to the "current card" and the columns correspond to the "next card." Each cell entry represents the number of times that a participant chose a particular "next card" after choosing a particular "current card." The diagonal entries therefore indicate the frequency with which participants decided to choose the "current card" again.

Table 7-15. The overall summary of choices made by the 40 participants who experienced the independent cues condition of the card-choice task.

		Next Card								
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Sum**
Current Card	**1**	0	0	0	0	0	0	0	0	**0**
	2	44	205	51	45	43	52	65	54	**559**
	3	14	26	67	23	21	35	17	26	**229**
	4	39	62	65	360	77	139	113	173	**1028**
	5	10	20	12	19	60	15	7	15	**158**
	6	37	75	76	127	47	363	99	158	**982**
	7	18	56	34	52	34	67	174	66	**501**
	8	62	74	76	197	73	186	152	662	**1482**
	Sum	**224**	**518**	**381**	**823**	**355**	**857**	**627**	**1154**	**4939**

Note. The rows of the matrix correspond to the "current card" and the columns correspond to the "next card." Each cell entry represents the number of times that a participant chose a particular "next card" after choosing a particular "current card." The diagonal entries therefore indicate the frequency with which participants decided to choose the "current card" again.

Table 7-16. The overall summary of choices made by the 40 participants who experienced the independent cues condition of the card-choice task.

		Next Card								
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Sum**
Current Card	**1**	238	90	85	93	95	81	94	153	929
	2	80	192	72	83	79	91	116	113	826
	3	65	117	218	100	110	129	81	109	929
	4	52	72	67	264	69	164	107	173	968
	5	82	125	99	125	208	110	101	118	968
	6	53	86	87	182	64	243	120	204	1039
	7	57	106	86	177	80	171	223	187	1087
	8	71	78	60	152	64	171	123	356	1075
	Sum	**698**	**866**	**774**	**1176**	**769**	**1160**	**965**	**1413**	**7821**

Note. The rows of the matrix correspond to the "current card" and the columns correspond to the "next card." Each cell entry represents the number of times that a participant chose a particular "next card" after choosing a particular "current card." The diagonal entries therefore indicate the frequency with which participants decided to choose the "current card" again.

particular condition. The standard error of each mean is also provided on each bar in Figure 7-5.

The motivation for these simulations of choice behavior was to explore the possibility (introduced in Section 7.8.2) that human participants were not actually learning reward probabilities but were instead employing a simple heuristic to choose successive stimuli. According to this possibility, probability matching is simply a consequence of the strategy. However, Figure 7-5 indicates that the best accounts of human choice data are not those that assume a heuristic in which participants stay with a stimulus when it is rewarded.

The standard error bars in Figure 7-5 are sufficiently short to indicate that within-condition differences in bar height represent significant differences between strategies with respect to the fit to data. In each condition, the two strategies that employ a Do Not Stay heuristic (the two rightmost bars for each condition) are significantly shorter than the other three strategies that employ a Stay heuristic. This suggests that staying with a rewarded stimulus does not provide a more accurate account of human choices than does an account that presumes that participants are using reward probabilities to choose successive stimuli.

Why do the Stay strategies provide poorer fits to human choice data? The primary reason is that each of these strategies inflates the diagonals of the choice matrix and correspondingly deflates the off-diagonal values. As noted in the discussion of Section 7.8.2, the lower diagonal values

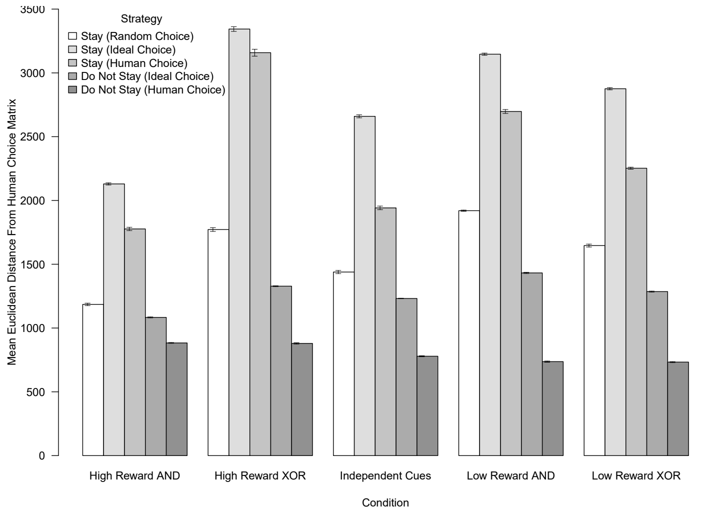


Figure 7-5. The average Euclidean distances between simulated choice data and observed choice data for each of the five conditions of the card-choice task. Note. Five strategies were used to simulate data for each condition.

of human choice matrices provides evidence that human participants are much less likely to stay with a rewarded card than any of the three Stay strategies of Figure 7-5.

7.9 Error Evidence: Analysis of Human Card Preferences

7.9.1 Error Evidence

The previous sections have examined the relative complexity evidence and the intermediate state evidence provided by the first phase of the study, the phase in which participants choose cards to learn about the reward structure of the card-choice task. Error evidence is an additional source of information that can be used to investigate the relationship between the behavior of humans and models. In this section, we consider a particular form of error evidence made available by the card preference judgments obtained in the second phase of the experiment. In particular, we can compare human preferences for each card to the preferences derived from ideal reward probabilities. In general, we are interested in using this comparison to answer a basic question: At the end of exploring the environment, how much did participants learn about its reward probabilities?

Another reason for exploring the human preferences that were obtained in the second phase of the card-choice task was revealed by the Chapter 6 simulations that used operant perceptrons. Those simulations demonstrated that very different choice behaviors—increasing returns versus gambler's fallacy—could lead to identical estimates of reward probabilities. This was discovered by examining the connection weights of the operant perceptrons at the end of training. However, if information about perceptron structure was not available, and if we could rely on only the choice behavior of networks, then we might infer that very different patterns of choice reflect very different models of reward probabilities. This inference would be mistaken. The examination of human card preferences provides a different measure of probability learning from the choice behavior that has been examined up to this point.

7.9.2 Row Sums of Human Preference Matrices

We began by preprocessing the card preference data using the same procedure described in Section 7.3.1. This relabeling assigned the same name to the different cards that provided the same function (i.e., the same reward probability) to different participants in the same condition. Next, for each participant we created an 8×8 preference matrix. Each row of this matrix was associated with one of the eight cards, as was each column. A particular entry in the matrix indicated the number of times that the card represented by the matrix's row was preferred over the card represented by the matrix's column. For each participant, the sum of the entries in the matrix was 112, as each pair of cards was judged four times, but a card was never compared to itself.

Table 7-17. Some summary statistics for the choice summary matrices (e.g., Table 7-14) created for each of the conditions in the card-choice task.

Logistic Regression	**High-Reward AND**	**High-Reward XOR**	**Independent Cues**	**Low-Reward AND**	**Low-Reward XOR**
Probability of staying with the current card (overall)	0.300	0.273	0.300	0.300	0.267
Probability of staying with the current card when rewarded	0.375	0.370	0.375	0.375	0.392
Probability of staying with the current card when not rewarded	0.241	0.212	0.241	0.241	0.178
R	0.620	0.809	0.620	0.620	0.941
R	0.832	0.906	0.832	0.832	0.920

Table 7-18 provides the sum of each of the preference matrices for each of the 40 participants in the independent cues condition. It represents 4,480 preference judgments. It shows, for instance, that Card 1—with the cue pattern (0,0,0)—was preferred over Card 2, with the cue pattern (0,0,1), six times in the preference phase of the study. In contrast, Card 2 was preferred over Card 1 154 times.

Table 7-18 also provides the sum of the entries in each row. These sums give a general sense of how preferred a particular card was in general in comparison to the other seven cards taken as a whole. For instance, the sum of the first row is 53, whereas the sum of the second row is 465. This result indicates that participants preferred Card 2 to the other cards more than 8 times more frequently than they preferred Card 1 to the other cards. As Card 2 had a probability of reward of 0.4, and Card 1 was never rewarded in this condition, these values make sense because they are consistent with the reward structure of the independent cues condition. We expect, in fact, that there should be a strong relationship between the sum of each row and the reward probabilities from Table 7-2. For instance, the R^2 between the row sums in Table 7-18 and the independent cues reward probabilities from Table 7-2 is 0.87.

This relationship suggests one method for relating each participant's preference judgments to the ideal reward probabilities associated with the cards in their environment: Compute the R^2 between the row sums of each participant's preference matrix and the card-choice task reward probabilities. Table 7-19 provides the mean R^2 for each of the five card-choice task conditions, where each mean is the average of this measure taken over 40 participants.

Two analyses were conducted on the data used to create the means presented in Table 7-19. The first was an ANOVA involving the four conditions of the card-choice task that make up a 2×2 factorial design which crosses interaction type (XOR vs. AND) with probability of reward (high vs. low). This ANOVA reveals a significant main effect of type, $F(1, 156) = 27.263$, $p = 5.6\text{e-}07$, $\eta 2 = 0.145$, and a significant interaction between type and reward, $F(1, 156) = 4.625$, $p = .033$, $\eta 2 = 0.025$. However, the main effect of reward was not significant, $F(1, 156) = 0.099$, $p = .754$, $\eta 2 = 0.001$. Post hoc tests conducted using the Tukey HSD statistic reveal that there was a significantly better fit between sums of rows of preference matrices and sets of ideal reward probabilities for participants who experienced the AND interaction than for those who experienced the XOR interaction ($p < .300\text{e-}07$). The significant interaction reflects the fact that the fit between these two variables was lower for Low-Reward AND than for High-Reward AND, but the effect of reward was in the opposite direction for the two XOR conditions.

Table 7-19. The mean squared correlation between the sums of the rows of each participant's preference matrix and the ideal reward probabilities.

	High-Reward AND	High-Reward XOR	Independent	Low-Reward AND	Low-Reward XOR
Mean R^2	0.655	0.349	0.638	0.579	0.451
SE	0.036	0.047	0.045	0.039	0.043

Note. Each mean is based on 40 participants. The SE row provides the standard errors of these means.

The second statistical analysis involved using Welch two-sample *t* tests to compare performance of participants in the independent cues condition to the other four conditions. The results of these comparisons are provided in Table 7-20. These results indicate that in terms of the relationship between preference matrix row sums and ideal reward probabilities, there was no significant

Table 7-18. The sum of each preference matrix for the 40 participants in the independent cues condition of the card-choice task.

Cue Pattern	Card	1	2	3	4	5	6	7	8	Sum
(0,0,0)	1	0	6	8	6	7	8	7	11	53
(0,0,1)	2	154	0	83	29	108	21	40	30	465
(0,1,0)	3	152	77	0	24	94	22	22	25	416
(0,1,1)	4	154	131	136	0	135	71	106	35	768
(1,0,0)	5	153	52	66	25	0	11	20	29	356
(1,0,1)	6	152	139	138	89	149	0	104	38	809
(1,1,0)	7	153	120	138	54	140	56	0	37	698
(1,1,1)	8	149	130	135	125	131	122	123	0	915
	Sum	1067	655	704	352	764	311	422	205	4480

differences between participants in the independent cues condition and participants in either of the conditions in which there was an AND interaction between cues. In contrast, participants in the independent cues condition had a significantly stronger relation between row sums and reward probabilities than was the case for either of the conditions involving an XOR interaction between cues. The pattern of results presented in Table 7-20 because it reveals that performance for the independent cues condition is higher when measured by Phase II preferences than might be predicted from performance in this condition when measured by Phase I choice behavior.

Table 7-20. The results of using Welch two-sample t

	High-Reward AND	**High-Reward XOR**	**Low-Reward AND**	**Low-Reward XOR**
Independent Cues	*t* = -0.29741	*t* = 4.4577	*t* = -0.99842	*t* = 3.0268
	df = 74.863	*df* = 77.734	*df* = 76.792	*df* = 77.874
	p = .767	*p* = 2.749e-05	*p* = .3212	*p* = .00335

7.9.3 Distance From Simulated Preference Matrices

The previous section examined the relationship between human card preferences and ideal reward probabilities using one measure: the sums of the rows of participants' preference matrices. In this section, we explore human preferences with a different approach that involves measuring the distances between participants' preference matrices and simulated preference matrices created from ideal reward probabilities.

Imagine a pair of objects *i* and *j* that can be assigned some real value score that permits the two objects to be compared (s_i and s_j). For example, the two objects might be sports teams, and their score might be a measure of team ability. If the two teams played one another, what is the probability that team *i* would defeat team *j*? One answer to this question comes from statistics in the form of the Bradley–Terry model (Bradley & Terry, 1952). According to this model, this probability—represented as *P(i > j)*—can be defined as in Equation 7-7:

$$P(i > j) = \frac{s_i}{(s_i + s_j)} \tag{7-7}$$

The Bradley–Terry model can be used to simulate card preferences, and these simulated card preferences can be compared to the actual preference matrix for each participant. Each simulated preference matrix is based upon the assumption that an ideal agent knows the actual probability of reward (Table 7-2) for each stimulus. These probabilities are used as the scores in Equation 7-7 to compute the probability that card *i* is preferred to card *j* for any pair of compared cards. Table 7-21 presents a simulated preference matrix for the independent cues condition. This matrix was created by first simulating the 112 preferences made by a single participant using ideal probabilities in Equation 7-7. A random number between 0 and 1 was generated and compared to the Equation 7-7 probability for each compared pair. If the random number was greater than the value from Equation 7-7, then cell (*j, i*) in the preference matrix was incremented by 1. Otherwise, cell (*i, j*) was incremented by 1. This process was then repeated until the data from 500 participants had been simulated. These 500 preference matrices were then averaged to produce the ideal preference matrix presented in Table 7-21. The same procedure, using appropriate probabilities from Table 7-2, was used to generate ideal preference matrices for the remaining four conditions of the card-choice task.

The logic of creating these five ideal preference matrices was to compare them to the actual preference matrices generated by each individual participant. This comparison was conducted by computing the Euclidean distance between an ideal matrix and a participant matrix; the smaller the measured distance, the stronger the relationship between ideal probabilities and the dimension used by a participant to decide whether to choose one card over another from the various card pairs. Table 7-22 provides the average distance between ideal and participant preference matrices for each condition in the card-choice task, where this average is the mean distance computed over the 40 participants in each condition.

As was the case in Section 7.9.2, two analyses were conducted on the data used to create the means presented in Table 7-22. The first was an ANOVA involving the four conditions of the card-choice task that make up a 2×2 factorial design that crosses interaction type (XOR vs. AND) with probability of reward (high vs. low). This ANOVA reveals a significant main effect of type, *F*(1, 156) = 23.020, *p* = 3.72e-06, η2 = 0.128. Neither the main effect of reward, *F*(1, 156) = 0.635, *p* = .427, η2 = 0.004, nor the Type × Reward interaction, *F*(1, 156) = 0.231, *p* = .631, η2 = 0.001, were statistically significant. Post hoc tests conducted using the Tukey HSD statistic reveal that the distance between ideal and participant preference matrices was significantly shorter for conditions involving

Table 7-21. The ideal preference matrix for Phase II of the independent cues condition of the card-choice task.

Cue Pattern	Card	1	2	3	4	5	6	7	8	Sum
(0,0,0)	1	0	0.012	0.014	0.006	0.03	0.004	0.012	0.006	0.084
(0,0,1)	2	3.988	0	2.686	1.682	2.964	1.798	2.21	1.626	16.954
(0,1,0)	3	3.986	1.314	0	1.112	2.302	1.134	1.532	0.988	12.368
(0,1,1)	4	3.994	2.318	2.888	0	3.164	2.134	2.478	1.844	18.82
(1,0,0)	5	3.97	1.036	1.698	0.836	0	0.884	1.248	0.814	10.486
(1,0,1)	6	3.996	2.202	2.866	1.866	3.116	0	2.432	1.84	18.318
(1,1,0)	7	3.988	1.79	2.468	1.522	2.752	1.568	0	1.416	15.504
(1,1,1)	8	3.994	2.374	3.012	2.156	3.186	2.16	2.584	0	19.466
	Sum	27.916	11.046	15.632	9.18	17.514	9.682	12.496	8.534	112

Note. This matrix presents the average set of preferences for the 112 card comparisons generated from the Bradley–Terry model for 500 simulated participants.

the AND interaction in comparison to conditions involving the XOR interaction ($p < 3.7e\text{-}06$).

Table 7-22. The mean distance between ideal preference matrices and participant preference matrices.

	High-Reward AND	High-Reward XOR	Independent Cues	Low-Reward AND	Low-Reward XOR
M M Euclidean Distance	9.897	11.656	9.395	10.026	12.177
SE	0.309	0.356	0.193	0.382	0.544

Note. Each mean is based on 40 comparisons. The SE row provides the standard errors of these means.

The second statistical analysis involved using Welch two-sample *t* tests to compare preferences of participants in the independent cues condition to the other four conditions. The results of these comparisons are provided in Table 7-23.

These results indicate that in terms of the distance between ideal and participant preference matrices, there were no significant differences between participants in the independent cues condition and participants in either of the conditions in which there was an AND interaction between cues. In contrast, participants in the independent cues condition had significantly shorter distances between matrices than was the case for either of the conditions involving an XOR interaction between cues. This pattern of results parallels those reported in Table 7-20 for the analyses of preference matrix row sums.

7.9.4 Summary

This section has explored performance of human participants in the second phase of the card-choice task by examining their preferences when faced with the job of choosing which card in presented pairs is associated with a higher reward. These card preferences provide an alternative measure of what participants had learned about the probability structure of the environment.

Two measures of card preferences were considered. One compared the sums of the rows of card preference matrices to the reward probabilities associated with each card. The other examined the Euclidean distance between each participant's preference matrix and an ideal matrix simulated from reward probabilities.

Analyses of both measures revealed an important finding that was also present in the analyses of probability learning in the first phase of the card-choice task. When human preferences were compared to ideal reward probabilities, performance was more accurate for conditions involving an AND interaction between cues than for conditions involving an XOR interaction between cues.

Table 7-23. The results of using Welch two-sample t tests to compare the distances for the independent cue condition to distances computed for the four other conditions in the chard-choice task.

	High-Reward AND	High-Reward XOR	Low-Reward AND	Low-Reward XOR
Independent Cues	$t = -1.3756$	$t = -5.5821$	$t = -1.4717$	$t = -4.8205$
	$df = 65.372$	df = 60.09	$df = 57.66$	$df = 48.67$
	$p = 0.1736$	$p = 6.032e\text{-}07$	$p = .1465$	$p = 1.445e\text{-}05$

Analyses of both measures also suggested that preferences for the independent cue condition were more accurate (i.e., more strongly related to ideal reward probabilities) than was apparent when choice behavior was examined. This finding is more consistent with the various simulation results that have been presented earlier. Phase II of the card-choice task may have produced this result because it measured participant performance after all learning was complete. All of the choice behavior described earlier included choices made by participants in the early trials of the study when they had no knowledge of the probability structure of the environment. It is possible that participants learned more about the independent cues condition than choice behavior can reveal, and that preference measures taken at the end of learning provide a more accurate measure of overall learning.

One interesting difference between measures of choice behavior and measures of card preferences was that the latter did not reveal a significant effect of level of reward associated with an interaction between cues. It is not clear why this variable would affect performance in Phase I of the card-choice task, but would not affect performance in Phase II. One possibility is that the two tasks tap different kinds of processing (e.g., learning vs. decision-making) and that probability discrimination learning is affected by reward level, but making judgments after this learning has taken place is not. Another possibility is that because the preference task forces participants to choose, their behavior is more "digital" than is the case when learning is occurring. For instance, Bruner et al. (1956, Chap. 7) discussed this type of conversion in the context of learning probabilistically defined concepts. Clearly, this issue requires future exploration.

7.10 Summary and Implications

7.10.1 Summary

Chapters 4, 5, and 6 provided many formal and empirical details about what perceptrons learn when they are confronted with an uncertain environment. However, perceptrons are very simple artificial neural networks, and this in turn means that there are limits to what they can learn. The connectionist approach to human cognition began to thrive only after researchers discovered methods for training networks that were more powerful because their structure was more complex (Medler, 1998). This is because it was presumed that human cognition was complicated enough to require powerful network models that included hidden units.

The purpose of the current chapter was to determine whether simple networks like perceptrons are possible models of human probability learning. It did so by collecting data from human participants in a card-choice task whose structure was analogous to the three-cue task used to train perceptrons in Chapters 5 and 6. Of interest is whether human performance would demonstrate patterns similar to those observed in these simple networks. If so, then this would suggest that both human and perceptron learn to discriminate probabilities by behaving like naïve Bayesians, which in turn supports the notion that perceptrons are possible models of human performance.

In general, the results of the card-choice task support the claim that human probability learners do indeed behave as if they are naïve Bayesians. Our analyses of choice behavior revealed that the type of interaction between cues, and the level of reward associated with this interaction, affected human choices in a similar fashion to that observed for perceptrons. That is, the relationship between human probability matching and ideal reward probabilities was stronger for the AND interaction than for the XOR interaction and was stronger when the interaction signaled lower reward than when it signaled higher reward. Our analyses of preference judgments replicated the effect of interaction type. The fact that human performance was affected by cue interactions, and was more adversely affected by a linearly nonseparable interaction than by a linearly separable interaction, provides support for using perceptrons to model human probability discrimination learning. At the very least the results of the card-choice task did not suggest that human probability discrimination learning was so powerful that it demanded modeling by systems more sophisticated than the perceptron.

One interesting discrepancy between human choice behavior and perceptron responses was observed for the independent cues condition. In general, human participants were poorer probability matchers for this condition than were perceptrons. It was suggested that this result might be changed if human participants received more training in this condition. Ignoring this possibility, this result certainly does not motivate the exploration of more powerful models. This is because the human participants are not performing as well as simple networks when facing independent cues.

Human performance in the card-choice task was also examined to find evidence that participants, like perceptrons, are actually learning about reward probabilities. This was accomplished by determining whether simple heuristic processes, such as "randomly choose but stay when rewarded," could account for the succession of choices from card to card by human participants. It could be hypothesizes that by following such a strategy, humans could appear to be probability matchers without learning any reward probabilities at all. However, by comparing a number of simulated choice strategies to human data, we discovered that the best strategies (a) did not include a rule like "stay with a card that rewards" and (b) used reward probabilities to choose the next card. We have no evidence to support the claim that our participants were not learning the probabilities of reward associated with the various stimuli in the card-choice task.

7.10.2 Implications

In general, the analyses of human performance on the card-choice task support the view that can plausibly be modeled by simple artificial neural networks like perceptrons. Our human participants behaved as if they were having difficulties identifying signals associated with cue interactions, producing behavior that was similar to that of the perceptrons. This indicates that for this particular probability discrimination task, both perceptrons and people are behaving like naïve Bayesians. Our results do not support exploring more powerful or sophisticated systems as models of human probability discrimination learning.

There are many important implications of a potential correspondence between the probability learning of people and the probability learning of perceptrons. First, this means that one can explore human probability learning by exploiting our formal understanding of how probability is related to perceptron structure (e.g., Chapter 4). Second, this means that one can use the probability learning of perceptrons to explore a variety of experimental situations to find promising results that can be pursued by studying human participants. Indeed, the entire experiment that has been detailed in this chapter was motivated by the simulation results presented in Chapters 5 and 6. Third, this suggests that a well-understood phenomenon—associative learning—can provide an account of basic probability learning processes in biological agents. These implications are unpacked in more detail in the next and final chapter of this monograph.

Chapter 8: Synthetic Psychology And Probability Learning

The purpose of this final chapter is to review the results that have been reported in this monograph and to consider their implications. The chapter proceeds by first considering the computational results from formal examinations of how perceptrons adapt to uncertain environments. The key result was establishing that perceptrons belong to the class of naïve Bayesians. The chapter then considers the algorithmic results from simulation studies of perceptrons and from experimental studies of human participants. The results of the simulation studies are consistent with expectations based on the claim that perceptrons are naïve Bayesians. The results from the card-choice task support the claim that human participants can also be viewed as naïve Bayesians. The chapter ends by considering future avenues of using artificial neural networks to study probability learning, future variations of the card-choice task, and the prospects of using the methodology of synthetic psychology to study probability learning.

8.1 Adapting to Uncertain Worlds

8.1.1 The Causal Texture of the Environment

In 1935, Edward Tolman and Egon Brunswik presented a theory of the relationship between biological agents and their environment (Tolman & Brunswik, 1935). At the core of this theory is the notion that the environment is a causal texture of events; different events regularly depend on one another, and therefore one event can be used as a local representative or signal of another. Thus, an organism can behave appropriately in the context of distal or directly unobservable situations by responding to locally available and observable signals of these distal events.

Another critical property of this causal texture is that dependencies between events in the world are equivocal to some degree. Local signals are not connected to distal events in a one-to-one fashion but can instead signal different distal events with different frequencies. In short, our relation to the world is intrinsically probabilistic. For Tolman and Brunswik (1935), this meant that both the perception of signals and the selection of appropriate actions require organisms to generate hypotheses. "A hypothesis 'asserts' that a given 'a' is the local representative of a given 'b'" (Tolman & Brunswik, 1935, p. 47). However, the probabilistic nature of the causal texture of the environment imparts uncertainty to these hypotheses. "The organism behaves 'as though'. That is, he ventures a hypothesis. He may be right; but he may also be wrong" (Tolman & Brunswik, 1935, p. 46).

In Chapter 1, we reviewed three sources of uncertainty about the environment. One is epistemic uncertainty, our inability to predict the future perfectly due to our limited knowledge of the world. Another is somatic uncertainty, in which constraints in our bodily structure (e.g., sensory systems) prevent us from acquiring all of the information that we need to predict the future. A third is ontic uncertainty, which results from the fact that various worldly causal relations or events are intrinsically probabilistic. All three of these sources of uncertainty can coexist, and each is consistent with Tolman and Brunswik's (1935) position that the causal texture of an organism's environment is intrinsically uncertain.

8.1.2 Coping With Uncertainty

For Tolman and Brunswik (1935), an organism's ability to survive within an environment that has an uncertain causal texture requires it to generate appropriate hypotheses. From whence do these hypotheses emerge, and how might they be modified by experience?

We saw in Chapter 1 (e.g., Figure 1-1) that Brunswik and his followers attempted to answer this question via the theory of probabilistic functionalism (Brunswik, 1943, 1952, 1955; Brunswik & Herma, 1951; Hursch et al., 1964; Peterson, Hammond, & Summers, 1965a, 1965b; Tucker, 1964). Many other theories have also arisen. These include viewing organisms as intuitive statisticians (P. W. Cheng & Holyoak, 1995; Gigerenzer & Murray, 1987; Peterson & Beach, 1967), as implementing mathematical decision or choice theory (Bock & Jones, 1968; Einhorn & Hogarth, 1981; Luce, 1959; Resnik, 1987), as exploiting cognitive heuristics of one form or another (Gigerenzer, 2000, 2010; Tversky & Kahneman, 1974), as exploiting mechanisms that predict future sensory stimulation (Clark, 2016; Hohwy, 2013), as estimators of Bayesian probabilities (Chater, Oaksford, Hahn, & Heit, 2010; Doya, 2007; Oaksford & Chater, 2007) or as using learning mechanisms that emphasize probabilistic relations like contingency (Allan, 1993; Allan, Siegel, & Tangen, 2005; Jenkins & Ward, 1965; Wasserman, Elek, Chatlosh, & Baker, 1993).

This list is incomplete, and these various theories need not be mutually exclusive. If we take the probabilistic nature of the causal texture of the environment as a given, then it makes sense that organisms may possess a variety of means for coping with uncertainty.

8.1.3 Cognitive Science, Association, and Uncertainty

The current monograph has explored another theory about adapting to uncertainty. It focused on the associative learning of an extremely simple system, the perceptron. We have considered how perceptron learning is affected when the relationship between stimuli and responses is probabilistic. One reason for studying this simple system is that its connection to core psychological theories of associative learning has long been established (Dawson, 2008; Gluck & Bower, 1988; Gluck & Myers, 2001; Sutton & Barto, 1981). As a result, by understanding how perceptrons learn in probabilistic environments, we are also in a position to make claims about how theories of associative learning are related to uncertainty.

The current research demonstrates the study of associative learning and probability by exploiting two related approaches from cognitive science. The first is using synthetic psychology to explore learning under uncertainty. In synthetic psychology, one studies a phenomenon by first building a simple working system and only then by examining its behavior when it is placed in different environments (Braitenberg, 1984; Dawson, 2004, 2013; Dawson et al., 2010). The logic of this approach is that a simple system is often capable of generating interesting, complicated, or surprising behavior because of the manner in which it is coupled with its environment. Furthermore, given that the researcher has constructed this simple system, she should be in a much better position to explain how this surprising behavior emerges. For Braitenberg (1984), synthetic psychology is a methodology for producing simpler theories of complicated performance.

Synthetic psychology explores “what if” questions by using working systems (Braitenberg, 1984; Minsky, 1985). We have followed this approach by investigating what happens to perceptron learning if the relationship between training stimuli and desired responses is uncertain. In the beginning, we trained perceptrons on very simple problems, including probabilistic variants of Boolean operators (Chapters 2, 3, and 4). These simulations revealed several important regularities. First, when learning in uncertain environments, perceptron responses quickly adapt and reach an equilibrium in which the activity of an output unit represents the probability of reward associated with a stimulus (Figures 1-8, 2-2, 2-3). In other words, perceptrons exhibit probability matching similar to that long studied by psychologists (Estes, 1964). Second, this equilibrium is dynamic: If the uncertain relationship between cues and rewards is altered, perceptrons quickly adapt to match the new probabilities (Figure 2-5). Third, the probabilities being learned by perceptrons appear to be Bayesian in nature (Table 4-2).

The second methodology adopted in the current monograph is cognitive science’s multiple levels of analysis. In general, cognitive science analyses phenomena at different levels, using different paradigms, in an attempt to capture different kinds of regularities (Marr, 1982; Pylyshyn, 1984). At the computational level of analysis, formal methods are used to define what information-processing problem is being solved by a system. At the algorithmic level of analysis, behavioral methods are employed to discover the information processing procedures that the system is using to solve this problem. At the implementational level of analysis, attempts are made to determine how these information processing procedures are realized by physical mechanisms.

The current monograph has explored learning under uncertainty at the computational level by using mathematical analyses to determine what kind of probabilities are being computed by perceptrons and how these computations are reflected in the details of perceptron structure. It has also explored learning under uncertainty at the algorithmic level by examining perceptron performance when the conditions of a simple three-cue probability discrimination task are manipulated. In particular, simulation studies have determined how both nonoperant and operant perceptron learning is affected when multiple cues simultaneously signal reward probability, including situations in which signals are carried by cue interactions. Furthermore, the performance of human participants in a card-choice task has been studied in an attempt to relate human probability learning processes to those of perceptrons. We have not explicitly considered probability learning at the implementational level, although some might argue that one advantage of using perceptrons to study this phenomena is their biological or neuronal plausibility (McClelland, Rumelhart, & Hinton, 1986; McCulloch & Pitts, 1943). We consider this issue in more detail in Section 8.6.

The remainder of this chapter summarizes our results and considers their implications. It begins by first considering the main computational results and by reviewing algorithmic results from both networks and human participants. It then considers the implications of these results in terms of relationships to current theories and models, as well as with respect to future research directions.

8.2 Computational-Level Results

8.2.1 Perceptrons Are Naïve Bayesians

Our computational analyses of probability learning by perceptrons were motivated by simulation results indicating that perceptron responses can be interpreted as estimates of the probability of reward (Chapters 2 and 3; see also Dawson & Dupuis, 2012; Dawson et al., 2009a;

Dawson & Gupta, 2017). Given these results, we were interested in determining the mathematical basis of perceptron responses. We did so by translating these responses into mathematical equations of conditional probability.

Our first computational analysis considered the simplest of perceptrons, a network with a single input unit and a single output unit (Chapter 3). Such a network can be trained on a probabilistic problem typically represented using a 2×2 contingency table (e.g., Table 3-1). Under the assumption that the network's output unit employed the popular logistic activation function, we were able to translate network responses into a particular conditional probability equation, Bayes's theorem.

Our second computational analysis extended the simple proof of Chapter 3 into the situation in which the probability of reward is signaled by two cues that can appear simultaneously. We translated perceptron activity into Bayes's theorem, but in this case the equation took the specific form of naïve Bayes (Equation 4-5). This equation computes the probability of reward, using states of different cues, under the assumption that the cues are independent signals. We used this result to predict the performance of perceptrons trained on a variety of stochastically defined Boolean operators (Table 4-11). Finally, we developed this proof in such a way that it could be extended to cases in which reward probability was signaled by three or more different cues.

8.2.2 Weights and Odds Ratios

The computational analyses just summarized focus on one characteristic: the relationship between perceptron activity and Bayes's equation for conditional probability. Important to note, computational analyses also provide important information about perceptron structure, detailing exactly how connection weights and biases represent causal relations that are converted into conditional probabilities.

In particular, as part of our proofs in Chapters 3 and 4, it was demonstrated that the value of any connection weight or bias in a modern perceptron is related to a particular measure of association, the odds ratio. An odds ratio represents the effect of a variable on an outcome. It is the ratio of the odds that the outcome will occur if the variable is present to the odds that the outcome will occur if the variable is absent. Our computational analyses show that any perceptron weight is the natural logarithm of the odds ratio associated with the cue with a signal that is modified by that weight. This means that perceptrons work in the world of natural logarithms, summing these logarithms up when net input is calculated, and finally using the exponential component of the logistic activation function to convert logarithmic net input into probability.

There are several implications of this finding that are discussed in more detail in Section 8.3. Let us simply note them now. First, we now have an explicit understanding of how perceptron structure is related to computing conditional probability. Second, this permits the structure of a perceptron to be mapped directly onto the coefficients of logistic regression. We took advantage of this mapping in Chapters 5 and 6 to confirm our expectations about perceptron structure. Third, this suggests that the odds ratio deserves a more explicit role in theories about how agents associate stimuli and responses in uncertain environments.

8.2.3 Input Encodings

One additional computational analysis was performed later in the monograph (Section 7.6.4). This analysis examined the effect of different encodings of cues. The proofs in Chapters 3 and 4, and the simulations in Chapters 5 and 6, represented the presence of a cue with 1, and represented the absence of a cue with 0. However, cues were never absent when participants examined stimuli in the card-choice task (Chapter 7); They were always visible and were in one of two states that were represented by using two colors. A mathematical analysis of perceptron structure was conducted to determine how perceptron structure would be altered by using different encodings of the two possible states of a cue.

The computational analysis described in Section 7.6.4 compared perceptron structure using two encodings. The first used the values [0, 1], consistent with the interpretation that a cue in one state was literally absent. The second used the values [1, 2], consistent with the interpretation that a cue was always present, but had different appearances in its two states. Our analysis revealed three important effects of encoding.

First, changes in encoding did not alter the values of a perceptron's connection weights. This is not surprising, because regardless of encoding, each cue has the same relationship to the desired output (i.e., each cue signals the same probability of reward). As a result, each cue should be associated with the same odds ratio with either encoding and therefore should be assigned the same connection weight.

Second, marked differences were observed in the biases of perceptrons that experienced different input encodings. In other words, perceptrons learned to deal with encoding differences by keeping the same connection weights and by adjusting their bias. Again, because connection weights should be independent of encoding, this makes perfect sense—the only structural component that can be adjusted is the bias.

Third, a simple expression (Equation 7-6) indicated the relationship between biases of perceptrons trained with different input encodings. One can determine the bias of a [0, 1] encoding perceptron by summing the bias and the weights of a [1, 2] encoding perceptron. The main import of this computational analysis is that the results that we obtained from perceptrons are applicable to those obtained from human participants regardless of the encoding that was used to train the networks.

This section has briefly summarized three major results of our computational analyses of perceptrons trained in uncertain environments. The next section turns to considering a number of implications of these results.

8.3 Computational-Level Implications and Future Directions

8.3.1 Toward Bayesian Mechanisms

One theme of this final chapter is the utility of considering adaptation to uncertainty from multiple levels of analysis: computational, algorithmic, and implementational (Marr, 1982). However, Marr did far more than simply argue for multiple levels of analysis: He argued that it was necessary to establish links between levels and provided case studies of how to do so. For example, in Chapter 2 of his seminal 1982 book, Marr made a computational argument for using a difference of Gaussians function to detect visual edges, proposes algorithms for applying this function and related this function to single-cell recordings from neurons in the lateral geniculate nucleus. One can look at his figure of these recordings (Marr, 1982, Figure 2-17) and literally see their relation to the mathematical function!

Marr established links between different levels of analysis via an explicit top-down strategy. He began with the most abstract (the computational) level and then worked downward through the more concrete algorithmic and then the implementational levels. Marr offered two justifications for this top-down approach. First, he observed that one could study the computational level in the absence of results at lower levels of analysis. Hence, the computational level was to Marr a natural and obvious starting point for an investigation. Second, Marr believed that lower-level results would not constrain higher-level inquiries. Thus, one should proceed with a top-down research strategy because the constraints propagate from the computational level downward and not in the reverse direction.

One finds modern examples of Marr's top-down approach in Bayesian theories of human reasoning, which offer one approach to explaining adapting to uncertainty (Oaksford & Chater, 2007, 2009; Oaksford et al., 2000). This work deliberately begins at the computational level. Oaksford and Chater (e.g., 2007) pointed out that a computational theory of reasoning provides an account of a system's competence or of its ideal behavior. Bayesian probability is their choice for a computational account of human reasoning and rationality.

However, Bayesian cognitive science illustrates a problem that emerges when the top-down research strategy is interrupted—when links to lower levels of analysis are not pursued. For example, Oaksford and Chater (e.g., 2007) are reluctant to explore their theory of reasoning at other levels of analysis. That is, they do not propose Bayesian algorithms or mechanisms for human reasoning. They believe that algorithms or mechanisms will at best only approximate their computational theory.

> We suspect that, in general, the probabilistic problems faced by the cognitive system are simply too complex to be solved directly, by probabilistic calculation. Instead, we suspect that the cognitive system has developed relatively computationally "cheap" methods for reaching solutions that are "good enough" probabilistic solutions to be acceptable. (Oaksford & Chater, 2007, p. 15)

Critics claim the failure to explore probabilistic algorithms or mechanisms is a fundamental flaw of Bayesian cognitive science (Bowers & Davis, 2012; Jones & Love, 2011). Although there is a growing literature concerning the Bayesian nature of neural mechanisms (Doya, 2007; Rao et al., 2002), skeptics point out that this evidence is behavioral and not biological (Bowers & Davis, 2012). In addition, it is argued that the nature of Bayesian computation is incompatible with psychologically plausible algorithms or mechanisms for carrying them out, such as artificial neural networks (Jones & Love, 2011). In other words, critics claim that agents may behave as if they are using Bayesian probability to make judgments or decisions, but Bayesian cognitive scientists have failed to provide an account of procedures or mechanisms that explain exactly how this actually accomplished.

One implication of the computational results reviewed in Section 8.2, and of the general approach used to obtain them, is that they point the way to algorithmic and possibly implementational accounts of Bayesian inference. To begin, these results clearly establish that perceptrons behave "as if" they are using the naïve Bayesian equation to estimate the probability of reward signaled by a particular pattern of cues. However, because these results were developed in a synthetic framework—we conducted a computational analysis of a working system, the perceptron—we also have a clear understanding about *how* this "as if" behavior comes about. First, we are already armed with a detailed understanding about how perceptrons learn associations between stimuli and responses, not to mention how this processing is related to well-developed psychological accounts of associative learning (Dawson, 2008) like the Rescorla–Wagner model (Rescorla & Wagner, 1972). Second, our results permit us to say much more than the associations learned by a perceptron represent the probabilistic relation between a cue and a reward. We can also say exactly *how* the relation is represented—as the natural logarithm of an odds ratio—and *how* this representation is converted into a conditional probability, via the logistic activation function. In short, our analyses do not merely say that perceptrons behave as if they are naïve Bayesians, they also provide a detailed account of how this behavior is implemented by the structure that perceptrons attain via associative learning in an uncertain environment.

8.3.2 The Importance of Odds

One important discovery from our computational analysis of perceptron structure is that perceptron weights are literally the natural logarithm of a particular measure of association, the odds ratio. This discovery has implications for relating perceptrons to the psychological study of association, and has implications for relating earlier computational analysis of more complex networks to this same literature.

As was noted earlier in Section 3.6, the odds ratio is one of the most important statistical measures of association for a contingency table (Agresti, 2002; Y. M. M. Bishop et al., 1975; Rudas, 1998). Manipulating such an association is fundamental to the psychological study of how biological agents learn contingent relations between stimuli and responses. One core issue in this literature how should this association be measured or represented (Allan, 1980; Allan & Jenkins, 1980, 1983; P. W. Cheng, 1997; P. W. Cheng & Holyoak, 1995; P. W. Cheng & Novick, 1990, 1992; De Houwer & Beckers, 2002; Jenkins & Ward, 1965; Rescorla, 1967, 1968; Shanks, 2007; Ward & Jenkins, 1965). Allan and Jenkins examined several different measures of association and concluded that the best one is the difference in conditional probabilities ΔP, which is equal to the difference between the conditional probability of the response when the stimulus is present and the conditional probability of the response when the stimulus is absent.

However, the odds ratio is one measure of association that has *not* been considered in the psychological literature on contingency, even though it is the preferred measure statisticians. This is unfortunate, because we saw in Chapter 3 that for simple contingency tables, there is a very strong relationship between ΔP and the natural logarithm odds ratio (Figure 3-1). Furthermore, we saw that the odds ratio offers additional advantages over ΔP. First, as we know from Chapters 5 and 6, the odds ratio can be used in situations involving many-dimensioned contingency tables, whereas ΔP is only defined for the 2×2 case. As well, the odds ratio can be computed when some contingency table cells are equal to zero, but this is not true for ΔP. Finally, the odds ratio can generate results that are consistent with the ΔP literature. For instance, the behavior of the modern perceptron can generate ΔP even though it does not explicitly represent this metric in is structure (Dawson & Dupuis, 2012).

It appears that our computational results suggest a new metric of association is worthy to be considered by psychologists interested in contingency learning. A preliminary investigation of our own suggests that this exploration will be fruitful. One study provides the frequency values for 25 different 2×2 contingency tables (constructed to generate an interesting range of ΔP values), as well as human ratings related to these tables obtained in two studies (Wasserman et al., 1990). Wasserman et al. found that ΔP was highly correlated with both sets of human ratings ($r = .94$ with one study, $r = .93$ with the other). We took these same 25 contingency tables and computed the natural logarithm of the odds ratio for each one. For this sample of contingency data, the correlation between ΔP and $ln(ad/bc)$ was essentially perfect ($r = .999$). Given this relationship between the two metrics, it is no surprise that the natural logarithm of the odds ratio generated equally high correlations with the human judgments ($r = .94$ with one set, $r = .93$ with the other). Clearly the odds ratio is worthy of further study in the contingency literature and offers the possibility for extending this literature into the study of more complicated (e.g., three or more cue) situations.

8.3.3 Interpreting Deep Belief Networks

The history of using artificial neural networks to study human cognition reveals a trend toward using networks of greater and greater complexity (Medler, 1998). The earliest networks that could learn from experience had no hidden units intervening between input and output units (Rosenblatt, 1958, 1962; Widrow & Hoff, 1960). The limited power of these networks resulted in declining interest in connectionist models of cognition (Minsky & Papert, 1969). A revival in connectionist cognitive science occurred in the mid-1980s with the discovery of techniques that could train networks that included hidden units, and which were therefore far more powerful than earlier architectures (McClelland & Rumelhart, 1986; Rumelhart et al., 1986; Rumelhart & McClelland, 1986). Networks with a single layer of hidden units had the in principle power to model the complex capabilities of human cognition (Hornik et al., 1989; Lippmann, 1989; Siegelmann, 1999; Siegelmann & Sontag, 1995). However, in practice, the learning rules for these systems had a great deal of trouble realizing the potential power of these networks, and again connectionist cognitive science waned. More recently, new learning rules have been discovered for training of so-called deep belief network (Bengio, 2009; Bengio et al., 2013; Hinton, 2002, 2007; Hinton et al., 2006). These networks have several layers of hidden units, and new advances in deep learning permit training such networks to solve difficult problems in speech recognition, object detection, and recognition and a variety of other domains that depend on discovering regularities in big data (LeCun, Bengio, & Hinton, 2015).

Artificial neural networks have two roles in the study of cognition. One is serving as an artifact to accomplish some intelligent task, such as performing gene sequencing (DePristo et al., 2011), detecting diabetic retinopathy from photographs (Gulshan et al., 2016), predicting traffic flow (Lv, Duan, Kang, Li, & Wang, 2015), or discovering new drugs by finding regularities in large amounts of experimental data (Gawehn, Hiss, & Schneider, 2016; Mamoshina, Vieira, Putin, & Zhavoronkov, 2016). Deep belief networks excel as intelligent artifacts and are driving many new developments in artificial intelligence (LeCun et al., 2015).

The other role of artificial neural networks is to provide theoretical insights to cognitive science. However, the ability to create a working model does not immediately imply a greater understanding of a phenomenon (Lewandowsky, 1993). Computer simulations of cognitive processes often face Bonini's paradox: The simulation can be as difficult to understand as is the phenomenon being modeled (Dutton & Starbuck, 1971). This may be particularly true of artificial neural networks, for network structures are typically very difficult to understand because of their messy, distributed, nonlinear nature (McCloskey, 1991; Mozer & Smolensky, 1989; Seidenberg, 1993). "One thing that connectionist networks have in common with brains is that if you open them up and peer inside, all you can see is a big pile of goo" (Mozer & Smolensky, 1989, p. 3). Accordingly, connectionists typically accept that the internal structure of their networks is very difficult to understand, and detailed analyses of network structure are rare occurrences in the literature. With their many layers of hidden units, the interpretation of deep belief networks presents an even greater challenge (Erhan et al., 2010).

Certainly, networks are hard to interpret. However, it is not impossible to explore the internal structure of a trained network in order to explain how it converts its inputs into its responses. Connectionist cognitive scientists have developed many techniques for interpreting the internal structure of artificial neural networks that are simpler than modern deep belief networks (Baesens, Setiono, Mues, & Vanthienen, 2003; Berkeley, Dawson, Medler, Schopflocher, & Hornsby, 1995; Dawson, 2004, 2005; Gallant, 1993; Hanson & Burr, 1990; Hayashi, Setiono, & Yoshida, 2000; Hinton, 1986; Moorhead, Haig, & Clement, 1989; Omlin & Giles, 1996; Setiono, Baesens, & Mues, 2011; Setiono, Thong, & Yap, 1998; Taha & Ghosh, 1999). These techniques can reveal surprising new theories of cognitive phenomena that have been discovered by networks as they learn to accomplish a task (Dawson, 2018).

There is a pressing need to develop similar techniques for interpreting the structure of deep belief networks. First, the successes of these networks in learning difficult tasks indicate that they are more and more likely to be explored by cognitive scientists. However, their usefulness for cognitive science requires the ability to understand how they actually solve complicated problems. Second, new legalities are forcing deep learning researchers to move beyond mere developing an artifact that can solve problems and to provide means of interpreting why networks respond as they do. For instance, in May 2018 the European Union activated the General Data Protection Regulation (GDPR), which is a set of laws designed to strengthen and unify data protection for all European Union members. Some provisions in the GDPR have been interpreted as making the use of deep belief networks illegal if the reasons that these networks make particular responses is not understood. For instance, Article 22 of the GDPR stipulates that individuals cannot be subject to a decision (e.g., being denied a loan) based on a purely algorithmic decision, and Recital 71 gives an individual the right to have the reason for an algorithmic decision explained to them. If deep belief networks cannot be interpreted, then they will be unable to meet these legal requirements.

Methods for understanding the internal workings of deep belief networks are being developed, but this research is in its infancy (Erhan et al., 2010). Much of this research emphasizes the interpretation of networks that have been trained on visual tasks. This is because different approaches can be used to visualize what hidden units are responding to, permitting hierarchies of visual features to be mapped onto the processing layers of a network (Liu et al., 2017; Mousavi, Siqueira, Barros, Fernandes, & Wermter, 2016; Zeiler & Fergus, 2014). Other approaches for networks trained on nonvisual problems have also been explored (Mallat, 2016; Montavon et al., 2018) but are less common, and sometimes approach the issue by redefining the purpose of interpretation. For instance, Montavon et al. (2018) "seek to characterize the model's black-box behavior, without however trying to elucidate its inner workings or shed light on its internal representations" (p. 2).

Of course, the networks we have been exploring in this monograph are dramatically simpler than modern deep belief networks. Nevertheless, our understanding of perceptron structure that results from adapting to uncertainty has implications for the problem of understanding the internal structure of modern systems. This is because the properties of a perceptron are in fact quite similar to the properties of a processing layer in a deep belief network. For example, in many deep belief networks, there are layers of hidden units connected to a layer of units that provide weighted signals. The hidden units often employ an activation function like the logistic. The hidden units operate by computing their logistic activation and then by turning on with the probability indicated by this activation. In other words, each hidden unit in this layer operates in a fashion that is very similar to an operant perceptron.

Our analyses indicate that the probabilistic behavior of such a system can be explained by using the odds ratio as a measure of the association between an input signal and the output probability. Under the assumption that this understanding maps onto a processing layer in a deep belief network, this suggests a method for considering links between layers in terms of a specific probabilistic measure of input effects on responses. This could provide a means of interpreting the kinds of (probabilistic) features that hidden units respond to even in cases where a deep belief network has not been trained on a visual problem. In addition, the link between perceptrons and deep network layers would also suggest that these probabilistic features do not involve interactions, because the probabilistic relationship between layers would be consistent with naïve Bayes.

8.3.4 Cue Encoding and Bias

Section 7.6 provided an extended analysis of the effect of input encoding on perceptron structure. This analysis examined simulation results that compared perceptrons trained with different encodings of the same problem and developed a proof to show how the structure of a perceptron trained with one encoding could be translated into the structure of a different perceptron trained with another encoding. The key implication of this analysis was that the formal results developed in Chapters 3 and 4 are still relatable to human performance in the card-choice task in spite of the fact that in the latter no cues were ever absent.

Another implication of the computational analysis provided in Section 7.6 is that manipulations of cue states in something like the card-choice task should have very specific effects. Consider a logistic regression equation that predicts the probability of reward associated with a particular stimulus (i.e., a particular pattern of cues). This prediction starts with a general expectation of reward probability that is represented by constant of the regression equation. This general expectation is then modified upward or downward by combining the constant with the state of each cue multiplied by the regression coefficient associated with the cue. The same method is implemented in a perceptron, where the general expectation is represented by an output unit's bias, and this expectation is further modified by the signals arriving through the network's weighted connections.

The analysis in Section 7.6 indicates that changes in encoding will not affect the predictive power of individual cues (e.g., connection weights). However, the same analysis predicts that changes in encoding will alter the general expectation of reward (e.g., output unit bias). This formal analysis suggests if one were to manipulate the relative salience of cue states in the card-choice task, then one should affect a participant's general expectation of reward but not the predictive power of the individual cues. Furthermore, the size of this effect should be related to cue salience (see Equation 7-6). This prediction could be tested by seeing, for example, if manipulations of cue salience affected the likelihood that human participants choose the card with no cues present.

This is a straightforward example of one advantage of examining a phenomenon at multiple levels. In this particular case, a computational result predicts a result that should arise in an algorithmic or behavioral analysis. In addition, the use of simulations in Section 7.6 provides an example of one advantage of the synthetic approach. Computational results have established the relationship between models of animal learning and the perceptron (Dawson, 2008). This relationship in turn implies that regularities observed when perceptrons learn should also be observed in animal learning. Important to note, it is much easier to simulate experiments with perceptrons than with biological agents. Experiments on perceptrons can be used to discover interesting or surprising effects, or to identify manipulations that are not of interest. Once the simulations have been used to survey a variety of experiments, one can then use these results to motivate experiments using biological agents. The idea is that these experiments are expensive and time-consuming to conduct, so the synthetic approach offers some guidance to maximize experimental payoff. This theme is encountered again later in this chapter.

8.3.5 Alternative Activation Functions

The formal analyses presented in Chapters 3 and 4 proceeded by translating the mathematical expression for Bayes's theorem into the mathematical expression for the logistic activation function. The ability to perform this translation provides the proof that the perceptron is implementing Bayesian probability theory.

One consequence of this formal approach is the tacit assumption that the probabilistic behavior of a perceptron depends upon its using the logistic activation function. One problem with this assumption is that it constrains the applicability of the relationships that we have explored between perceptron behavior and probability. This is because the logistic equation is only one of many mathematical functions that have been explored in the connectionist literature. One review discovered that researchers have explored the use of 640 activation functions in artificial neural networks (Duch & Jankowski, 1999). Might networks that use activation functions other than the logistic also adapt appropriately to uncertain environments and generate responses that can be related to one theory of probability or another?

This issue is particularly relevant in our lab because it has an extensive history of studying artificial neural networks that use a Gaussian activation function instead of the logistic (Dawson & Schopflocher, 1992). These systems, called *networks of value units*, have many interesting properties. First, they lead to networks that have more computational power than do similar-sized networks that only use the logistic (Dawson & Schopflocher, 1992). Second, although networks of value units have the same formal relationship to models of animal learning as do more traditional perceptrons, they can generate different responses because the Gaussian is essentially a different model of how internal associations are converted into behavior (Dawson, 2008). Third, networks of value units have certain emergent features that make them particularly well suited to projects that require interpreting the internal structure of networks after they have been trained (Berkeley et al., 1995; Dawson, 2018).

A pilot study suggests that value unit networks, like perceptrons, can adapt to uncertain environments, but their responses take on different values than those observed in networks that use the logistic. We trained two types of perceptrons on five training sets used in an earlier study (Dawson & Gupta, 2017). Each of these training sets uses four cues, with each cue serving as an independent signal of reward probability. The ideal reward probability associated with each of the 16 cue patterns in a training set is provided in Table 8-1. Both types of perceptrons were trained for 2,500 epochs on a training set using a learning rate of 0.05; networks were initialized using the same procedure reported earlier in Chapters 5 an 6. A network was trained only once on each of the five training sets. One set of five networks used the logistic activation function and were therefore identical to the perceptrons described in Chapter 5. The other set of five networks used the Gaussian activation function. The mean response of each type of network (averaging over the five networks) for each stimulus is also provided in Table 8-1.

Table 8-1. The average responses of two types of perceptrons to the five independent cue training sets studied by Dawson and Gupta (2017).

Cue Pattern	Ideal Reward Probability	Average Logistic Response	Average Gaussian Response
(0,0,0,0)	0.000	0.155	0.332
(0,0,0,1)	0.630	0.754	0.796
(0,0,1,0)	0.640	0.554	0.630
(0,0,1,1)	0.900	0.955	0.962
(0,1,0,0)	0.350	0.371	0.551
(0,1,0,1)	0.880	0.908	0.918
(0,1,1,0)	0.810	0.798	0.822
(0,1,1,1)	0.980	0.986	0.889
(1,0,0,0)	0.210	0.225	0.331
(1,0,0,1)	0.790	0.826	0.814
(1,0,1,0)	0.700	0.659	0.638
(1,0,1,1)	0.960	0.970	0.989
(1,1,0,0)	0.470	0.480	0.556
(1,1,0,1)	0.940	0.939	0.940
(1,1,1,0)	0.770	0.860	0.848
(1,1,1,1)	0.960	0.991	0.913

Note. One type of perceptron uses the logistic activation function, whereas the other uses the Gaussian activation function. The average is the mean of the responses of five networks, each trained on a different version of the training set, to the same stimulus.

The pilot results presented in Table 8-1 replicate the finding that perceptrons using the logistic activation function do an excellent job at matching the ideal reward probabilities. The squared correlation between the ideal probabilities and the logistic responses is 0.961. Perceptrons using the Gaussian activation function are poorer probability matchers, but their relationship to the ideal probabilities is still surprisingly high. The squared correlation between the ideal probabilities and the logistic is 0.915. This is a surprising value because when the Gaussian activation function is used to replace the logistic, the formal results relating perceptron responses to Bayesian probabilities no longer hold.

The value unit networks in Table 8-1 perform surprisingly well, which suggests that probability matching is possible even when alternate activation functions are used. This raises a number of new computational questions that require formal exploration. When the Gaussian activation function is used, can network responses be translated into probabilities? If so, are these probabilities Bayesian, or do networks of value units implement an alternative probability theory? What is the relationship between the structure of a value unit network trained on an uncertain training set and probability theory—do its weights represent odds ratios too? Finally, it is well established that networks that use value units are more powerful than those with units that use the logistic, because a value unit perceptron can learn to solve some linearly nonseparable problems (Dawson, 2005; Dawson & Schopflocher, 1992). Does this mean that such networks will behave differently when presented cue interactions than do the perceptrons that have been reported in this monograph?

8.4 Algorithmic-Level Results

The two preceding sections have reviewed the computational results reported earlier in this monograph and have considered the implications of these results. We now turn to reviewing results that have emerged from studies that have considered probability learning at the algorithmic level of analysis. These results come from two types of studies: computer simulations involving perceptrons, and experiments involving human participants. These studies reflect the algorithmic level of analysis because they focus on empirical observations of learning behavior and do not involve formal proofs.

8.4.1 Probability Discrimination by Nonoperant Perceptrons

The formal analyses of Chapters 3 and 4 led to the basic conclusion that when a perceptron learns about the reward probabilities signaled by multiple cues, it acts as a naïve Bayesian. In other words, the probabilities that it learns are constrained by the assumption that each cue is an independent signal of reward probability. A naïve Bayesian is blind to interactions between cues.

Chapter 5 presented a number of computer simulations that empirically tested this formal conclusion. Each simulation involved training perceptrons (with logistic activation functions) on a task in which reward was uncertain. These perceptrons were trained with a nonoperant procedure, meaning that they learned after every stimulus presentation; they did not choose whether to learn. In this task, perceptrons were presented stimuli defined by three cues, each of which could be present or absent. Thus, eight stimuli could be presented.

The pattern of cues on a stimulus signaled reward probability. In one condition, each of the three cues was an independent signal of the likelihood of reward. In four other conditions, reward probability was signaled by the state of one cue and by the state of the interaction between the other two cues. These four conditions defined a 2×2 factorial design. One factor was the logical type of the interaction between the two cues. In two conditions the logical AND of the two cues was the signal of reward probability, whereas in the other two the logical XOR of the two cues provided this signal. This factor was included because perceptrons can represent linearly separable functions like AND but cannot represent linearly nonseparable functions like XOR (Minsky & Papert, 1969). The other factor was the size of reward probability that was signaled by the interaction between cues. In two conditions, the interaction signaled a smaller probability of reward than it did in the other two conditions. This factor was included because when the reward associated with an interaction between cues is higher, the interaction is more important, and therefore should pose greater difficulties to a naïve Bayesian system (Dawson & Gupta, 2017).

The simulation results are consistent with the hypothesis that perceptrons are naïve Bayesians. First, consider the four conditions that crossed the two factors of the type of interaction between cues and the amount of reward signaled by this interaction. Analyses were performed to examine how closely network responses to each stimulus matched the ideal reward probability associated with the stimulus. These analyses found a significant effect of type of interaction: probability matching was better for perceptrons exposed to the AND interaction than for perceptrons exposed to the XOR interaction. A significant effect of amount of reward was also revealed: Probability matching was better for perceptrons when the cue interaction signaled a low reward than when it signaled a high reward. A significant interaction between these two factors was also evident because the manipulation of amount of reward had a stronger effect on the XOR interaction than it did on the AND interaction.

Second, consider the relationship between perceptron responses and reward probabilities when the independent cues condition is compared to the other four conditions. It was discovered that the independent cues condition led to a significantly better relationship between responses and probabilities than three of the conditions (High-Reward AND, High-Reward XOR, Low-Reward XOR) but a significantly poorer relationship than was the case for the Low-Reward AND condition. This last result was surprising, but further investigation of the stochastically generated training sets for the conditions indicated that there was enough conditional dependence randomly present in the independent cues condition to account for this finding.

If perceptrons are naïve Bayesians, then their structure (weights and bias) should represent odds ratios. This hypothesis was also tested in Chapter 5 by performing logistic regression on each of the training sets. The coefficients delivered by this type of analysis are known to represent odds ratios. When the structures of the various perceptrons were compared to the regression equations that were obtained, a very strong relationship between the two was observed. This provided additional empirical support to the notion that the perceptrons were behaving as expected from the Chapter 4 formalisms.

8.4.2 Probability Discrimination by Operant Perceptrons

One purpose of training perceptrons with three cues that signaled reward probability was to provide simulation results that could be compared to human performance on the card-choice task. However, one difference between the Chapter 5 simulations and the human experiment reported in Chapter 7 is that human performance in the card-choice task is operant. Each event in the first phase of the card-choice task requires a human participant to choose a stimulus for learning. One concern is that the operant nature of the card-choice task distinguishes it from the Chapter 5 simulations. To address this issue, Chapter 6 reported the results of simulations that used an operant technique to train perceptrons.

Chapter 6 studied the same experimental conditions used in the Chapter 5 simulations. The difference between the simulations in the two chapters was that Chapter 6 studied two different operant paradigms. In each paradigm, on being presented a stimulus the perceptron generated a response (its prediction of reward probability for that stimulus). It then used this probability to decide whether to learn about the stimulus. These operant procedures were intended to make perceptron learning more similar to human performance in the card-choice task, in the sense that the networks (like the humans) learned only about chosen stimuli.

One operant paradigm used an increasing returns rule, for which the probability of choosing to learn about a stimulus was equal to the predicted reward probability. The other operant paradigm used a gambler’s fallacy rule, for which the probability of choosing to learn about a stimulus was equal to 1 minus the predicted reward probability. This meant that the two paradigms implemented opposite types of choice rules.

One major finding in Chapter 6 was that, regardless of operant paradigm, the same pattern of results was observed as was reviewed in Section 8.3.1. Furthermore, regardless of operant paradigm, perceptrons achieved the same final structures (weights and biases) as did the Chapter 5 nonoperant perceptrons. The only reliable difference between operant paradigms involved the number of stimuli that perceptrons chose. In general, although the gambler’s fallacy networks chose to learn about half of the stimuli that they encountered, increasing returns networks chose to learn about one third of the stimuli. However, differences in the number of stimuli learned about, or in the particular stimuli selected for learning, did not cause the perceptrons to respond differently to the stimuli in the different conditions once training had ended.

8.4.3 Probability Discrimination by Human Participants

The formal analyses of Chapters 3 and 4, and the algorithmic analyses of Chapters 5 and 6, clearly established the naïve Bayesian nature of perceptrons. An important question that was not answered, though, was whether this knowledge also applies to human probability learners. This is a particularly important issue because perceptrons are frequently ignored as models of human information processing because of the general assumption that they are not powerful enough to capture the computational power of human cognition (Dawson, 2004). A vital issue is to determine whether models of human probability learning need to be more powerful than perceptrons.

Chapter 7 presented the results of human participation in the card-choice task. The purpose of this task was to determine whether perceptrons could possibly serve as models of human probability learning. The card-choice task was designed to be analogous to the three-cue simulations used in Chapters 5 and 6. In Phase I, participants explored an environment composed of eight cards. Each card was identified by a unique pattern of three symbols, each in one of two colors. Each cue signaled the probability that a participant would receive a reward if she selected it. This phase of the study ended after each participant made 320 different card choices. They then proceeded to

Phase II, in which they were presented pairs of cards and had to choose which member was more likely to offer a reward on the basis of Phase I learning.

The purpose of the card-choice task was to collect human data that were analogous to the simulation data collected in Chapters 5 and 6. To this end, five different experimental conditions were created, with each condition implementing the same relationship between cue states and reward probabilities that characterized the different conditions of the simulation studies. There were 40 human participants in each of these conditions.

The question of interest was whether manipulations of the mapping from cues to rewards would affect human participants in a similar fashion as they affected probability learning in perceptrons. Phase I of the card choice replicated two of the key effects observed in the simulations (e.g., Table 7-4). Like perceptrons, the relationship between reward probability and human probability matching (i.e., the likelihood of human participants choosing the various cards) was poorer when there was an XOR interaction between two cues than when there was an AND interaction between two cues. As well, and again like perceptrons, there was a poorer relationship between reward probability and human probability matching when an interaction between cues signaled a higher reward than when an interaction signaled a lower reward.

One surprising discrepancy in the Phase I data collected from human participants was that, in terms of their probability matching behavior, they performed poorer than expected when the three cues on the cards were independent signals of reward probability. It was suggested that this might be because human participants needed more training in this condition. Nevertheless, this discrepancy is not one that suggests that perceptrons are not powerful enough to model human probability matching. This is because with respect to this condition, perceptrons are better performers than are the human participants. In short, this discrepancy does not motivate exploring models that are more powerful.

One of the hypotheses that motivated the card-choice task was that the perceptron might serve as a possible model of human probability learning. This hypothesis was tested, as just reviewed, by comparing performance between the different conditions. This hypothesis was also tested by exploring how well human data could be fit using logistic regression equations. The fit of these equations to human performance was examined because a logistic regression equation provides a mathematical description of a perceptron. These analyses indicated that these equations captured a great deal of the variance in human choice behavior in each condition (Table 7-12). They also revealed striking similarities between the equations fit to human choices and the structures of networks trained on analogous problems (Table 7-13).

Additional analyses were conducted on the Phase I data to explore the possibility that human participants *appeared* to be choosing cards in a pattern consistent with reward probability but were actually using a different strategy that did not require learning anything at all about the likelihood that the different cards would be rewarded. For instance, participants might simply randomly choose a card and keep on choosing this card as long as it provided a reward. However, data simulations based on this strategy did not fit human choices as well as other simulations that assumed that participants were using reward probabilities to guide card choices (Figure 7-5).

Phase II of the card-choice task was conducted, in part because of the operant simulation results from Chapter 6. Those simulations demonstrated that very different patterns of stimulus choices could lead to the same underlying knowledge of an environment's probability structure. This motivated measuring the knowledge of human participants using some other means than stimulus choices, which provided the key dependent measure for Phase I. Card preferences provide the alternative dependent measure in Phase II; such preferences are presumed indicators of underlying probabilities (Savage, 1954) accomplished this by measuring card preferences. Card preferences were summarized by creating a preference matrix for each participant that indicated the frequency that one card in a presented pair was preferred over the other. These matrices were then examined in two ways: by correlating the sums of their rows to ideal reward probabilities (Table 7-19), and by measuring the distance between a preference matrix and other matrices simulated by applying the Bradley–Terry model to ideal reward probabilities (Table 7-22).

Statistical analyses of the relationship between the sums of preference matrix rows and ideal reward probabilities replicated a key finding from Phase I: The preferences of human participants were more strongly related to reward probabilities for AND interaction conditions than for XOR interaction conditions. These analyses also revealed that performance in the independent cues condition was better in Phase II than might have been expected from the Phase I data. However, these analyses did not reveal any statistically significant effects of manipulating the probability of reward signaled by interacting cues. When distances between preference matrices and simulated matrices were examined, it was again discovered that there was a stronger relationship for AND interaction conditions than for XOR interaction conditions. Again, no significant effect was revealed involving the probability of reward associated with cue interactions. Finally, comparisons of distance measures between conditions placed the independent cues conditions as having a significantly better relation to reward probabilities than was the case for the two XOR conditions, and as having a significantly poorer relation to reward probabilities than was the case for the two AND conditions. This pattern of results was similar to that observed in Phase I of the card-choice task.

8.5 Algorithmic-Level Implications and Future Directions

8.5.1 Algorithmic Implications

When the cognitive revolution began in psychology in the middle of the 20th century, one of the topics that flourished was the study of concept learning (Bruner et al., 1956; Hunt, 1962). Bruner et al. developed a set of stimuli that varied on a number of features. A rule based on these features was created to define a category, and participants had the task of learning this rule. In some studies, participants chose specific cards and learned whether the selected card belonged to the target category. In other studies, participants were presented stimuli, again being informed about the relationship between each stimulus and the target category. This general approach was used to study difficulty of learning different kinds of category rules, strategies for exploring a stimulus space in search of the correct rule, and other topics central to cognitive psychology.

One issue that received a good deal of attention in the concept learning literature during this era was the relative ease of learning conjunctive concepts versus disjunctive concepts. A conjunctive concept involves a category defined by the necessary presence of two or more features (e.g., red triangle). In other words, it is defined using the logical AND of features. In contrast, a disjunctive concept is defined using the logical OR of features (e.g., red and/or triangle). For this latter example, a red stimulus, a triangle stimulus, or a red triangle stimulus all belong to the disjunctive concept. Results from a variety of experiments indicate that conjunctive concepts are more easily learned by human participants than are disjunctive concepts (Bruner et al., 1956; Conant & Trabasso, 1964; Haygood & Bourne, 1965; Laughlin & Jordan, 1967; Neisser & Weene, 1962; Wells, 1963).

One question of interest was the source of the difference in ease of learning of conjunctive versus disjunctive concepts. Two main hypotheses were explored. One was that conjunctive concepts are more common in everyday experience and are easier to talk about, which leads to more practice and skill in thinking or using them. It is certainly the case that people believe that conjunctive categories are more common than are disjunctive categories (Brockner, Paruchuri, Idson, & Higgins, 2002) and have a preference for hypothesizing conjunctive categories over disjunctive categories (Hunt & Hovland, 1960). The other hypothesis about the ease of learning conjunctive concepts over disjunctive concepts was that "disjunctive concepts are inherently more difficult, or psychologically complex, because the human brain is not built to think disjunctively" (Snow & Rabinovitch, 1969, p. 1). Evidence supported this second hypothesis. For instance, children of different ages show the same differences in performance between conjunctive and disjunctive concepts, indicating that this difference is not due to different amounts of experience (King, 1966; Snow & Rabinovitch, 1969).

The results of algorithmic-level analyses that have been presented in this monograph represent an extension of this older research tradition, because our results concern concepts that are defined by probabilistic rules. Our results from human participants show that performance is better for AND interactions than for XOR interactions, which is consistent with earlier research demonstrating better learning of conjunctive concepts over disjunctive concepts. XOR is a different kind of disjunction (Hunt, 1962). Furthermore, our results from computer simulations, coupled with our formal understanding of perceptrons and their limitations, position us to propose one reason why this difference occurs. This reason is that human participants, like perceptrons, learn reward probabilities in the card-choice task by behaving like naïve Bayesians who have difficulty with cue interactions. As a disjunctive interaction defined by XOR introduces more conditional dependence than does a conjunctive interaction defined by AND, probability learning in a disjunctive condition is poorer than this learning in a conjunctive condition. Furthermore, the observed effect of manipulating reward as well as the type of cue interaction (on perceptron learning and on human probability matching in Phase I) adds further support to this Bayesian hypothesis because reward can affect performance when the logical structure of the task is held constant. This effect is because reward manipulations can also affect conditional dependence, which in turn alters the performance of naïve Bayesians (Dawson & Gupta, 2017).

When cognitive psychology began to study concept learning, it started by considering this learning from older perspectives more aligned with psychological behaviorism (Hunt, 1962). It concluded that these older theories were not adequate and proceeded to develop new theories that were consistent with cognition as being a form of information processing. For instance, Hunt explored computer simulations that developed decision trees that could be used to model human learning of categories. Bruner et al. (1956) considered category learning as involving strategies for proposing, testing, and modifying hypotheses. We saw in Section 8.1.2 that these information-processing inspired approaches evolved in various directions (probabilistic functionalism, intuitive statistics, decision theory, heuristics, and Bayesian inference) and that many of these approaches are applicable to uncertain environments.

Perhaps one major implication of our algorithmic-level analyses is that they suggest that associative learning still has a role to play in this literature. We have identified relationships between the performance of human participants and the performance of perceptrons when both learn about similar uncertain environments. Furthermore, we know that this relationship is particularly important because of another established relationship between perceptrons and psychological models of associative learning (Dawson, 2008). Because human performance appears to show some of the limitations that are revealed by studying perceptrons indicates that models of associative learning are relevant to studying adapting to uncertainty.

This is not to say that the current monograph is proposing the perceptron as a competitor to other cognitive theories. It is reasonable to expect that cognitive processes, such as using strategies to test hypotheses or employing heuristics to generate educate guesses, are important components of adapting to uncertainty. Nonetheless, we are claiming that associative learning has a role to play in this processing and might indeed be capable of providing information to be used by other cognitive processes (e.g., generating subjective probabilities that can be used to guide strategy).

This is also not to say that the current monograph has presented all of the necessary evidence to confirm the role of associative learning in adapting to uncertainty. Clearly more evidence—both algorithmic and computational—is required. However, the evidence that has been reported herein suggests that gathering additional information concerning association and probability is a worthy enterprise. Fortunately, the card-choice task is flexible enough to serve as a source for exploring this issue in more detail, as discussed in the next section.

8.5.2 Future Card-Choice Task Directions

The particular version of the card-choice task detailed in Chapter 7 was an attempt to collect data about human probability learning that could be directly compared to the performance of perceptrons trained on an analogous task. The particular manipulations that defined the various conditions described in that chapter were specifically designed to match manipulations that were explored in the Chapter 5 and 6 simulations. Some of the results of the card-choice task have themselves raised additional questions; other questions arise from our computational understanding of artificial neural networks. In our view, the card-choice task is flexible enough to be used to explore a variety of different variables in an attempt to further our understanding of human probability learning. Let us briefly consider some of these variables.

Additional cue interactions. Section 8.5.1 noted that a long-standing issue in the concept learning literature is the logical relationship used to define concepts as combinations of features. The card-choice task explored this issue by using two relationships (AND and XOR) to signal reward probability using the logical combination of two cues. If we restrict ourselves to two interacting cues, then 16 logical relationships can be defined between them. These different relationships are associated with different degrees of conditional dependence, not to mention different kinds of linear separability (e.g., Table 4-2). The card-choice task can be easily modify to explore alternative relationships between cues. Essentially, all that is required is to define the probability of reward associated with the cue relationship of interest and to then use the additive probability rule to determine the probability of reward associated with each stimulus. Once this is done, the card-choice task can proceed in the same fashion as described in Chapter 7. For additional experiments of this sort, our understanding of the perceptron would suggest that human performance on new logical relationships could be predicted from the conditional dependence that exists between interacting cues.

Additional training. In Chapter 7, one issue that emerged was that human choice probabilities did not match ideal reward probabilities to the expected degree for the independent cues condition. It was suggested (e.g., Figure 7-4) that this was possibly because human participants required more experience in the independent cues condition in order to learn the subtleties of its reward probabilities. Clearly the amount of training permitted in the card-choice task could be easily modified (i.e., by varying the number of choices participants can make in Phase I) to determine how this factor affects human probability learning.

Alternative reward probabilities. One variable that could be responsible for participants requiring more training in a card-choice task condition is the difference between reward probabilities associated with different stimuli. For instance, in Section 7.5.3 it was suggested that human participants found the independent cues condition difficult because there were relatively small differences in the reward probabilities associated with different stimuli. This possibility could be studied further with variations of the card-choice task. For instance, different versions of the independent cues condition could be created by varying the reward probability signaled by each cue, which in turn would manipulate similarities or differences between reward likelihoods of different stimuli. If the Section 7.5.3 hypothesis were correct, then it would be expected that probability learning in this condition would be hindered in conditions in which there were high similarities between reward probabilities.

Changing probabilities midway through learning. In Chapter 2, it was noted that when perceptrons are being trained in an uncertain environment, they adopt a structure that is maintained in a dynamic equilibrium (e.g., Figure 2-5). This structure provides the perceptron the best probability estimates possible given the fact that the network is never going to be completely correct when it predicts environmental rewards. However, this equilibrium is dynamic in the sense that if an environment's reward probabilities are altered, then the perceptron quickly modifies its structure to adapt to the new contingencies. This kind of rapid adaptation was not studied in Chapter 7 but could be easily implemented in the card-choice task. This would be accomplished by using one set of reward probabilities for a set number of participant choices and then by using a different set of reward probabilities for later choices. Do human participants also quickly adapt to changing contingencies? This methodology could be used to answer this question. It could also be used to explore whether learning one kind of probability in the first set of choices (e.g., one involving a particular logical relationship between cues) would help or hinder adapting to a different kind of probability when reward probabilities change.

Alternative encodings of cues. One issue that was singled out for attention in Chapter 7 involved cue encoding in the card-choice task. In particular we explored whether the (0, 1) encoding of cues used to study perceptrons prevented these results from being compared to human performance. This is because the perceptron encoding tacitly assumes that cues are either present or absent, but for human participants cues were always visible in one state or another. Additional computational analyses suggested that the perceptron results were applicable (Section 7.6). However, this issue can be explored in variations of the card-choice task. For instance, in one variation cues could literally be present or absent. Will this alter human probability learning? In another variation, the salience of the difference between a cue's two states could be systematically varied. The analyses of Section 7.6 predict that this latter manipulation would alter participants' general expectation of being rewarded, because if they are behaving like perceptrons, then this manipulation alters the bias term of the logistic equation.

Alternative numbers of cues. The card-choice task can easily be extended to involve variations of the cues used to signal reward probabilities. One obvious way to do so would be to use a different number of cues. For instance, if one were to use four cues instead of three, human results could be compared to perceptrons of the sort studied by Dawson and Gupta (2017). However, a more interesting approach might be to keep the number of cues constant but to vary the number of states that cues can adopt. For instance, classic studies of concept learning often used three values for each cue (Bruner et al., 1956). The card-choice task could easily be changed to explore this kind of manipulation. For instance, cues might take on one of three colors instead of two, with each color used to modify the probability of reward being signaled. This would make the environmental probabilities much more complicated to learn. It would be interesting to determine whether basic associative processes could still be used to model human probability learning in this more complicated environment.

The preceding paragraphs have suggested a number of different issues that could be explored by varying some basic properties of the Chapter 7 card-choice task. It is also true that this task records a great deal of information that could be used to explore probability learning in ways that were not discussed in Chapter 7. The next paragraphs briefly consider some additional analyses of the card-choice task that could be used to gain more insight into human probability learning.

Broadening the notion of cues being attended to by participants. The analyses in Chapter 7 were all conducted under the assumption that human participants were treating the cues on each card as signals of reward probability. This assumption is justified in the sense that participants were explicitly told that this was the case and were explicitly told that other facets of the task (such as card location) did not signal reward. However, this does not mean that these other properties did not affect performance. For instance, it is well established that certain variables can affect performance in navigation tasks even when these variables are irrelevant to the task, and when the task can be accomplished by paying attention to other highly reliable cues (K. Cheng, 1986). Thus, it is possible that one factor influencing human card choices involves card location. This factor might be more important in conditions where learning from cues is more difficult than in conditions where learning from cues is less difficult. The program that conducts the card-choice task records the location of a card whenever it is selected by a participant. Thus, it is possible to explore whether additional cues, like card location, are also being used.

Considering alternative choice strategies. It was already noted that the data collected for each participant in the card-choice task could be used to explore choice strategies: The process used by a participant to choose the next card. It was argued that the expected probability of reward played an important role in this decision (Section 7.8). Additional influences on card-choice strategy can also be explored. For instance, perhaps card location also influences which card is to be selected next—all things being equal in terms of expected reward probability, perhaps participants prefer choosing a next card that is closer to the current card than are others. This possibility suggests another manipulation of the card-choice environment: the physical arrangement of stimuli. If location is affecting some component of human choices, then these choices may be altered when stimuli are arranged in a different layout.

8.6 Implementational-Level Considerations

8.6.1 Networks and Biological Plausibility

Cognitive science's earliest approaches to modeling cognition were inspired by the digital computer (Dawson, 1998, 2013). As a result, cognitive scientists adopted functionalist philosophy and were concerned with what *functions* were involved in information processing; they were not concerned with how those functions were physically instantiated. Functionalism made computer simulations of cognition possible, because it permitted researchers to argue that their computer models were relevant even though the physical implementation of computers is enormously different from the biological implementation of cognition by the brain. Functionalism makes the physical differences between computers and brains irrelevant.

However, the rise of connectionism in cognitive science occurred because some cognitive scientists were convinced that biological implementation was important to consider. Connectionist cognitive scientists were motivated by the belief that artificial neural networks could fill a biological vacuum left by functionalism (Amit, 1989; Clark, 1989; Feldman & Ballard, 1982; McClelland et al., 1986; Rojas, 1996). Artificial neural networks were touted as being biologically plausible or neuronally inspired. Processing units are roughly equivalent to neurons, and connections between processors are roughly equivalent to synapses (see, e.g., the visual analogy rendered in Rumelhart, Hinton, & McClelland, 1986, Figure 1). The link between artificial neural networks and the brain has a long history. For example, the earliest networks arose from attempts to describe the all-or-none behavior of neurons with propositional logic (McCulloch & Pitts, 1943). The most detailed exposition of the perceptron was titled *Principles of Neurodynamics* (Rosenblatt, 1962). Clearly, artificial neural networks have provided a medium for emphasizing the relationship between cognition and the brain. These networks have been described as providing an implementational account of associative learning (Shanks, 1995). Critics have, in fact, argued that research that employs artificial neural networks cannot be properly viewed as being "cognitive" because these networks provide only implementational accounts (Broadbent, 1985; Fodor & Pylyshyn, 1988).

However, it is also true that many researchers are highly skeptical of the claim that artificial neural networks are more biologically plausible than are other kinds of models (Reeke & Edelman, 1988). For instance, one can generate long lists of properties of artificial neural networks that are clearly not true of the brain (Crick & Asanuma, 1986; Smolensky, 1988). As a result, PDP models are often vilified as oversimplifications by neuroscientists; some have called them stick and ball models (Douglas & Martin, 1991). Reeke and Edelman (1988) offered this blunt assessment of the neurophysiological relevance of PDP connectionism: "These new approaches, the misleading label 'neural network computing' notwithstanding, draw their inspiration from statistical physics and engineering, not from biology" (p. 144).

8.6.2 Are Perceptrons Biologically Plausible?

With respect to the current monograph and its relationship to the implementational level, one could ask whether the perceptron is biologically plausible. This monograph does not make this claim. Instead, and as has been shown in earlier sections of Chapter 8, the perceptron is of interest because it provides new computational and algorithmic insights into probability learning, and these insights can be used to generate new ideas for experiments.

That being said, it could be plausibly argued that the understanding of the perceptron that we have developed in the current monograph is such that it could be used to guide the search for implementational evidence about probability learning. An architecture in cognitive science provides an account of the basic or primitive characteristics of an information-processing system (Dawson, 2013).

We have developed a detailed architectural understanding of how perceptrons learn to estimate probabilities that includes an account of how information about probabilities is encoded in connection weights and how this encoding is transformed by the perceptron's activation function. If one was intent on using the perceptron to suggest an implementational account, then the first step would be to explore how these basic architectural properties could be implemented by neural systems. From this perspective, the perceptron might be more biologically plausible than other models of associative learning, like the Rescorla–Wagner model, which do not typically do not provide architectural details to be related to biological mechanisms.

8.7 The Synthetic Approach and Associative Learning

8.7.1 The Synthetic Approach

Section 8.5 suggested a number of variations of the card-choice task; many other variants of this task are likely possible. In pursuing any of these possibilities, there are a number of specific design decisions to be made (e.g., specific values of reward probabilities signaled by cues, specific values of cue attributes, etc.). How might one explore these card-choice task possibilities, seeking studies that might lead to particularly interesting or surprising results?

The answer to this question lies in one of the main themes of the current monograph: studying a phenomenon from a variety of perspectives. In particular, we have studied not only human probability learning experimentally but also this phenomenon computationally and algorithmically by using a particular artificial neural network, the perceptron. The relationship between this kind of network and psychological models of learning is well established (Dawson, 2008; Gluck & Bower, 1988; Sutton & Barto, 1981). One of the main implications of Chapter 7 is that there also exists a relationship between perceptron learning and human performance in a particular type of probability learning task. Together, these implications indicate that one can use perceptron simulations to explore the effects of a variety of manipulations on probability learning. The results of these simulations can identify effects that seem particularly interesting or important. These simulation results can then motivate particular experimental studies of human or animal participants.

The use of simulations to explore the potential benefits of various experiments is an example of the allure synthetic psychology (Braitenberg, 1984; Dawson, 2004, 2013; Dawson et al., 2010a). As was noted earlier in this chapter, synthetic psychology proceeds by building simple systems for exploring "what if" questions about behavior. Perceptrons are examples of simple systems; they are easily trained, and one can simulate a variety of experiments with the clever construction of different training sets. As a result, perceptrons can be quickly used to explore a variety of different experimental configurations in search of manipulations that seem particularly promising. It is much simpler and faster to perform this search with these networks than it is to conduct pilot studies with biological participants. If one is confident that perceptrons are potential models of human performance, an interesting result with networks is also likely to be discovered when an analogous study is performed with human or animal participants.

In many respects, the current monograph is a case study in this synthetic approach. The computational studies of perceptrons led to simulations that explored the effects of two variables (type of cue interaction, probability of reward signaled by the interaction) on probability learning. These simulations revealed significant effects associated with each of these factors. This result was deemed interesting enough to lead to a parallel probability learning study involving human participants. This study revealed a similar pattern of results and supported the notion that both perceptrons and human participants treated this particular type of probability learning as naïve Bayesians.

Of course, the synthetic psychology of probability learning can include other models than perceptrons. Even if we were to (unwarrantedly) restrict it to connectionist systems, we have already noted that other artificial neural networks may provide insight into adapting to uncertainty. These networks could include multilayer networks and networks that use activation functions other than the logistic. There is no reason that a synthetic approach could not also be developed around other kinds of simulations. However, we are confident that the results that have been summarized in the current chapter are sufficient to justify the further exploration of probability learning in a synthetic approach that focuses on simple perceptrons.

8.7.2 The Centrality of Associative Learning

One reading of the current monograph is as a case study in the synthetic psychology of probability learning. Another reading is as an exploration of the centrality of theories of associative learning. This exploration proceeds with one established link—the relationship between psychological models of associative learning and the kind of learning performed by the perceptron (Dawson, 2008). It has then explored links between perceptron learning and other core topics: probability theory, information theory, different types of feedback in cybernetics, and systems theory. There are important mappings from artificial neural networks to all of these disciplines. This, in turn, means that these same mappings should hold between these disciplines and psychological theories of associative learning.

To a large extent, theories of associative learning were casualties of the cognitive revolution in the 20th century (G. A. Miller, 2003). With the rise of new cognitive theories that focused on active information processing, theories that were linked to psychological behaviorism were abandoned because they were interpreted as modeling psychological processing as the passive response to the environment. Many criticisms of associative theories claim that associations are not powerful enough to model human cognition (Bever et al., 1968; Chomsky, 1959; Fodor & Pylyshyn, 1988). However, a slight change of perspective reveals that associationist models provide alternative proposals for information processing that share many properties with the cognitivist theories that reacted against them (Dawson, 1998, 2013, 2018). It is both possible and fruitful to consider associative learning as a topic within cognitive science (Shanks, 1995). I hope that this monograph has shown not only that theories of associative learning provide insight into probability learning but also that they do so by appealing to ideas that are foundations of the very disciplines that gave birth to cognitivism.

References

Agresti, A. (2002). *Categorical data analysis* (2nd ed.). New York, NY: Wiley-Interscience. https://doi.org/10.1002/0471249688

Albert, J. (2009). *Bayesian computation with R* (2nd ed.). New York, NY: Springer. https://doi.org/10.1007/978-0-387-92298-0

Allan, L. G. (1980). A note on measurement of contingency between two binary variables in judgment tasks. *Bulletin of the Psychonomic Society, 15*(3), 147–149. https://doi.org/10.3758/BF03334492

Allan, L. G. (1993). Human contingency judgments: Rule based or associative? *Psychological Bulletin, 114*(3), 435–448. https://doi.org/10.1037/0033-2909.114.3.435

Allan, L. G., & Jenkins, H. M. (1980). The judgment of contingency and the nature of the response alternatives. *Canadian Journal of Psychology-Revue Canadienne De Psychologie, 34*(1), 1–11. https://doi.org/10.1037/h0081013

Allan, L. G., & Jenkins, H. M. (1983). The effect of representations of binary variables on judgment of influence. *Learning and Motivation, 14*(4), 381–405. https://doi.org/10.1016/0023-9690(83)90024-3

Allan, L. G., Siegel, S., & Tangen, J. M. (2005). A signal detection analysis of contingency data. *Learning & Behavior, 33*(2), 250–263. https://doi.org/10.3758/BF03196067

Alloy, L. B., & Abramson, L. Y. (1979). Judgment of contingency in depressed and nondepressed students: Sadder but wiser. *Journal of Experimental Psychology-General, 108*(4), 441–485. https://doi.org/10.1037/0096-3445.108.4.441

Amari, S. (1967). A theory of adaptive pattern classifiers. *IEEE Transactions on Electronic Computers, Ec16*(3), 299–307. https://doi.org/10.1109/PGEC.1967.264666

Amit, D. J. (1989). *Modeling brain function: The world of attractor neural networks*. Cambridge, UK: Cambridge University Press. https://doi.org/10.1017/CBO9780511623257

Anderson, J. A. (1995). *An introduction to neural networks*. Cambridge, MA: MIT Press.

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Arthur, W. B. (1989). Competing technologies, increasing returns, and lock-in by historical events. *Economic Journal, 99*(394), 116–131. https://doi.org/10.2307/2234208

Arthur, W. B. (2015). *Complexity and the economy*. Oxford, UK: Oxford University Press.

Ashby, W. R. (1956). *An introduction to cybernetics*. London, UK: Chapman & Hall. https://doi.org/10.5962/bhl.title.5851

Ashby, W. R. (1960). *Design for a brain* (2nd ed.). New York, NY: Wiley & Sons. https://doi.org/10.1007/978-94-015-1320-3

Attneave, F. (1959). *Applications of information theory to psychology*. New York, NY: Holt.

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning, 47*(2–3), 235–256. https://doi.org/10.1023/A:1013689704352

Baesens, B., Setiono, R., Mues, C., & Vanthienen, J. (2003). Using neural network rule extraction and decision tables for credit-risk evaluation. *Management Science, 49*(3), 312–329. https://doi.org/10.1287/mnsc.49.3.312.12739

Baldi, P. (2002). A computational theory of surprise. *Information, Coding and Mathematics, 687*, 1–25. https://doi.org/10.1007/978-1-4757-3585-7_1

Baldi, P., & Itti, L. (2010). Of bits and wows: A Bayesian theory of surprise with applications to attention. *Neural Networks, 23*(5), 649–666. https://doi.org/10.1016/j.neunet.2009.12.007

Bayes, T. (1763). An essay toward solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London, 53*, 370–418.

Bechtel, W., & Abrahamsen, A. A. (2002). *Connectionism and the mind: Parallel processing, dynamics, and evolution in networks* (2nd ed.). Malden, MA: Blackwell.

Behrend, E. R., & Bitterman, M. E. (1961). Probability-matching in the fish. *American Journal of Psychology, 74*(4), 542–551. https://doi.org/10.2307/1419664

Belis, M., & Guiasu, S. (1968). A quantitative-qualitative measure of information in cybernetic systems. *IEEE Transactions on Information Theory, 14*(4), 593–594. https://doi.org/10.1109/tit.1968.1054185

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning, 2*(1), 1–127. https://doi.org/10.1561/2200000006

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 1798–1828. https://doi.org/10.1109/tpami.2013.50

Berkeley, I. S. N., Dawson, M. R. W., Medler, D. A., Schopflocher, D. P., & Hornsby, L. (1995). Density plots of hidden value unit activations reveal interpretable bands. *Connection Science, 7*, 167–186. https://doi.org/10.1080/09540099550039336

Berkson, J. (1955). Maximum likelihood and minimum chi^2 estimates of the logistic function. *Journal of the American Statistical Association, 50*(269), 130–162. https://doi.org/10.2307/2281102

Berlyne, D. E. (1960). *Conflict, arousal, and curiosity*. New York, NY: McGraw-Hill. https://doi.org/10.1037/11164-000

Berlyne, D. E. (1971). *Aesthetics and psychobiology*. New York, NY: Appleton-Century-Crofts.

Bever, T. G., Fodor, J. A., & Garrett, M. (1968). A formal limitation of associationism. In T. R. Dixon & D. L. Horton (Eds.), *Verbal behavior and general behavior theory* (pp. 582–585). Englewood Cliffs, NJ: Prentice-Hall.

Binmore, K. G. (2009). *Rational decisions*. Princeton, NJ: Princeton University Press.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford, UK: Oxford University Press.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York, NY: Springer.

Bishop, Y. M. M., Fienberg, S. E., & Holland, P. W. (1975). *Discrete multivariate analysis: Theory and practice*. Cambridge, MA: MIT Press.

Bitterman, M. E. (1956). Information and effect in incidental learning. *American Journal of Psychology, 69*(3), 410–416. https://doi.org/10.2307/1419043

Bock, R. D., & Jones, L. V. (1968). *The measurement and prediction of judgment and choice*. San Francisco, CA: Holden-Day.

Boole, G. (2003). *The laws of thought*. Amherst, NY: Prometheus Books. (Original work published 1854)

Bowers, J. S., & Davis, C. J. (2012). Bayesian just-so stories in psychology and neuroscience. *Psychological Bulletin, 138*(3), 389–414. https://doi.org/10.1037/a0026450

Bradley, R. A., & Terry, M. E. (1952). Rank analysis of incomplete block designs. 1. The method of paired comparisons. *Biometrika, 39*(3–4), 324–345. https://doi.org/10.1093/biomet/39.3-4.324

Braitenberg, V. (1984). *Vehicles: Explorations in synthetic psychology*. Cambridge, MA: MIT Press.

Broadbent, D. (1985). A question of levels: Comment on McClelland and Rumelhart. *Journal of Experimental Psychology: General, 114*, 189–192. https://doi.org/10.1037/0096-3445.114.2.189

Brockner, J., Paruchuri, S., Idson, L. C., & Higgins, E. T. (2002). Regulatory focus and the probability estimates of conjunctive and disjunctive events. *Organizational Behavior and Human Decision Processes, 87*(1), 5–24. https://doi.org/10.1006/obhd.2000.2938

Bruner, J. S., Goodnow, J., & Austin, G. A. (1956). *A study of thinking*. New York, NY: Wiley & Sons.

Brunswik, E. (1939). Probability as a determiner of rat behavior. *Journal of Experimental Psychology, 25*(2), 175–197. https://doi.org/10.1037/h0061204

Brunswik, E. (1943). Organismic achievement and environmental probability. *Psychological Review, 50*, 255–272. https://doi.org/10.1037/h0060889

Brunswik, E. (1952). *The conceptual framework of psychology* (Vol. 1). Chicago, IL: University of Chicago Press.

Brunswik, E. (1955). Representative design and probabilistic theory in a functional psychology. *Psychological Review, 62*(3), 193–217. https://doi.org/10.1037/h0047470

Brunswik, E., & Herma, H. (1951). Probability learning of perceptual cues in the establishment of a weight illusion. *Journal of Experimental Psychology, 41*(4), 281–290. https://doi.org/10.1037/h0057751

Cannon, W. B. (1932). *The wisdom of the body*. New York, NY: Norton.

Chater, N., & Oaksford, M. (1999). Ten years of the rational analysis of cognition. *Trends in Cognitive Sciences, 3*(2), 57–65. https://doi.org/10.1016/S1364-6613(98)01273-X

Chater, N., Oaksford, M., Hahn, U., & Heit, E. (2010). Bayesian models of cognition. *Wiley Interdisciplinary Reviews-Cognitive Science, 1*(6), 811–823. https://doi.org/10.1002/wcs.79

Cheng, K. (1986). A purely geometric module in the rat's spatial representation. *Cognition, 23*, 149–178. https://doi.org/10.1016/0010-0277(86)90041-7

Cheng, K. (2005). Reflections on geometry and navigation. *Connection Science, 17*(1–2), 5–21. https://doi.org/10.1080/09540090500138077

Cheng, K., & Newcombe, N.S. (2005). Is there a geometric module for spatial orientation? Squaring theory and evidence. *Psychonomic Bulletin & Review, 12*(1), 1–23. https://doi.org/10.3758/BF03196346

Cheng, P. W. (1997). From covariation to causation: A causal power theory. *Psychological Review, 104*(2), 367–405. https://doi.org/10.1037/0033-295X.104.2.367

Cheng, P. W., & Holyoak, K. J. (1995). Complex adaptive systems as intuitive statisticians: Causality, contingency, and prediction. In H. L. Roitblat & J.-A. Meyer (Eds.), *Comparative approaches to cognitive science* (pp. 271–302). Cambridge, MA: MIT Press.

Cheng, P. W., & Novick, L. R. (1990). A probabilistic contrast model of causal induction. *Journal of Personality and Social Psychology, 58*(4), 545–567. https://doi.org/10.1037/0022-3514.58.4.545

Cheng, P. W., & Novick, L. R. (1992). Covariation in natural causal induction. *Psychological Review, 99*(2), 365–382. https://doi.org/10.1037/0033-295X.99.2.365

Chernoff, H., & Moses, L. E. (1959). *Elementary decision theory*. New York, NY: Wiley.

Cherry, C. (1957). *On human communication*. New York, NY: Science Editions.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory, 2*(3), 113–124. https://doi.org/10.1109/TIT.1956.1056813

Chomsky, N. (1957). *Syntactic structures* (2nd ed.). Berlin, Germany: Mouton de Gruyter.

Chomsky, N. (1959). A review of B. F. Skinner's *Verbal behavior*. *Language, 35*, 26–58. https://doi.org/10.2307/411334

Christensen, R. (1997). *Log-linear models and logistic regression* (2nd ed.). New York, NY: Springer.

Clark, A. (1989). *Microcognition*. Cambridge, MA: MIT Press.

Clark, A. (2016). *Surfing uncertainty: Prediction, action, and the embodied mind*. Oxford, UK: Oxford University Press. https://doi.org/10.1093/acprof:oso/9780190217013.001.0001

Conant, M. B., & Trabasso, T. (1964). Conjunctive and disjunctive concept formation under equal-information conditions. *Journal of Experimental Psychology, 67*(3), 250–255. https://doi.org/10.1037/h0043068

Cornfield, J. (1962). Joint dependence of risk of coronary heart disease on serum cholesterol and systolic blood pressure: Discriminant function analysis. *Federation Proceedings, 21*(4), 58–61.

Cornfield, J. (1967). Bayes' theorem. *Revue De L Institut International De Statistique-Review of the International Statistical Institute, 35*(1), 34–49. https://doi.org/10.2307/1401634

Cornfield, J. (1969). The Bayesian outlook and its application. *Biometrics, 25*(4), 617–657. https://doi.org/10.2307/2528565

Cox, D. R. (1958a). The regression analysis of binary sequences. *Journal of the Royal Statistical Society Series B-Statistical Methodology, 20*(2), 215–242.

Cox, D. R. (1958b). Two further applications of a model for binary regression. *Biometrika, 45*(3–4), 562–565. https://doi.org/10.2307/2333203

Cox, R. T. (1961). *The algebra of probable inference*. Baltimore, MD: Johns Hopkins Press.

Cramer, J. S. (2003). *Logit models from economics and other fields*. Cambridge, UK: Cambridge University Press. https://doi.org/10.1017/CBO9780511615412

Crick, F., & Asanuma, C. (1986). Certain aspects of the anatomy and physiology of the cerebral cortex. In J. McClelland & D. E. Rumelhart (Eds.), *Parallel distributed processing* (Vol. 2, pp. 333–371). Cambridge, MA: MIT Press.

Cumming, G. (2013). The new statistics: A how-to guide. *Australian Psychologist, 48*(3), 161–170. https://doi.org/10.1111/ap.12018

Cumming, G. (2014). The new statistics: Why and how. *Psychological Science, 25*(1), 7–29. https://doi.org/10.1177/0956797613504966

David, F. N. (1962). *Games, gods, and gambling: The origins and history of probability and statistical ideas from the earliest times to the Newtonian era*. New York, NY: Hafner.

Dawson, M. R. W. (1991). The how and why of what went where in apparent motion: Modeling solutions to the motion correspondence process. *Psychological Review, 98*, 569–603. https://doi.org/10.1037/0033-295X.98.4.569

Dawson, M. R. W. (1998). *Understanding cognitive science*. Oxford, UK: Blackwell.

Dawson, M. R. W. (2004). *Minds and machines: Connectionism and psychological modeling*. Malden, MA: Blackwell. https://doi.org/10.1002/9780470752999

Dawson, M. R. W. (2005). *Connectionism: A hands-on approach*. Oxford, UK: Blackwell. https://doi.org/10.1002/9780470694077

Dawson, M. R. W. (2008). Connectionism and classical conditioning. *Comparative Cognition and Behavior Reviews, 3*(Monograph), 1–115. https://doi.org/10.3819/ccbr.2008.30008

Dawson, M. R. W. (2013). *Mind, body, world: Foundations of cognitive science*. Edmonton, Canada: Athabasca University Press.

Dawson, M. R. W. (2018). *Connetionist representations of tonal music: Discovering musical patterns by interpreting artificial neural networks*. Edmonton, Canada: Athabasca University Press. https://doi.org/10.15215/aupress/9781771992206.01

Dawson, M. R. W., Boechler, P. M., & Orsten, J. (2005). An artificial neural network that uses coarse allocentric coding of direction to represent distances between locations in a metric space. *Spatial Cognition and Computation, 5*, 29–67. https://doi.org/10.1207/s15427633scc0501_2

Dawson, M. R. W., Boechler, P. M., & Valsangkar-Smyth, M. (2000). Representing space in a PDP network: Coarse allocentric coding can mediate metric and nonmetric spatial judgements. *Spatial Cognition and Computation, 2*, 181–218. https://doi.org/10.1023/A:1013127731819

Dawson, M. R. W., & Dupuis, B. (2012). Equilibria of perceptrons for simple contingency problems. *IEEE Transactions on Neural Networks and Learning Systems*. https://doi.org/10.1109/TNNLS.2012.2199766

Dawson, M. R. W., Dupuis, B., Spetch, M. L., & Kelly, D. M. (2009). Simple artificial networks that match probability and exploit and explore when confronting a multiarmed bandit. *IEEE Transactions on Neural Networks, 20*(8), 1368–1371. https://doi.org/10.1109/TNN.2009.2025588

Dawson, M. R. W., Dupuis, B., & Wilson, M. (2010). *From bricks to brains: The embodied cognitive science of LEGO robots*. Edmonton, Canada: Athabasca University Press.

Dawson, M. R. W., & Gupta, M. (2017). Probability matching in perceptrons: Effects of conditional dependence and linear nonseparability. *Plos One, 12*(2), e0172431. https://doi.org/10.1371/journal.pone.0172431

Dawson, M. R. W., Kelly, D. M., Spetch, M. L., & Dupuis, B. (2008). Learning about environmental geometry: A flaw in Miller and Shettleworth's (2007) operant model. *Journal of Experimental Psychology-Animal Behavior Processes, 34*(3), 415–418. https://doi.org/10.1037/0097-7403.34.3.415

Dawson, M. R. W., Kelly, D. M., Spetch, M. L., & Dupuis, B. (2010). Using perceptrons to explore the reorientation task. *Cognition, 114*(2), 207–226. https://doi.org/10.1016/j.cognition.2009.09.006

Dawson, M. R. W., & Schopflocher, D. P. (1992). Modifying the generalized delta rule to train networks of nonmonotonic processors for pattern classification. *Connection Science, 4*, 19–31. https://doi.org/10.1080/09540099208946601

Dawson, M. R. W., & Zielinski, J. A. Z. (2018). Key-finding by artificial neural networks that learn about key profiles. *Canadian Journal of Experimental Psychology-Revue Canadienne De Psychologie Experimentale, 72*(3), 153–170. https://doi.org/10.1037/cep0000135

De Houwer, J., & Beckers, T. (2002). A review of recent developments in research and theories on human contingency learning. *Quarterly Journal of Experimental Psychology Section B-Comparative and Physiological Psychology, 55*(4), 289–310. https://doi.org/10.1080/02724990244000034

DePristo, M. A., Banks, E., Poplin, R., Garimella, K. V., Maguire, J. R., Hartl, C., … Daly, M. J. (2011). A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature Genetics, 43*(5), 491–498. https://doi.org/10.1038/ng.806

Detambel, M. H. (1955). A test of a model for multiple-choice behavior. *Journal of Experimental Psychology, 49*(2), 97–104. https://doi.org/10.1037/h0046595

Devaney, R. L. (1989). *An introduction to chaotic dynamical systems*. Redwood City, CA: Addison-Wesley.

Douglas, R. J., & Martin, K. A. C. (1991). Opening the grey box. *Trends in Neuroscience, 14*, 286–293. https://doi.org/10.1016/0166-2236(91)90139-L

Doya, K. (2007). *Bayesian brain: Probabilistic approaches to neural coding*. Cambridge, MA: MIT Press.

Dretske, F. (1981). *Knowledge and the flow of information*. Cambridge, MA: MIT Press.

Duch, W., & Jankowski, N. (1999). Survey of neural transfer functions. *Neural Computing Surveys, 2*, 163–212.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. New York, NY: Wiley.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed.). New York, NY: Wiley.

Dupuis, B., & Dawson, M. R. W. (2013a). Differentiating models of associative learning: Reorientation, superconditioning, and the role of inhibition. *Journal of Experimental Psychology: Animal Behavior Processes*. https://doi.org/10.1037/a0032174

Dupuis, B., & Dawson, M. R. W. (2013b). Get out of the corner: Inhibition and the effect of location type and number on perceptron and human reorientation. *Learning & Behavior, 41*(4), 360–378. https://doi.org/10.3758/s13420-013-0111-0

Dutton, J. M., & Starbuck, W. H. (1971). *Computer simulation of human behavior*. New York, NY: Wiley & Sons.

Eagle, A. (2011). *Philosophy of probability: Contemporary readings*. New York, NY: Routledge.

Edwards, W., Lindman, H., & Savage, L. J. (1963). Bayesian statistical inference for psychological research. *Psychological Review, 70*(3), 193–242. https://doi.org/10.1037/h0044139

Eells, E. (1982). *Rational decision and causality*. Cambridge, UK: Cambridge University Press. https://doi.org/10.1017/CBO9781316534823

Egger, M. D., & Miller, N. E. (1962). Secondary reinforcement in rats as a function of information value and reliability of stimulus. *Journal of Experimental Psychology, 64*(2), 97–104. https://doi.org/10.1037/h0040364

Egger, M. D., & Miller, N. E. (1963). When is a reward reinforcing: An experimental study of information hypothesis. *Journal of Comparative and Physiological Psychology, 56*(1), 132–137. https://doi.org/10.1037/h0040744

Einhorn, H. J., & Hogarth, R. M. (1981). Behavioral decision theory: Processes of judgment and choice. *Annual Review of Psychology, 32*, 53–88. https://doi.org/10.1146/annurev.ps.32.020181.000413

Ekeland, I. (1988). *Mathematics and the unexpected*. Chicago, IL: University of Chicago Press.

Erhan, D., Courville, A., & Bengio, Y. (2010). *Understanding representations learned in deep architectures* (Tech. Rep. 1355). Monreal, Canada: Departement d'Informatique et Recherche Operationnelle, Universite de Montreal.

Estes, W. K. (1950). Toward a statistical theory of learning. *Psychological Review, 57*(2), 94–107. https://doi.org/10.1037/0033-295x.101.2.282

Estes, W. K. (1957a). Of models and men. *American Psychologist, 12*(5), 609–617. https://doi.org/10.1037/h0046778

Estes, W. K. (1957b). Theory of learning with constant, variable, or contingent probabilities of reinforcement. *Psychometrika, 22*(2), 113–132. https://doi.org/10.1007/bf02289048

Estes, W. K. (1964). Probability learning. In A. W. Melton (Ed.), *Categories of human learning* (pp. 89–128). New York, NY: Academic Press. https://doi.org/10.1016/B978-1-4832-3145-7.50010-8

Estes, W. K. (1976). The cognitive side of probability learning. *Psychological Review, 83*(1), 37–64. https://doi.org/10.1037/0033-295x.83.1.37

Estes, W. K. (1991). Cognitive architectures from the standpoint of an experimental psychologist. *Annual Review of Psychology, 42*, 1–28. https://doi.org/10.1146/annurev.ps.42.020191.000245

Estes, W. K., Burke, C. J., Atkinson, R. C., & Frankmann, J. P. (1957). Probabilistic discrimination learning. *Journal of Experimental Psychology, 54*(4), 233–239. https://doi.org/10.1037/h0048585

Estes, W. K., & Straughan, J. H. (1954). Analysis of a verbal conditioning situation in terms of statistical learning theory. *Journal of Experimental Psychology, 47*(4), 225–234. https://doi.org/10.1037/h0060989

Farmer, D., Toffoli, T., & Wolfram, S. (Eds.). (1984). *Cellular automata: Proceedings of an interdisciplinary workshop, Los Alamos, New Mexico 87545, USA, March 7–11, 1983*. New York, NY: North-Holland Physics Pub.

Feldman, J. A., & Ballard, D. H. (1982). Connectionist models and their properties. *Cognitive Science, 6*, 205–254. https://doi.org/10.1207/s15516709cog0603_1

Fischer, M. E., Couvillon, P. A., & Bitterman, M. E. (1993). Choice in honeybees as a function of the probability of reward. *Animal Learning & Behavior, 21*(3), 187–195. https://doi.org/10.3758/BF03197981

Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A., 222*, 309–368. https://doi.org/10.1098/rsta.1922.0009

Fodor, J. A. (1968). *Psychological explanation: an introduction to the philosophy of psychology*. New York, NY: Random House.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture. *Cognition, 28*, 3–71. https://doi.org/10.1016/0010-0277(88)90031-5

Foster, M. (1899). *Claude Bernard*. New York, NY: Longmans, Green.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning, 29*(2–3), 131–163. https://doi.org/10.1023/a:1007465528199

Gallant, S. I. (1993). *Neural network learning and expert systems*. Cambridge, MA: MIT Press. https://doi.org/10.7551/mitpress/4931.001.0001

Gardner, H. (1984). *The mind's new science*. New York, NY: Basic Books.

Gardner, R. A. (1957). Probability-learning with two and three choices. *American Journal of Psychology, 70*(2), 174–185. https://doi.org/10.2307/1419319

Garner, W. R. (1975). *Uncertainty and structure as psychological concepts*. Huntington, NY: R. E. Krieger.

Gawehn, E., Hiss, J. A., & Schneider, G. (2016). 6 deep learning in drug discovery. *Molecular Informatics, 35*(1), 3–14. https://doi.org/10.1002/minf.201501008

Ghirlanda, S. (2005). Retrospective revaluation as simple associative learning. *Journal of Experimental Psychology: Animal Learning and Cognition, 31*(1), 107–111. https://doi.org/10.1037/0097-7403-31.1.107

Gigerenzer, G. (1989). *The empire of chance: How probability changed science and everyday life*. Cambridge, UK: Cambridge University Press. https://doi.org/10.1017/CBO9780511720482

Gigerenzer, G. (2000). *Adaptive thinking: Rationality in the real world*. New York, NY: Oxford University Press.

Gigerenzer, G. (2010). *Rationality for mortals: How people cope with uncertainty*. Oxford, UK: Oxford University Press.

Gigerenzer, G., & Hoffrage, U. (1995). How to improve Bayesian reasoning without instruction: Frequency formats. *Psychological Review, 102*(4), 684–704. https://doi.org/10.1037//0033-295x.102.4.684

Gigerenzer, G., & Murray, D. J. (1987). *Cognition as intuitive statistics*. Hillsdale, NJ: Erlbaum.

Gittins, J. C. (1979). Bandit processes and dynamic allocation indexes. *Journal of the Royal Statistical Society Series B-Methodological, 41*(2), 148–177. https://doi.org/10.1111/j.2517-6161.1979.tb01068.x

Gittins, J. C. (1989). *Multi-armed bandit allocation indices*. Chichester, NY: Wiley.

Gleick, J. (1987). *Chaos*. New York, NY: Penguin.

Gluck, M. A., & Bower, G. H. (1988). From conditioning to category learning—An adaptive network model. *Journal of Experimental Psychology-General, 117*(3), 227–247. https://doi.org/10.1037/0096-3445.117.3.227

Gluck, M. A., & Myers, C. (2001). *Gateway to memory: An introduction to neural network modeling of the hippocampus and learning*. Cambridge, MA: MIT Press.

Gluck, M. A., Shohamy, D., & Myers, C. (2002). How do people solve the "weather prediction" task?: Individual variability in strategies for probabilistic category learning. *Learning & Memory, 9*(6), 408–418. https://doi.org/10.1101/lm.45202

Glymour, C. N. (2001). *The mind's arrows: Bayes nets and graphical causal models in psychology*. Cambridge, MA: MIT Press. https://doi.org/10.7551/mitpress/4638.001.0001

Good, I. J. (1983). *Good thinking: The foundations of probability and its applications*. Minneapolis: University of Minnesota Press.

Goodnow, J. J., & Postman, L. (1955). Probability learning in a problem-solving situation. *Journal of Experimental Psychology, 49*(1), 16–22. https://doi.org/10.1037/h0047072

Graf, V., Bullock, D.H., & Bitterman, M. E. (1964). Further experiments on probability-matching in the pigeon. *Journal of the Experimental Analysis of Behavior, 7*(2), 151–157. https://doi.org/10.1901/jeab.1964.7-151

Gray, E. R., Bloomfield, L. L., Ferrey, A., Spetch, M. L., & Sturdy, C. B. (2005). Spatial encoding in mountain chickadees: features overshadow geometry. *Biology Letters, 1*(3), 314–317. https://doi.org/10.1098/rsbl.2005.0347

Green, D. M., & Swets, J. A. (1974). *Signal detection theory and psychophysics*. Huntington, NY: R. E. Krieger.

Gregory, R. L. (1970). *The intelligent eye*. London, UK: Weidenfeld & Nicolson.

Gregory, R. L. (1978). *Eye and brain*. New York, NY: McGraw-Hill.

Grimson, E. (1981). *From images to surfaces*. Cambridge, MA: MIT Press.

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., … Webster, R. (2016). Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA: The Journal of the American Medical Association, 316*(22), 2402–2410. https://doi.org/10.1001/jama.2016.17216

Hacking, I. (2006). *The emergence of probability: A philosophical study of early ideas about probability, induction, and statistical inference* (2nd ed.). Cambridge, UK: Cambridge University Press. https://doi.org/10.1017/CBO9780511817557

Hammond, D. (2003). *The science of synthesis: Exploring the social implications of general systems theory*. Boulder: University Press of Colorado.

Hanson, S. J., & Burr, D. J. (1990). What connectionist models learn: Learning and representation in connectionist networks. *Behavioral and Brain Sciences, 13*, 471–518. https://doi.org/10.1017/S0140525X00079760

Hartman, E., Keeler, J. D., & Kowalski, J. M. (1989). Layered neural networks with Gaussian hidden units as universal approximation. *Neural Computation, 2*, 210–215. https://doi.org/10.1162/neco.1990.2.2.210

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning* (2nd ed.). New York, NY: Springer. https://doi.org/10.1007/978-0-387-84858-7

Hayashi, Y., Setiono, R., & Yoshida, K. (2000). A comparison between two neural network rule extraction techniques for the diagnosis of hepatobiliary disorders. *Artificial Intelligence in Medicine, 20*(3), 205–216. https://doi.org/10.1016/s0933-3657(00)00064-6

Haygood, R. C., & Bourne, L.E . (1965). Attribute and rule learning aspects of conceptual behavior. *Psychological Review, 72*(3), 175–195. https://doi.org/10.1037/h0021802

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical-data. *Machine Learning, 20*(3), 197–243. https://doi.org/10.1007/bf00994016

Helmholtz, H., & Southall, J. P. C. (1856/1962). *Helmholtz's treatise on physiological optics* (Vol. 1–2). New York, NY: Dover.

Heyck, H. (2012). Producing reason. In M. Solovey & H. Cravens (Eds.), *Cold war social science: Knowledge production, liberal democracy, and human nature* (pp. 99–116). New York, NY: Palgrave Macmillan.

Hildreth, E. C. (1983). *The measurement of visual motion*. Cambridge, MA: MIT Press.

Hillix, W. A., & Marx, M. H. (1960). Response strengthening by information and effect in human learning. *Journal of Experimental Psychology, 60*(2), 97–102. https://doi.org/10.1037/h0041634

Hinton, G. E. (1986). *Learning distributed representations of concepts*. Paper presented at the the 8th Annual Meeting of the Cognitive Science Society, Ann Arbor, MI.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation, 14*(8), 1771–1800. https://doi.org/10.1162/089976602760128018

Hinton, G. E. (2007). Learning multiple a layers of representation. *Trends in Cognitive Sciences, 11*(10), 428–434. https://doi.org/10.1016/j.tics.2007.09.004

Hinton, G. E., & Anderson, J. A. (1981). *Parallel models of associative memory*. Hillsdale, NJ: Erlbaum.

Hinton, G. E., McClelland, J., & Rumelhart, D. (1986). Distributed representations. In D. Rumelhart & J. McClelland (Eds.), *Parallel distributed processing* (Vol. 1, pp. 77–109). Cambridge, MA: MIT Press.

Hinton, G. E., Osindero, S., & Teh, Y. (2006). A fast learning algorithm for deep belief nets. *Neural Computation, 18*(7), 1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527

Hohwy, J. (2013). *The predictive mind*. Oxford, UK: Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199682737.001.0001

Holland, J. H. (1992). *Adaptation in natural and artificial systems*. Cambridge, MA: MIT Press. https://doi.org/10.7551/mitpress/1090.001.0001

Holland, J. H. (1995). *Hidden order: How adaptation builds complexity*. Reading, MA: Addison-Wesley.

Holland, J. H. (1998). *Emergence*. Reading, MA: Perseus.

Holland, J. H. (2012). *Signals and boundaries : Building blocks for complex adaptive systems*. Cambridge, MA: MIT Press. https://doi.org/10.7551/mitpress/9412.001.0001

Hornik, M., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*, 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Hosmer, D. W., & Lemeshow, S. (2000). *Applied logistic regression* (2nd ed.). New York, NY: Wiley. https://doi.org/10.1002/0471722146

Humphreys, L. G. (1939). Acquisition and extinction of verbal expectations in a situation analogous to conditioning. *Journal of Experimental Psychology, 25*(3), 294–301. https://doi.org/10.1037/h0053555

Hunt, E. B. (1962). *Concept learning: An information processing problem*. New York, NY: Wiley. https://doi.org/10.1037/13135-000

Hunt, E. B., & Hovland, C. I. (1960). Order of consideration of different types of concepts. *Journal of Experimental Psychology, 59*(4), 220–225. https://doi.org/10.1037/h0039962

Hursch, C. J., Hammond, K. R., & Hursch, J. L. (1964). Some methodological considerations in multiple-cue probability studies. *Psychological Review, 71*(1), 42–60. https://doi.org/10.1037/h0041729

Itti, L., & Baldi, P. (2009). Bayesian surprise attracts human attention. *Vision Research, 49*(10), 1295–1306. https://doi.org/10.1016/j.visres.2008.09.007

Jarvik, M. E. (1951). Probability learning and a negative receny effect in the serial anticipation of alternative symbols. *Journal of Experimental Psychology, 41*(4), 291–297. https://doi.org/10.1037/h0056878

Jeffrey, R. C. (2004). *Subjective probability: The real thing*. Cambridge, UK: Cambridge University Press. https://doi.org/10.1017/CBO9780511816161

Jeffreys, H. (1939). *Theory of probability*. Oxford, UK: Clarendon.

Jenkins, H. M., & Ward, W. C. (1965). Judgment of contingency between responses and outcomes. *Psychological Monographs, 79*(1), 1–17. https://doi.org/10.1037/h0093874

Jones, M., & Love, B. C. (2011). Bayesian fundamentalism or enlightenment? On the explanatory status and theoretical contributions of Bayesian models of cognition. *Behavioral and Brain Sciences, 34*(4), 169–231. https://doi.org/10.1017/s0140525x10003134

Jordan, M. I. (1995). *Why the logistic function? A tutorial discussion on probabilities and neural networks* (Computational Cognitive Science Tech. Rep. No. 9503). Cambridge. MA: MIT.

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research, 4*, 237–285. https://doi.org/10.1613/jair.301

Kahneman, D., & Tversky, A. (1972). Subjective probability: Judgment of representativeness. *Cognitive Psychology, 3*(3), 430–454. https://doi.org/10.1016/0010-0285(72)90016-3

Kahneman, D., & Tversky, A. (1973). Psychology of prediction. *Psychological Review, 80*(4), 237–251. https://doi.org/10.1037/h0034747

Kamin, L. J. (1968). Attention-like processes in classical conditioning. In M. R. Jones (Ed.), *Miami symposium on the prediction of behavior: Aversive stimulation* (pp. 9–32). Miami, FL: University of Miami Press.

Kamin, L. J. (1969). Selective association and conditioning. In N. J. Mackintosh & W. K. Honig (Eds.), *Fundamental issues in associative learning* (pp. 42–64). Halifax, Canada: Dalhousie University Press.

Kauffman, S. A. (1995). *At home in the universe: The search for laws of self-organization and complexity*. New York, NY: Oxford University Press.

Keasar, T., Rashkovich, E., Cohen, D., & Shmida, A. (2002). Bees in two-armed bandit situations: foraging choices and possible decision mechanisms. *Behavioral Ecology, 13*(6), 757–765. https://doi.org/10.1093/beheco/13.6.757

Keri, S., Kelemen, O., Szekeres, G., Bagoczky, N., Erdelyi, R., Antal, A., … Janka, Z. (2000). Schizophrenics know more than they can tell: Probabilistic classification learning in schizophrenia. *Psychological Medicine, 30*(1), 149–155. https://doi.org/10.1017/s0033291799001403

King, W. L. (1966). Learning and utilization of conjunctive and disjunctive classification rules: A developmental study. *Journal of Experimental Child Psychology, 4*(3), 217–231. https://doi.org/10.1016/0022-0965(66)90022-1

Kirk, K. L., & Bitterman, M. E. (1965). Probability-learning by the turtle. *Science, 148*(3676), 1484–1485. https://doi.org/10.1126/science.148.3676.1484

Knowlton, B. J., Squire, L., & Gluck, M. (1994). Probabilistic classification learning in amnesia. *Learning & Memory, 1*(106–120).

Knowlton, B. J., Squire, L. R., Paulsen, J. S., Swerdlow, N. R., Swenson, M., & Butters, N. (1996). Dissociations within nondeclarative memory in Huntington's disease. *Neuropsychology, 10*(4), 538–548. https://doi.org/10.1037/0894-4105.10.4.538

Koski, T., & Noble, J. M. (2009). *Bayesian networks: An introduction*. Chichester, UK: Wiley. https://doi.org/10.1002/9780470684023

Kruschke, J. K. (2011). *Doing Bayesian data analysis: A tutorial With R And BUGS*. Burlington, MA: Academic Press.

Kruschke, J. K., & Liddell, T. M. (2017). The Bayesian new statistics: Hypothesis testing, estimation, meta-analysis, and power analysis from a Bayesian perspective. *Psychonomic Bulletin & Review*, 1–29. https://doi.org/10.3758/s13423-016-1221-4

Ku, H. H., & Kullback, S. (1974). Loglinear models in contingency table analysis. *American Statistician, 28*(4), 115–122. https://doi.org/10.2307/2683334

Kyburg, H. E., & Smokler, H. E. (1964). *Studies in subjective probability*. New York, NY: Wiley.

Labouliere, C. D., Terranova, K., Steinglass, J., & Marsh, R. (2016). Implicit learning on a probabilistic classification task in adults and adolescents with Bulimia Nervosa. *Journal of Psychiatric Research, 77*, 35–41. https://doi.org/10.1016/j.jpsychires.2016.02.004

Langton, C. G. (1995). *Artificial life: An overview*. Cambridge, MA: MIT Press.

Laplace, P. S. (1952). *A philosophical essay on probabilities* (Student's ed.). New York, NY: Dover. (Original work published 1796)

Larochelle, H., Mandel, M., Pascanu, R., & Bengio, Y. (2012). Learning algorithms for the classification restricted Boltzmann machine. *Journal of Machine Learning Research, 13*, 643–669.

Laughlin, P. R., & Jordan, R. M. (1967). Selection strategies in conjunctive, disjunctive, and biconditional concept attainment. *Journal of Experimental Psychology, 75*(2), 188–193. https://doi.org/10.1037/h0024983

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, M. D., & Wagenmakers, E.-J. (2013). *Bayesian cognitive modeling: A practical course*. Cambridge, UK: Cambridge University Press. https://doi.org/10.1017/CBO9781139087759

Leibovic, K. N. (1969). *Information processing in the nervous system*. New York, NY: Springer-Verlag.

Lewandowsky, S. (1993). The rewards and hazards of computer simulations. *Psychological Science, 4*, 236–243. https://doi.org/10.1111/j.1467-9280.1993.tb00267.x

Lindley, D., Tversky, A., & Brown, R. V. (1979). Reconciliation of probability assessments. *Journal of the Royal Statistical Society Series a-Statistics in Society, 142*, 146–180. https://doi.org/10.2307/2345078

Lindley, D. V. (1964). The Bayesian analysis of contingency tables. *Annals of Mathematical Statistics, 35*(4), 1622–1643. https://doi.org/10.1214/aoms/1177700386

Lippmann, R. P. (1989, November). Pattern classification using neural networks. *IEEE Communications*, pp. 47–64. https://doi.org/10.1109/35.41401

Liu, M. C., Shi, J. X., Li, Z., Li, C. X., Zhu, J., & Liu, S. X. (2017). Toward better analysis of deep convolutional neural networks. *IEEE Transactions on Visualization and Computer Graphics, 23*(1), 91–100. https://doi.org/10.1109/tvcg.2016.2598831

Longo, N. (1964). Probability-learning and habit-reversal in the cockroach. *American Journal of Psychology, 77*(1), 29–41. https://doi.org/10.2307/1419269

Luce, R. D. (1959). *Individual choice behavior*. New York, NY: Wiley.

Luce, R. D. (2003). Whatever happened to information theory in psychology? *Review of General Psychology, 7*(2), 183–188. https://doi.org/10.1037/1089-2680.7.2.183

Lunn, D., Jackson, C., Best, N., Thomas, A., & Spiegelhalter, D. (2012). *The BUGS book: A practical introduction to Bayesian analysis*. Boca Raton, FL: CRC Press.

Lv, Y. S., Duan, Y. J., Kang, W. W., Li, Z. X., & Wang, F. Y. (2015). Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems, 16*(2), 865–873. https://doi.org/10.1109/tits.2014.2345663

MacKay, D. (1969). *Information, mechanism and meaning*. Cambridge, MA: MIT Press.

MacKay, D. J. C. (1992). A practical Bayesian framework for backpropagation networks. *Neural Computation, 4*(3), 448–472. https://doi.org/10.1162/neco.1992.4.3.448

MacKay, D. J. C. (1995). Probable networks and plausible predictions: A review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems, 6*(3), 469–505. https://doi.org/10.1088/0954-898x/6/3/011

MacKay, D. J. C. (2003). *Information theory, inference, and learning algorithms*. Cambridge, UK: Cambridge University Press.

Mallat, S. (2016). Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A—Mathematical Physical and Engineering Sciences, 374*(2065). https://doi.org/10.1098/rsta.2015.0203

Mamoshina, P., Vieira, A., Putin, E., & Zhavoronkov, A. (2016). Applications of deep learning in biomedicine. *Molecular Pharmaceutics, 13*(5), 1445–1454. https://doi.org/10.1021/acs.molpharmaceut.5b00982

Mandelbrot, B. B. (1983). *The fractal geometry of nature* (Updated ed.). New York, NY: Freeman.

Marr, D. (1982). *Vision*. San Francisco, CA: Freeman.

McClelland, J. (1998). Connectionist models and Bayesian inference. In M. Oaksford & N. Chater (Eds.), *Rational models of cognition* (pp. 21–53). Oxford, UK: Oxford University Press.

McClelland, J. L., & Rumelhart, D. E. (1986). *Parallel distributed processing, Vol. 2*. Cambridge, MA: MIT Press.

McClelland, J. L., Rumelhart, D. E., & Hinton, G. E. (1986). The appeal of parallel distributed processing. In D. Rumelhart & J. McClelland (Eds.), *Parallel distributed processing* (Vol. 1, pp. 3–44). Cambridge, MA: MIT Press.

McCloskey, M. (1991). Networks and theories: The place of connectionism in cognitive science. *Psychological Science, 2*, 387–395. https://doi.org/10.1111/j.1467-9280.1991.tb00173.x

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics, 5*, 115–133. https://doi.org/10.1007/BF02478259

McGee, V. M. (1971). *Principles of statistics: Traditional and Bayesian*. New York, NY: Appleton-Century-Crofts.

McGrayne, S. B. (2011). *The theory that would not die: How Bayes' rule cracked the enigma code, hunted down Russian submarines, & emerged triumphant from two centuries of controversy* (E-book ed.). New Haven, CT: Yale University Press.

Meadows, D. H., & Wright, D. (2008). *Thinking in systems: A primer*. White River Junction, VT: Chelsea Green.

Medler, D. A. (1998). A brief history of connectionism. *Neural Computing Surveys, 1*, 18–72.

Mellor, D. H. (2005). *Probability: A philosophical introduction*. London, UK: Routledge.

Miller, G. A. (2003). The cognitive revolution: a historical perspective. *Trends in Cognitive Sciences, 7*(3), 141–144. https://doi.org/10.1016/S1364-6613(03)00029-9

Miller, G. A., & Frick, F. C. (1949). Statistical behavioristics and sequences of responses. *Psychological Review, 56*(6), 311–324. https://doi.org/10.1037/h0060413

Miller, R. R. (2006). Challenges facing contemporary associative approaches to acquired behavior. *Comparative Cognition & Behavior Reviews, 1*, 77–93. https://doi.org/10.3819/ccbr.2008.10005

Miller, R. R., Barnet, R. C., & Grahame, N. J. (1995). Assessment of the Rescorla-Wagner model. *Psychological Bulletin, 117*(3), 363–386. https://doi.org/10.1037/0033-2909.117.3.363

Minsky, M. L. (1985). *The society of mind*. New York, NY: Simon & Schuster.

Minsky, M. L., & Papert, S. (1969). *Perceptrons: An introduction to computational geometry*. Cambridge, MA: MIT Press.

Mitchell, M. (1996). *An introduction to genetic algorithms*. Cambridge, MA: MIT Press.

Mohamed, A., Dahl, G. E., & Hinton, G. E. (2012). Acoustic modeling using deep belief networks. *IEEE Transactions on Audio Speech and Language Processing, 20*(1), 14–22. https://doi.org/10.1109/tasl.2011.2109382

Molina, E. C. (1931). Bayes' theorem—An expository presentation. *Bell System Technical Journal, 10*(2), 273–283. https://doi.org/10.1002/j.1538-7305.1931.tb01277.x

Montavon, G., Samek, W., & Muller, K. R. (2018). Methods for interpreting and understanding deep neural networks. *Digital Signal Processing, 73*, 1–15. https://doi.org/10.1016/j.dsp.2017.10.011

Moorhead, I. R., Haig, N. D., & Clement, R. A. (1989). An investigation of trained neural networks from a neurophysiological perspective. *Perception, 18*, 793–803. https://doi.org/10.1068/p180793

Mousavi, N., Siqueira, H., Barros, P., Fernandes, B., & Wermter, S. (2016). Understanding how deep neural networks learn face expressions. In *2016 International Joint Conference on Neural Networks* (pp. 227–234). Piscataway, NJ: IEEE. https://doi.org/10.1109/IJCNN.2016.7727203

Mowrer, O. H. (1960). *Learning theory and the symbolic processes*. New York, NY: Wiley. https://doi.org/10.1037/10772-000

Mozer, M. C., & Smolensky, P. (1989). Using relevance to reduce network size automatically. *Connection Science, 1*, 3–16. https://doi.org/10.1080/09540098908915626

Neal, R. M. (2012). *Bayesian learning for neural networks*. New York, NY: Springer.

Neimark, E. D. (1956). Effets of type of nonreinforcement and number of alternative responses in two verbal conditioning situations. *Journal of Experimental Psychology, 52*(4), 209–220. https://doi.org/10.1037/h0047325

Neisser, U., & Weene, P. (1962). Hierarchies in concept attainment. *Journal of Experimental Psychology, 64*(6), 640–645. https://doi.org/10.1037/h0042549

Nessler, B., Pfeiffer, M., Buesing, L., & Maass, W. (2013). Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLOS Computational Biology, 9*(4). https://doi.org/10.1371/journal.pcbi.1003037

Neumann, J., & Morgenstern, O. (1944). *Theory of games and economic behavior* (60th ed.). Princeton, NJ: Princeton University Press.

Neyman, J. (1937). Outline of a theory of statistical estimation based on the classical theory of probability. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences, 236*(767), 333–380. https://doi.org/10.1098/rsta.1937.0005

Niv, Y., Joel, D., Meilijson, I., & Ruppin, E. (2002). Evolution of reinforcement learning in uncertain environments: A simple explanation for complex foraging behaviors. *Adaptive Behavior, 10*(1), 5–24. https://doi.org/10.1177/1059-712302-010001-01

Oaksford, M., & Chater, N. (1991). Against logicist cognitive science. *Mind & Language, 6*, 1–38. https://doi.org/10.1111/j.1468-0017.1991.tb00173.x

Oaksford, M., & Chater, N. (1998). *Rationality in an uncertain world: Essays on the cognitive science of human reasoning*. Hove, UK: Psychology Press.

Oaksford, M., & Chater, N. (2001). The probabilistic approach to human reasoning. *Trends in Cognitive Sciences, 5*(8), 349–357. https://doi.org/10.1016/S1364-6613(00)01699-5

Oaksford, M., & Chater, N. (2007). *Bayesian rationality: The probabilistic approach to human reasoning*. Oxford, UK: Oxford University Press. https://doi.org/10.1093/acprof:oso/9780198524496.001.0001

Oaksford, M., & Chater, N. (2009). The uncertain reasoner: Bayes, logic, and rationality. *Behavioral and Brain Sciences, 32*(1), 105–120. https://doi.org/10.1017/s0140525x0900051x

Oaksford, M., Chater, N., & Larkin, J. (2000). Probabilities and polarity biases in conditional inference. *Journal of Experimental Psychology-Learning Memory and Cognition, 26*(4), 883–899. https://doi.org/10.1037//0278-7393.26.4.883

Omlin, C. W., & Giles, C. L. (1996). Extraction of rules from discrete-time recurrent neural networks. *Neural Networks, 9*, 41–52. https://doi.org/10.1016/0893-6080(95)00086-0

Papert, S. (1988). One AI or many? *Daedalus, 117*(1), 1–14.

Pavlov, I. P. (1927). *Conditioned reflexes*. New York, NY: Oxford University Press.

Peterson, C. R., & Beach, L. R. (1967). Man as an intuitive statistician. *Psychological Bulletin, 68*(1), 29–46. https://doi.org/10.1037/h0024722

Peterson, C. R., Hammond, K. R., & Summers, D. A. (1965a). Multiple probability learning with shifting weights of cues. *American Journal of Psychology, 78*(4), 660–663. https://doi.org/10.2307/1420932

Peterson, C. R., Hammond, K. R., & Summers, D. A. (1965b). Optimal responding in multiple-cue probability learning. *Journal of Experimental Psychology, 70*(3), 270–276. https://doi.org/10.1037/h0022261

Pfaffelhuber, E. (1972). Learning and information theory. *International Journal of Neuroscience, 3*(2), 83–88. https://doi.org/10.3109/00207457209147016

Pólya, G. (1954). *Mathematics and plausible reasoning: Patterns of plausible inference* (Vol. 2). London, UK: Oxford University Press.

Prigogine, I. (1997). *The end of certainty: Time, chaos, and the new laws of nature*. New York, NY: Free Press.

Pylyshyn, Z. W. (1980). Computation and cognition: Issues in the foundations of cognitive science. *Behavioral and Brain Sciences, 3*(1), 111–132. https://doi.org/10.1017/S0140525X00002053

Pylyshyn, Z. W. (1984). *Computation and cognition*. Cambridge, MA.: MIT Press.

Quastler, H. (1955). Information theory in psychology; problems and methods. In *Proceedings of a conference on the estimation of information flow, Monticello, Illinois, July 5–9, 1954, and related papers*. Glencoe, IL: Free Press.

Ramage, M., & Shipp, K. (2009). *Systems thinkers*. London, UK: Springer. https://doi.org/10.1007/978-1-84882-525-3

Rao, R. P. N., Olshausen, B. A., & Lewicki, M. S. (2002). *Probabilistic models of the brain*. Cambridge, MA: MIT Press. https://doi.org/10.7551/mitpress/5583.001.0001

Reeke, G. N., & Edelman, G. M. (1988). Real brains and artificial intelligence. In S. Graubard (Ed.), *The artificial intelligence debate* (pp. 143–173). Cambridge, MA: MIT Press.

Rescorla, R. A. (1967). Pavlovian conditioning and its proper control procedures. *Psychological Review, 74*(1), 71–80. https://doi.org/10.1037/h0024109

Rescorla, R. A. (1968). Probability of shock in presence and absence of CS in fear conditioning. *Journal of Comparative and Physiological Psychology, 66*(1), 1–5. https://doi.org/10.1037/h0025984

Rescorla, R. A., & Wagner, A. R. (1972). A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. In A. H. Black & W. F. Prokasy (Eds.), *Classical conditioning II: Current research and theory* (pp. 64–99). New York, NY: Appleton-Century-Crofts.

Resnik, M. D. (1987). *Choices: An introduction to decision theory*. Minneapolis: University of Minnesota Press.

Richards, W. (1988). *Natural computation*. Cambridge, MA: MIT Press.

Rock, I. (1983). *The logic of perception*. Cambridge, MA: MIT Press.

Rojas, R. (1996). *Neural networks: A systematic exploration*. Berlin, Germany: Springer. https://doi.org/10.1007/978-3-642-61068-4

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review, 65*(6), 386–408. https://doi.org/10.1037/h0042519

Rosenblatt, F. (1962). *Principles of neurodynamics*. Washington, DC: Spartan Books.

Rozanov, Y. A. (1977). *Probability theory: A concise course* (Rev. ed.). New York, NY: Dover.

Rudas, T. (1998). *Odds ratios in the analysis of contingency tables*. Thousand Oaks, CA: Sage. https://doi.org/10.4135/9781412985338

Rumelhart, D.E., Hinton, G.E., & McClelland, J.L. (1986). A general framework for parallel distributed processing. In D. E. Rumelhart & G. E. Hinton (Eds.), *Parallel Distributed Processing, Volume 1* (pp. 45–76). Cambridge, MA: MIT Press.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*, 533–536. https://doi.org/10.1038/323533a0

Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel distributed processing*. Cambridge, MA: MIT Press.

Sarikaya, R., Hinton, G. E., & Deoras, A. (2014). Application of deep belief networks for natural language understanding. *IEEE-ACM Transactions on Audio Speech and Language Processing, 22*(4), 778–784. https://doi.org/10.1109/taslp.2014.2303296

Savage, L. J. (1951). The theory of statistical decision. *Journal of the American Statistical Association, 46*(253), 55–67. https://doi.org/10.2307/2280094

Savage, L .J. (1954). *The foundations of statistics*. New York, NY: Wiley.

Savage, L. J. (1962). *The foundations of statistical inference*. London, UK: Methuen.

Savage, L. J. (1972). *The foundations of statistics* (2nd Rev. ed.). New York, NY: Dover.

Schrödinger, E. (1945). *What is life? The physical aspect of the living cell*. Cambridge, UK: Cambridge University Press.

Schumacher, M., Rossner, R., & Vach, W. (1996). Neural networks and logistic regression .1. *Computational Statistics & Data Analysis, 21*(6), 661–682. https://doi.org/10.1016/0167-9473(95)00032-1

Seidenberg, M. (1993). Connectionist models and cognitive theory. *Psychological Science, 4*, 228–235. https://doi.org/10.1111/j.1467-9280.1993.tb00266.x

Setiono, R., Baesens, B., & Mues, C. (2011). Rule extraction from minimal neural networks for credit card screening. *International Journal of Neural Systems, 21*(4), 265–276. https://doi.org/10.1142/s0129065711002821

Setiono, R., Thong, J. Y. L., & Yap, C. S. (1998). Symbolic rule extraction from neural networks: An application to identifying organizations adopting IT. *Information & Management, 34*(2), 91–101. https://doi.org/10.1016/s0378-7206(98)00048-2

Shanks, D. R. (1995). *The psychology of associative learning*. Cambridge, UK: Cambridge University Press. https://doi.org/10.1017/CBO9780511623288

Shanks, D. R. (2007). Associationism and cognition: Human contingency learning at 25. *Quarterly Journal of Experimental Psychology, 60*(3), 291–309. https://doi.org/10.1080/17470210601000581

Shannon, C. E. (1938). A symbolic analysis of relay and switching circuits. *Transactions of the American Institute of Electrical Engineers, 57*, 713–723. https://doi.org/10.1109/T-AIEE.1938.5057767

Shannon, C. E., & Weaver, W. (1949). *The mathematical theory of communication*. Urbana: University of Illinois Press.

Shapin, S. (1996). *The scientific revolution*. Chicago, IL: University of Chicago Press. https://doi.org/10.7208/chicago/9780226750224.001.0001

Siegel, S., & Allan, L. G. (1996). The widespread influence of the Rescorla-Wagner model. *Psychonomic Bulletin & Review, 3*(3), 314–321. https://doi.org/10.3758/bf03210755

Siegelmann, H. T. (1999). *Neural networks and analog computation: Beyond the Turing limit*. Boston, MA: Birkhauser. https://doi.org/10.1007/978-1-4612-0707-8

Siegelmann, H. T., & Sontag, E. D. (1991). Turing computability with neural nets. *Applied Mathematics Letters, 4*, 77–80. https://doi.org/10.1016/0893-9659(91)90080-F

Siegelmann, H. T., & Sontag, E. D. (1995). On the computational power of neural nets. *Journal of Computer and System Sciences, 50*, 132–150. https://doi.org/10.1006/jcss.1995.1013

Silver, N. (2012). *The signal and the noise: Why so many predictions fail—But some don't*. New York, NY: Penguin Press.

Simon, H. A. (1982). *Models of bounded rationality*. Cambridge, MA: MIT Press.

Simon, H. A., Egidi, M., & Marris, R. L. (1995). *Economics, bounded rationality and the cognitive revolution*. Aldershot, England: E. Elgar.

Skinner, B. F. (1938). *The behavior of organisms*. New York, NY: Appleton-Century-Crofts.

Smolensky, P. (1988). On the proper treatment of connectionism. *Behavioral and Brain Sciences, 11*, 1–74. https://doi.org/10.1017/S0140525X00052432

Snow, C. E., & Rabinovitch, M. S. (1969). Conjunctive and disjunctive thinking in children. *Journal of Experimental Child Psychology, 7*(1), 1–9. https://doi.org/10.1016/0022-0965(69)90080-0

Sovrano, V. A., Bisazza, A., & Vallortigara, G. (2007). How fish do geometry in large and in small spaces. *Animal Cognition, 10*(1), 47–54. https://doi.org/10.1007/s10071-006-0029-4

Stone, L. D. (1975). *Theory of optimal search*. New York, NY: Academic Press.

Sutskever, I., & Hinton, G. E. (2008). Deep, narrow sigmoid belief networks are universal approximators. *Neural Computation, 20*(11), 2629–2636. https://doi.org/10.1162/neco.2008.12-07-661

Sutton, R. S., & Barto, A. G. (1981). Toward a modern theory of adaptive networks: Expectation and prediction. *Psychological Review, 88*(2), 135–170. https://doi.org/10.1037/0033-295X.88.2.135

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning*. Cambridge, MA: MIT Press.

Taha, I. A., & Ghosh, J. (1999). Symbolic interpretation of artificial neural networks. *IEEE Transactions on Knowledge and Data Engineering, 11*(3), 448–463. https://doi.org/10.1109/69.774103

Temperley, D. (2004). Bayesian models of musical structure and cognition. *Musicae Scientiae, 8*(2), 175–205. https://doi.org/10.1177/102986490400800204

Thorndike, E. L. (1901). Animal intelligence: An experimental study of the associative processes in animals. *Psychological Review Monograph Supplement, 2*, 1–109.

Toffoli, T., & Margolus, N. (1987). *Cellular automata machine : A new environment for modeling*. Cambridge, MA: MIT Press.

Tolman, E. C., & Brunswik, E. (1935). The organism and the causal texture of the environment. *Psychological Review, 42*, 43–77. https://doi.org/10.1037/h0062156

Tucker, L. R. (1964). A suggested alternative formulation in the developments by Hursch, Hammond, and Hursch and by Hammond, Hursch, and Todd. *Psychological Review, 71*(6), 528–530. https://doi.org/10.1037/h0047061

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty—Heuristics and biases. *Science, 185*(4157), 1124–1131. https://doi.org/10.1126/science.185.4157.1124

Tversky, A., & Kahneman, D. (1983). Extensional versus intuitive reasoning: The conjunction fallacy in probability judgment. *Psychological Review, 90*(4), 293–315. https://doi.org/10.1037/0033-295x.90.4.293

Twyman, A., Friedman, A., & Spetch, M. L. (2007). Penetrating the geometric module: Catalyzing children's use of landmarks. *Developmental Psychology, 43*(6), 1523–1530. https://doi.org/10.1037/0012-1649.43.6.1523

Ullman, S. (1979). *The interpretation of visual motion*. Cambridge, MA: MIT Press.

Van Gelder, T. (1991). What is the "D" in "PDP"? A survey of the concept of distribution. In W. Ramsey, S. P. Stich, & D. E. Rumelhart (Eds.), *Philosophy and connectionist theory* (pp. 33–59). Hillsdale, NJ: Erlbaum.

Van Hamme, L. J., & Wasserman, E. A. (1994). Cue competition in causality judgments: The role of nonpresentation of compound stimulus elements. *Learning and Motivation, 25*(2), 127–151. https://doi.org/10.1006/lmot.1994.1008

VanLehn, K. (1991). *Architectures for intelligence*. Hillsdale, NJ: Erlbaum.

von Bertalanffy, L. (1933). *Modern theories of development: An introduction to theoretical biology* (J. H. Woodger, Trans.). London, UK: Oxford University Press.

von Bertalanffy, L. (1952). *Problems of life: An evaluation of modern biological thought*. London, UK: Watts.

von Bertalanffy, L. (1968a). *General system theory: Foundations, development, applications*. New York, NY: G. Braziller.

von Bertalanffy, L. (1968b). *Organismic psychology and systems theory*. Worcester, MA: Clark University Press.

Waldrop, M. M. (1992). *Complexity: The emerging science at the edge of order and chaos*. New York, NY: Simon & Schuster.

Walkenbach, J., & Haddad, N. F. (1980). The Rescorla-Wagner theory of conditioning—A review of the literature. *Psychological Record, 30*(4), 497–509. https://doi.org/10.1007/BF03394701

Ward, W. C., & Jenkins, H. M. (1965). The display of information and the judgment of contingency. *Canadian Journal of Psychology, 19*(3), 231–241. https://doi.org/10.1037/h0082908

Warren, H. C. (1921). *A history of the association psychology*. New York, NY: Scribners. https://doi.org/10.1037/11522-000

Wasserman, E. A., Dorner, W. W., & Kao, S. F. (1990). Contributions of specific cell information to judgments of interevent contingency. *Journal of Experimental Psychology—Learning Memory and Cognition, 16*(3), 509–521. https://doi.org/10.1037//0278-7393.16.3.509

Wasserman, E. A., Elek, S. M., Chatlosh, D. L., & Baker, A. G. (1993). Rating causal relations: The role of probability in judgments of response outcome contingency. *Journal of Experimental Psychology: Learning Memory and Cognition, 19*(1), 174–188. https://doi.org/10.1037//0278-7393.19.1.174

Wasserstein, R. L., & Lazar, N. A. (2016). The ASA's statement on *p*-values: Context, process, and purpose. *American Statistician, 70*(2), 129–131. https://doi.org/10.1080/00031305.2016.1154108

Weiss, P. (1967). Uniquness axiom of uncertainty: A contribution to a concept of useful informaton. *Kybernetik, 4*(2), 65–65. https://doi.org/10.1007/bf00292173

Wells, H. (1963). Effects of transfer and problem structure in disjunctive concept formation. *Journal of Experimental Psychology, 65*(1), 63–69. https://doi.org/10.1037/h0047025

Werbos, P. J. (1994). *The roots of backpropagation: From ordered derivatives to neural networks and political forecasting*. New York, NY: Wiley.

West, S. A. (1979). Upper and lower probability inferences for the logistic function. *Annals of Statistics, 7*(2), 400–413. https://doi.org/10.1214/aos/1176344623

Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. *Institute of Radio Enginners, Wester Electronic Show and Convention, Convention Record, Part 4*, 96–104. https://doi.org/10.21236/AD0241531

Wiener, N. (1948). *Cybernetics: Or control and communciation in the animal and the machine*. Cambridge, MA: MIT Press.

Wiener, N. (1950). *The human use of human beings: Cybernetics and society*. Boston, MA: Houghton Mifflin.

Witnauer, J. E., & Miller, R. R. (2011). The role of within-compound associations in learning about absent cues. *Learning & Behavior, 39*(2), 146–162. https://doi.org/10.3758/s13420-010-0013-3

Woodward, P. (2012). *Bayesian analyisis made simple: An Excel GUI For WinBUGS*. Boca Raton, FL: CRC Press.

Woolf, B. (1957). The log likelihood ratio test (the G-test): Methods and tables for tests of heterogeneity in contingency tables. *Annals of Human Genetics, 21*(4), 397–409. https://doi.org/10.1111/j.1469-1809.1972.tb00293.x

Wystrach, A., & Beugnon, G. (2009). Ants learn geometry and features. *Current Biology, 19*(1), 61–66. https://doi.org/10.1016/j.cub.2008.11.054

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer vision—ECCV 2014, Pt I* (Vol. 8689, pp. 818–833). Berlin, Germany: Springer Verlag. https://doi.org/10.1007/978-3-319-10590-1_53