LETTER

Traditional Perceptrons Do Not Produce the Overexpectation Effect

Michael R. W. Dawson and Marcia L. Spetch

Department of Psychology, University of Alberta
Edmonton, Alberta, Canada T6G 2E9
E-mail: mdawson@ualberta.ca

(Submitted on December 14, 2004)

Abstract -- Perceptrons are typically viewed as being an artificial neural network that embodies the Rescorla-Wagner model of learning. One of the important properties of the Rescorla-Wagner model was its prediction of the overexpectation effect. However, we show below that a typical perceptron is not capable of generating this effect. This result brings into question assumed relationships between artificial neural networks and models of animal learning.

Keywords – Overexpectation effect, perceptrons, activation functions, learning theory

1. Introduction

One of the most theoretically important effects in associative learning is the overexpectation effect. The effect is produced when two conditioned stimuli (CSs), A and X, are independently paired with a given unconditioned stimulus (US) until asymptotic learning occurs, and then A and X are presented in compound and paired with the US. The result is that responding to A or X is reduced following AX-US pairings relative to a control condition in which no compound training is given. This result is intuitively surprising because A and X apparently lost associative strength despite continued reinforcement during the compound training. Nevertheless, overexpectation effects have been found in studies on Pavlovian fear conditioning in rats [1-3], appetitive conditioning in rats [4, 5] and autoshaping with pigeons [6]. Moreover, the effect can be reversed by naloxone injections, suggesting that it is modulated by the opioid system [7].

Although counter-intuitive, the overexpectation effect was predicted by the Rescorla-Wagner model [8]. Indeed, this effect, together with other effects such as blocking [9], provided strong support for the assumption that the effect of reinforcement on learning is relative to the organism's expectations of reinforcement. The notion that learning depends on the discrepancy between anticipated and obtained reinforcement is a key assumption of the Rescorla-Wagner model. This assumption is formalized in the Rescorla-Wagner equation, which defines the change in associative strength (ΔV_A) of some CS_A as:

$$\Delta V_A = k(\lambda - V_{SUM}) \tag{1}$$

In this equation, k is a learning rate parameter (e.g., reflecting the salience of the stimuli), λ is the maximum associative strength that can be supported by the UCS, and V_{SUM} is the total amount of associative strength for all stimuli that are present. The change in associative strength of any CS presented on a trial can have a positive or negative value. In the case of overexpectation, A and X are both trained to asymptote in the first phase and hence V for each would approximate λ. When A and X are subsequently presented in compound, their associative strengths are summed and hence V_{sum} exceeds λ and the change will be negative. Conceptually, the two stimuli together over-predict the US. Accordingly, decreases in associative strength will occur until V_{sum} equals λ. This will result in reduced responding to either A or X alone in the test phase. Although subsequent theories have been able to account for this overexpectation effect (see [1]), the direct prediction of this effect by the Rescorla-Wagner model has been theoretically important.

The Rescorla-Wagner model's prediction of the overexpectation effect is also theoretically important toother formal accounts of learning. There has been a growing interest in using artificial neural networks to study associative learning [10]. In particular, one type of artificial neural network, a perceptron, is viewed as being equivalent to the Rescorla-Wagner model [11, 12]. This paper examines this putative equivalence in the light of empirical data. In particular, we show below that a typical perceptron is not predicted to generate the overexpectation effect. This raises important questions about the relationships between artificial neural networks and theories of animal learning.

2. Perceptrons, Associative Learning, and Overexpectation

The perceptron is a simple artificial neural network (ANN). It consists of a set of input units that are used to represent stimuli. For example, one input unit could be associated with conditioned stimulus 1 (CS_1), and would be turned on if CS_1 was present, and turned off if CS_1 was absent. Connections with modifiable weights permit the input units to send signals to one or more output units. As a result, a connection weight is analogous to the associative strength between a conditioned stimulus and a conditioned response.

An output unit in a perceptron sums all of its incoming signals to compute its net input. It then applies a nonlinear transformation -- the activation function -- to the net input. For example, when this nonlinear transformation is defined by the step function, the net input is compared to a threshold. If it exceeds the threshold, then the output unit generates a response of 1. Otherwise, it generates a response of 0.

Other activation functions, such as the sigmoid-shaped logistic equation, can also be used to define the nonlinear transformation of net input:

$$f(net_i) = 1 / (1 + \exp(-net_i + \theta_j)) \tag{2}$$

In this equation, $f(net_i)$ is the activation being calculated for output unit i, net_i is the net input for that output unit (i.e., the sum of the weighted signals from the input units), and θ_j is called the bias of the output unit. When the net input to the logistic equation is equal to the bias (i.e., equal to θ_j), the activity that is generated is equal to 0.5. Because of this, it is typical to consider the bias of the logistic activation function as being analogous to the threshold of the step function.

Perceptrons are trained via supervised learning, in which a stimulus pattern is presented to the input units, and the perceptron responds using its existing connection weights. Error is calculated by comparing the observed output and the desired response, and is then used to adjust the connection weights to ensure that error is reduced. This process is repeated for another stimulus, and continued iteratively until the perceptron responds correctly to every stimulus.

The general learning rule for a perceptron is

$$\Delta w_{ij} = \eta(t_j - a_j)\, a_i \tag{3}$$

where Δw_{ij} is the change in the weight of the connection between input unit i and output unit j, η is a learning rate that will ordinarily range between 0 and1, $(t_j - a_j)$ is the error calculated for output unit j, and a_i is the activity of input unit i. If the error term indicates that an output unit has turned on when it should have turned off, then Eq.(3) will adjust the weight to decrease net input. If the error term indicates that an output unit has turned off when it should have turned on, then Eq.(3) will modify the weight in such a way to increase net input. If the error term is zero, then Eq.(3) will not change the weight.

When the activation function is continuous (e.g., the logistic equation), gradient descent learning can be used to modify weights. This changes connection weights in such a way that output unit error is reduced as quickly as possible by multiplying output unit error by the first derivative of the activation function ($f'(net_j)$ [13]. This requires an elaborated statement of error for output unit j, represented as δ_j. The first derivative of the logistic equation, i.e., Eq.(2), is equal to the value $a_j\,(1 - a_j)$. So, the equation for δ_j when the logistic function is used is:

$$\delta_j = (t_j - a_j)\, f'(net_j) \;= (t_j - a_j)\, a_j\, (1 - a_j) \tag{4}$$

A gradient descent learning rule for a perceptron that uses the logistic activation function is defined by inserting the error term from Eq.(4) into the generic learning rule that was given in Eq.(3)

$$\Delta w_{ij} = \eta\, \delta_j\, a_i = \eta\, (t_j - a_j)\, a_j\, (1 - a_j)\, a_i \tag{5}$$

The bias (θ_j) of output unit j can also be modified with a variation of Eq.(5).

By specifying learning rules like those in Eqs.(1) and (3), one can explore the relationship between them and other accounts of learning. For example, consider the Rescorla-Wagner model in Eq.(1). It is clear that its structure is very similar to that of the generic learning rule in Eq.(3). Indeed, many researchers have shown that the two equations are equivalent [12, 14, 15]. This is usually proven by labeling the various components of a perceptron to relate them to the Rescorla-Wagner model, and by then replacing the terms in Eq.(3) with the new labels. This translates Eq.(3) into the Rescorla-Wagner rule.

Given existing proofs that a perceptron that is trained with a supervised learning rule is an instance of the Rescorla-Wagner model, it should be no surprise that a perceptron can easily simulate an experiment in which an animal is trained to respond to two different stimuli, A and X, by using two input units, each representing the presence of one of the two stimuli.

However, if the perceptron uses a monotonic activation function like the step function or the logistic equation, then it will not generate the overexpectation effect. In Phase 1 of training, the network will learn to respond to A and to X, and learn not to respond in their absence. This suggests that the individual connection weights from both A and X to the output unit will each be substantially larger than the bias, which will lead to a signal through one of these connection weights to be sufficiently strong to turn the output unit on. In Phase 2 of training, the network will be presented A and X together, producing an even stronger signal to the output unit, which will again cause it to (correctly) turn on. Because it will generate this correct response – and correctly fail to respond when no signal is sent through the connection weights – the network's weights and bias will not be modified any further. Therefore the perceptron's response to the individual stimuli will not decrease, and the overexpectation effect will not be evident.

3. Simulation 1: Logistic Activation Function

The preceding section argued that a perceptron with an activation function that has one threshold (e.g., step function, logistic equation) will not generate the overexpectation effect. The simulation described below tests this prediction.

Three different training sets were created to represent the different stimulus sets used in the overexpectation paradigm, and are provided in Table 1. In Phase 1, the networks were trained to respond to the presence of A or X, and to not respond when both stimuli were absent (~AX). In Phase 2, the networks were trained to respond to the presence of both stimuli (AX), and not to respond to ~AX. At the start of Phase 2, the weights of the network were those that resulted from the first phase in training. In Phase 3, the network produced by Phase 2 was tested on all of the possible stimuli that could be presented to it (A, X, AX, ~AX) without undergoing additional training.

Table 1. The output unit activations generated by two different types of perceptrons at different points in the overexpectation paradigm.

					Responses of Perceptrons						
					Logistic Activation Function			**Gaussian Activation Function**			
Epochs					696	-	-	93	-	+2	-
	Stimulus	**Input 1**	**Input 2**	**Output**							
Phase 1	~AX	0	0	0	0.10			0.10			
	A	1	0	1	0.94			1.00			
	X	0	1	1	0.94			1.00			
Phase 2	~AX	0	0	0		0.10			0.10	0.02	
	AX	1	1	1		1.00			0.12	0.97	
Phase 3	~AX	0	0	-			0.10				0.02
	A	1	0	-			0.94				0.45
	X	0	1	-			0.94				0.46
	AX	1	1	-			1.00				0.97

A traditional perceptron was used in this simulation, and it consisted of two input units and one output unit. The logistic equation was used as the activation function in the output unit. The perceptron was trained using the gradient descent rule [16]. At the start of Phase 1, the connection weights were assigned random values in the range from –0.1 to +0.1. The bias of the activation function was assigned an initial value of 0. The learning rule was used to modify both connection weights and the bias after the presentation of each pattern. The learning rate was 0.5, and no momentum was used. The perceptron was trained in a series of epochs, where each epoch involved presenting every pattern in a training set once. The order of pattern presentation was randomized every epoch. The network was trained in Phase 1 until it generated a "hit" for every pattern. A hit was operationalized as an activation of 0.90 or higher when the desired activation was 1.00 and as an activation of 0.10 or lower when the desired activation was 0.00. The network generated a "hit" for each of the three Phase 1 patterns after 696 epochs of training.

In Phase 2, the perceptron was trained on the second set of patterns (see Table 1), but the starting weights were those produced by Phase 1 training. This perceptron was able to generate "hits" for the Phase 2 stimuli without any additional training. The responses of the network were then observed when all four possible stimuli were presented to it in Phase 3.

The results of training this network are also presented in Table 1. It is clear that the perceptron did not generate the overexpectation effect, as was predicted from our consideration of the general characteristics of this type of network. After learning to respond to A and X as individual stimuli, this perceptron was able to correctly respond to the compound stimulus AX without any additional training. As a result, at the end of the paradigm the perceptron generated equally strong responses to A, X, and AX. This is contrary to the notion that this perceptron is a valid embodiment of the Rescorla-Wagner rule.

4. Simulation 2: Gaussian Activation Function

The first simulation produced results that were consistent with the prediction that traditionally-defined perceptrons will not generate the overexpectation effect. However, this does not mean that one should abandon attempts to relate perceptron-like architectures to models of animal learning. This is because other variations of the perceptron are available. One is not restricted to using the logistic equation as an activation function [17]. If an output unit used a different activation function that had two thresholds – a lower threshold for turning activation on, and a higher threshold for turning activation off – then it should produce overexpectation.

One example of such an activation function is the Gaussian equation used by Dawson and Schopflocher [23] to create networks of value units. Their activation function was:

$$G(net_i) = \exp\,(-\pi(net_i - \mu_j)^2) \qquad (6)$$

In this equation, $G(net_i)$ is the activation being calculated for output unit j, net_i is the net input for that output unit, and μ_j is the mean of the Gaussian. When the net input to the output unit is equal to the mean (i.e., equal to μ_j), the activity that is generated is equal to 1.0. As a result, μ_j can be thought of as being similar to the bias of the logistic or the threshold of the step function. This activation function can be used in a perceptron; when this is done, the perceptron can be trained using a variation of the learning rule that was given above in Eq.(5) [18]. The variation of the learning rule requires a different expression of error (Eq.(4)) that builds upon the derivative of the Gaussian equation.

An output unit that employs Eq.(6) will only respond to a narrow range of net inputs; if the net input is either too small or too large, then the output unit will fail to respond. This type of function has been used to explore some issues in animal learning. For instance, [19] used this activation function to model attentional effects in associative learning [20-22], and found that this network generated results that were a better fit to results from animals in a patterning task than did a network that used a logistic activation function.

A perceptron that employs a function like the Gaussian is much more likely to generate the overexpectation effect. This is because when the perceptron is presented the combined stimuli in Phase 2 of training, the net input should be high enough to exceed the net input range that causes the output unit to turn on. As a result, the output unit will turn off, and – unlike the case in Simulation 1 – more training will be required. The purpose of the second simulation was to test this prediction, and to see whether the resulting behavior of the network resembled the overexpectation effect.

The method used in the second simulation was identical to that used in Simulation 1, with the exception that the output unit used the Gaussian activation function. Therefore in Simulation 2 the perceptron was trained with a variant of the gradient descent rule [23] that is specialized to work with the Gaussian activation function, but belongs to the same general family of learning rules that was used to train the other perceptron (e.g., [13]). In Phase 1, its connection weights and bias (μ_j) were randomized in the same fashion as before. It was trained with a learning rate of 0.1, with no momentum. It too was trained until a "hit" was generated to every pattern, which occurred after 93 epochs. At the start of Phase 2, it did not generate a hit to every pattern, but did so after only 2 additional epochs of training. The responses of this final network to the four possible stimuli were then observed in Phase 3.

The results of Simulation 2 are also provided in Table 1. It can be seen that the second perceptron did generate the overexpectation effect. At the start of Phase 2, the network failed to respond to the compound stimulus AX. This is because the net input from this compound stimulus was larger than the net input produced by either stimulus alone (i.e., A or X during Phase 1). The net input for AX was large enough to be outside of the "tuning range" of the Gaussian activation function, leading to low output unit activation. However, it only took the network 2 additional epochs of training to correct this problem. Phase 3 testing revealed that one of the effects of this small additional amount of training was to attenuate the network's responses to A and to X. It can be seen that training on the compound stimulus resulted in the network's responses to these two stimuli to be less than half of its response to AX.

This simulation clearly shows that a perceptron using a Guassian activation function, in contrast to one using a traditional activation function, can generate the overexpectation effect. However, this result does not suggest that the perceptron with a Guassian activation function provides the best model of conditioning effects, nor does it support an argument against other computational models of learning [24-26]. Rather, this simulation illustrates how the activation function alters the fit of the simulation to empirical predictions of the Rescorla-Wagner model. Whether this particular model will accurately fit other empirical constraints is a separate issue that may be addressed in future research.

5. Discussion

In relating artificial neural networks to models of animal learning, it has been argued that artificial neural networks provide an account of how formal theories of animal learning might be translated into biologically plausible implementations (e.g., Shanks, 1995). This view requires that artificial neural networks are consistent with accounts of learning at more abstract levels – that is, that they be formally equivalent.

The current results bring this assumption into question. If, for instance, standard perceptrons are formally equivalent to the Rescorla-Wagner model, then they should generate all of the effects predicted by this model. However, we have shown that this is not the case for the overexpectation effect.

This is not to say that the overexpectation effect cannot be generated by variants of the traditional perceptron architecture. For instance, in Simulation 2 we demonstrated that a perceptron that used a nonmonotonic activation can produce the effect. However, there is little comfort in this kind of demonstration to researchers who wish to relate perceptrons to models of animal learning. This is because the comparison between the two types of models is generally mute about the activation function employed by the network. The fact that when empirical data is brought to bear the nature of the activation function becomes critical suggests that the formal relationship between artificial neural networks and associative learning needs to be re-evaluated. Further research is clearly required to determine precise relationships between particular forms of artificial neural networks and particular learning theories.

Acknowledgements : The research reported in this manuscript was supported by NSERC and SSHRC research grants awarded to MRWD, and by an NSERC research grant to MLS . Author inquiries should be addressed to Dr. Michael Dawson, Department of Psychology, University of Alberta, Edmonton, AB, CANADA T6G 2E9 (www.bcp.psych.ualberta.ca/~mike/)

References

[1] A. P. Blaisdell, J. C. Denniston, and R. R. Miller, "Recovery from the overexpectation effect: Contrasting performance-focused and acquisition-focused models of retrospective revaluation," *Animal Learning & Behavior*, vol. 29, pp. 367-380, 2001.

[2] E. F. Kremer, "Rescorla-Wagner Model - Losses in Associative Strength in Compound Conditioned Stimuli," *Journal of Experimental Psychology-Animal Behavior Processes*, vol. 4, pp. 22-36, 1978.

[3] R. A. Rescorla, "Reduction in effectiveness of reinforcement after prior excitatory conditioning," *Learning and Motivation*, vol. 1, pp. 372-381, 1970.

[4] K. M. Lattal and S. Nakajima, "Overexpectation in appetitive Pavlovian and instrumental conditioning," *Animal Learning & Behavior*, vol. 26, pp. 351-360, 1998.

[5] R. A. Rescorla, "Summation and overexpectation with qualitatively different outcomes," *Animal Learning & Behavior*, vol. 27, pp. 50-62, 1999.

[6] Y. Khallad and J. Moore, "Blocking, unblocking, and overexpectation in autoshaping with pigeons," *Journal*

of the Experimental Analysis of Behavior, vol. 65, pp. 575-591, 1996.

[7] G. P. McNally, M. Pigg, and G. Weidemann, "Blocking, unblocking, and overexpectation of fear: A role for opioid receptors in the regulation of Pavlovian association formation," *Behavioral Neuroscience*, vol. 118, pp. 111-120, 2004.

[8] R. A. Rescorla and A. R. Wagner, "A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement," in *Classical Conditioning II: Current Research And Theory*, A. H. Black and W. F. Prokasy, Eds. New York, NY: Appleton-Century-Crofts, 1972, pp. 64-99.

[9] L. J. Kamin, "Attention-like processes in classical conditioning," in *Miami symposium on the prediction of behavior: Aversive stimulation*, M. R. Jones, Ed. Miami: University of Miami Press, 1968, pp. 9-32.

[10]D. R. Shanks, *The Psychology Of Associative Learning*. Cambridge, UK: Cambridge University Press, 1995.

[11]J. M. Pearce, *Animal Learning And Cognition: An Introduction*. East Sussex: Psychology Press, 1997.

[12]R. S. Sutton and A. G. Barto, "Toward a modern theory of adaptive networks: Expectation and prediction," *Psychological Review*, vol. 88, pp. 135-170, 1981.

[13]M. R. W. Dawson, *Minds And Machines : Connectionism And Psychological Modeling*. Malden, MA: Blackwell Pub., 2004.

[14]D. Danks, "Equilibria of the Rescorla-Wagner model," *Journal of Mathematical Psychology*, vol. 47, pp. 109-121, 2003.

[15]M. A. Gluck and C. Myers, *Gateway to memory : an introduction to neural network modeling of the hippocampus and learning*. Cambridge, Mass.: MIT Press, 2001.

[16]D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, vol. 1, D. E. Rumelhart and G. E. Hinton, Eds. Cambridge, MA: MIT Press, 1986, pp. 318-362.

[17]W. Duch and N. Jankowski, "Survey of neural transfer functions," *Neural Computing Surveys*, vol. 2, pp. 163-212, 1999.

[18]M. R. W. Dawson, *Connectionism : a hands-on approach*, 1st ed. Oxford, UK ; Malden, MA: Blackwell Pub., 2005.

[19]V. Yaremchuk, L. R. Willson, M. L. Spetch, and M. R. W. Dawson, "The implications of null patterns and output unit activation functions on simulation studies of learning: A case study of patterning," *Learning and Motivation*, vol. 36, pp. 88-103, 2005.

[20]J. K. Krushke, "Toward a unified model of attention in associative learning," *Journal of Mathematical Psychology*, vol. 45, pp. 812-863, 2001.

[21]J. K. Krushke, "Attention in learning," *Current Directions In Psychological Science*, vol. 12, pp. 171-175, 2003.

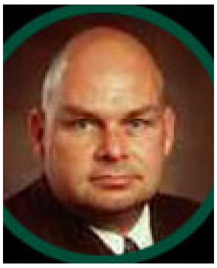
[22]N. J. Mackintosh, "A theory of attention: Variation in the associability of stimuli with reinforcement," *Psychological Review*, vol. 82, pp. 276-298, 1975.

[23]M. R. W. Dawson and D. P. Schopflocher, "Modifying the generalized delta rule to train networks of nonmonotonic processors for pattern classification," *Connection Science*, vol. 4, pp. 19-31, 1992.

[24]I. P. L. McLaren and N. J. Mackintosh, "An elemental model of associative learning: I. Latent inhibition and perceptual learning," *Animal Learning & Behavior*, vol. 28, pp. 211-246, 2000.

[25]I. P. L. McLaren and N. J. Mackintosh, "Associative learning and elemental representation: II. Generalization and discrimination," *Animal Learning & Behavior*, vol. 30, pp. 177-200, 2002.

[26]J. M. Pearce, "Evaluation and development of a connectionist theory of configural learning," *Animal Learning & Behavior*, vol. 30, pp. 73-95, 2002.



Michael R.W. Dawson received his Ph.D. in psychology from the University of Western Ontario in 1986, and is currently a full professor in the Psychology Department at the University of Alberta. His research interests include pure and applied research on artificial neural networks and the relationship of this research to empirical and theoretical issues in Cognitive Science. (Homepage: http://www.bcp.psych.ualberta.ca/~mike/)

Marcia L. Spetch received her Ph.D. from the University of British Columbia in 1982 and is currently at full professor in the Department of Psychology at the University of a Alberta. Her research interests are in learning and comparative cognition with a current focus on spatial learning and object recognition. (Homepage: http://www.psych.ualberta.ca/~mspetch/spetchm.htm)

.