Density Plots of Hidden Value Unit Activations Reveal Interpretable Bands

ISTVÁN S. N. BERKELEY, MICHAEL R. W. DAWSON, DAVID A. MEDLER, DON P. SCHOPFLOCHER & LORRAINE HORNSBY

(Received for publication July 1994; revised paper accepted December 1994)

A particular backpropagation network, called a network of value units, was trained to detect problem type and validity of a set of logic problems. This network differs from standard networks in using a Gaussian activation function. After training was successfully completed, jittered density plots were computed for each hidden unit, and used to represent the distribution of activations produced in each hidden unit by the entire training set. The density plots revealed a marked banding. Further analysis revealed that almost all of these bands could be assigned featural interpretations, and played an important role in explaining how the network classified input patterns. These results are discussed in the context of other techniques for analyzing network structure, and in the context of other parallel distributed processing architectures.

KEYWORDS: Connectionism, backpropagation, structure, interpretation, symbolic logic.

1. Introduction

Parallel distributed processing (PDP) models have been developed for a diverse range of phenomena, as a survey of almost any journal related to cognitive science will show. As a result, it has been suggested that connectionism represents a potential paradigm shift for the computational study of intelligence (e.g. Schneider, 1987). Unfortunately, there is growing concern that possible PDP contributions to cognitive science will be severely limited by the fact that trained networks are extremely difficult to interpret (e.g. Dawson & Shamanski, 1994; Dawson *et al.*, 1993; McCloskey, 1991; Robinson, 1992). This paper addresses this issue by describing some surprising behaviour of hidden units in a new connectionist architecture. This behaviour markedly facilitates our ability to interpret the structure of PDP networks.

I. S. N. Berkeley, Biological Computation Project, Department of Philosophy, University of Alberta, Edmonton, Alberta, Canada T6G 2E9. M. R. W. Dawson, D. A. Medler, D. P. Schopflocher and L. Hornsby, Biological Computation Project, Department of Psychology, University of Alberta, Edmonton, Alberta, Canada T6G 2E9. Correspondence to M. R. W. Dawson. E-mail: mike@psych.ualberta.ca.

0954-009 1/95/020167-20

2. The Value Unit Architecture

Recently, a variant of the generalized delta rule of Rumelhart *et al.* (1986) was developed to train value unit networks, which represent an extension of the generic PDP architecture (Dawson & Schopflocher, 1992). Value units are characterized by a non-monotonic activation function (a particular form of the Gaussian), rather than a sigmoid activation function (like the logistic) that characterizes generic processing units. (Following Ballard (1986) we call generic processing units 'integration devices'.) As a result, a value unit will only generate strong activations to a relatively narrow range of net inputs (see also Ballard, 1986). The Gaussian activation function that we use is $G(\text{net}_{pj}) = \exp[-\pi(\text{net}_{pj} - \mu_j)^2]$, in which net_{pj} is the net input to unit j produced by pattern p (computed in the same fashion as is done for standard backpropagation networks) and μ_j is the mean of the Gaussian activation function in unit j (and is analogous to the bias of a standard unit that uses a sigmoid activation function). This function carves what Hartman and Keeler (1991) call a Gaussian-shaped 'hyperbar' through a pattern space, and as a result a value unit network can also be differentiated from radial basis function networks (e.g. Moody & Darken, 1989) whose units carve Gaussian-shaped 'hypercones' through pattern spaces, and whose units use a distance measure for net input instead of the more typical dot product (for a detailed distinction between value unit and RBF networks see Dawson Schopflocher (1992, pp. 27–30).

Value unit networks have been shown to have a number of advantages over more traditional multi-layer perceptions including faster learning of linearly non-separable classes, better generalization and better ability to be 'scaled up' from toy problems (e.g. Dawson & Schopflocher, 1992; Dawson *et al.*, 1992, 1993; Medler & Dawson, 1994; Shamanski *et al.*, 1994). In the next section, we describe a newly discovered characteristic of value units that indicates that value unit networks may also be very straightforward to interpret

3. Characterizing Hidden Units with Jittered Density Plots

Consider using a relatively large number of patterns to train a PDP network. After training, one could present each pattern to the network, and record the activity that each pattern produced in each hidden unit. Then one could use this information to create a jittered density plot for each hidden unit. In such a plot, the horizontal position of each plotted point represents the activation produced by one of the training patterns, and a random vertical jittering is introduced to prevent points from overlapping (Chambers *et al.*, 1983, pp. 19–21). The purpose of the density plot is to provide some indication of the distribution of activities in the unit produced by the training set.

Because value units are 'tuned' to respond only to a narrow range of net input values, we predicted (at the start of this research programme) that a density plot for a hidden value unit would typically reveal two distinct high-density areas or 'bands', one near an activation of 0, the other near an activation of 1. The rest of the density plot was predicted to be empty. This hypothesis was based on the assumption that value units would generate a strong response to a subset of patterns that possessed a particular feature, and would generate a zero response to all the other patterns.

As described in detail below, we have recently observed that the qualitative

nature of this hypothesis is true: hidden value units generate banded density plots. However, the behaviour of hidden value units is much more sophisticated and interesting than we had originally anticipated. First, while density plots of value unit activations typically reveal distinct bands, it is frequently the case that more than two bands are evident. Second, each of these bands appears to support a coherent interpretation: each pattern that falls into a band is characterized by a specific feature or set of features.

To illustrate these properties, we trained a network of value units to solve a logical inference problem originally studied by Bechtel and Abrahamsen (1991, pp. 163–171). The problem set consists of six different types of arguments. The task of a trained network is, when presented with an argument, to identify its type and to classify it as being either valid or invalid. We elected to study this particular problem because it is psychologically relevant, it is sufficiently rich to make network interpretation challenging, and it is composed of a reasonably large number of stimulus patterns.

4. Method

4.1. Training Set

We trained the network using Bechtel and Abrahamsen's (1991) original stimulus set, which they kindly provided to us. Each pattern in the training set was a logical argument consisting of two sentences and a conclusion. The first sentence was composed of a connective and two variables; the second sentence and the conclusion were each composed of a single variable. Each of the four variables in an argument could be negated or not negated. The problem set consisted of four classes of problem (modus ponens, modus tollens, alternative syllogism and disjunctive syllogism); there were two different versions of each AS and DS syllogism type. Table I illustrates examples of valid arguments for each problem type, and also introduces the descriptive notation that we adopted to aid network interpretation.

Each argument was represented as a binary pattern of activity in a set of 14 input units (see Figure 1) using the representational scheme adopted by Bechtel and Abrahamsen (1991). Different examples of each argument type were constructed by selecting two variables from a set of four (A,B,C,D) and by allowing variables to be negated. For each type of argument, 48 different valid instances (the conclusion follows from the two sentences) and 48 different invalid instances (the conclusion does not follow from the two sentences) were used, creating a total training set of 576 patterns.

4.2. Network Architecture

A network of value units with 14 input units was trained on the problem set described above. The network had three output units. Two of the output units were used to represent one of four argument types (modus ponens, modus tollens, alternative syllogism, disjunctive syllogism); the third was used to indicate argument validity. In contrast to Bechtel and Abrahamsen's (1991) original network, which used two layers of 10 sigmoid hidden units, the value unit network had a

Table I. Examples of valid instances of each of the argument types in the problem set. The accompanying descriptive notation was used as an aid for interpreting network structures. Note that in the problem set any of the variables (i.e. S1(V1), S1(V2), S2, C) could be negated. If two variables are both positive or are both negative, we say that they have the same sign. Otherwise we say that they are opposite in sign. In the problem set, valid MP and MT arguments were turned into invalid ones by interchanging the letter for S2 and the letter for the conclusion. For all other problem types, valid arguments were turned into invalid ones by interchanging the sign of S2 and the sign of the conclusion

Problem type	Problem example	Descriptive notation
Modus ponens (MP)	*If* A *then* B	Connective: IF ... THEN
	A	S1(V1): A
		S1(V2): B
	Therefore B	S2: A
		C; B
Modus tollens (MT)	*If* A *then* C	Connective: IF ... THEN
	Not C	S1(V1) A
		S1(V2): C
	Therefore not A	S2: C
		S2 is negated
		C is negated
Alternative syllogism (AS)	D *or* A	Connective: OR
Type 1	Not D	S1(V1) A
		S1(V2): A
	Therefore A	S2: D
		S2 is negated
		C: A
Alternative syllogism (AS)	B *or* C	Connective: OR
Type 2	Not C	S1 (V1): B
		S1(V2): C
	Therefore B	S2: C
		S2 is negated
		C: B
Disjunctive syllogism (DS)	*Not both* C *and* D	Connective: NOT BOTH ... AND
Type 1	C	S1(V1: C
		S1(V2): D
	Therefore not D	S2: C
		C: D
		C is negated
Disjunctive syllogism (DS)	*Not both* A *and* D	Connective: NOT BOTH ... AND
Type 2	D	S1(V1): A
		S1(V2): D
	Therefore Not A	C: A
		C is negated

single layer of 10 hidden units (in general, because of the non-monotonic nature of the value unit's activation function, fewer hidden units are required to solve problems than are required by the standard architecture). Pilot studies had shown that a network with 10 hidden value units would reliably converge to a solution to the training problem, while nets with smaller numbers of hidden units would not.

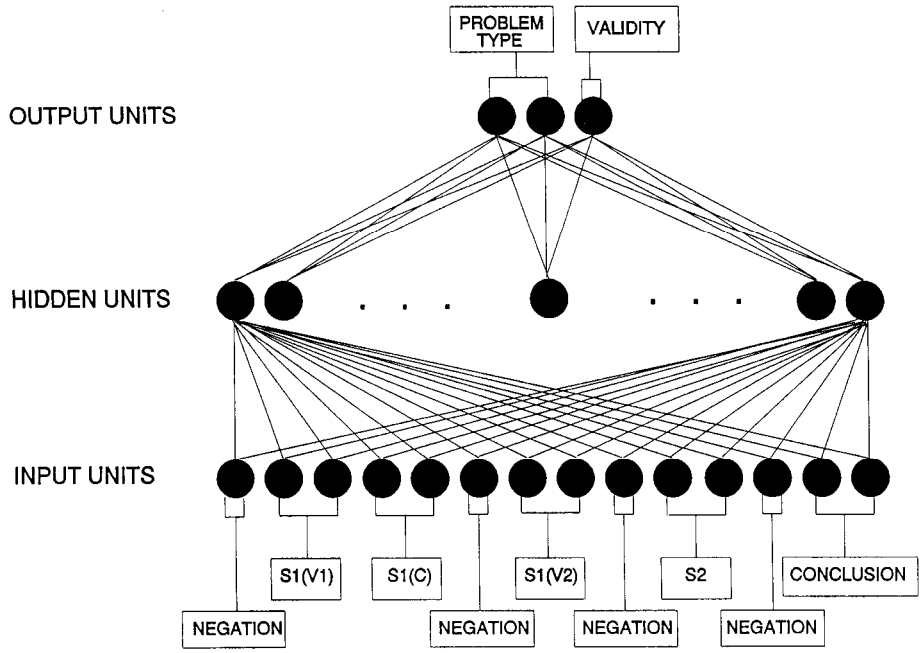

Figure 1. The network architecture for solving the logic problem, illustrating the functional role of the input and output units. In the integration device network, 15 hidden units were used, while only 10 hidden units were used in the value unit network. For both networks, the layer of input units was fully connected to the layer of hidden units, which in turn was fully connected to the output unit layer. The following representational scheme for input patterns was used, after Bechtel and Abrahamsen (1991, p. 169): (1) for the two connective bits, values of [1,1] represented 'IF ... THEN', [0,1] represented 'Or', and [1,0] represented 'NOT BOTH ... AND'. (2) For any pair of input units representing a variable, [0,1] represented 'A', [1,0] represented 'B', [1,1] represented 'C', and [0.0] represented 'D'. (3) If an input unit representing negation had a value of 1, then the variable following this bit was negated. If the unit had a value of 0, then the following variable was not negated.

4.3. Training Procedures

The network of value units was trained with the Dawson and Schopflocher (1992) extension of the generalized delta rule, using a learning rate of 0.03 and a momentum of 0.0. Network weights and biases (i.e. the means of the Gaussian function) were randomly set in the same range as the integration device network. However, in the simulations reported here, biases were not altered during learning. (Similar results to those reported here are obtained when biases are trained, and when they are held constant at zero.) While holding biases constant tends to slow learning down in value unit networks, it also increases the number of connection weights that fall near zero; in general, holding biases constant does not lead to learning in generic networks (Dawson *et al.*, 1992). Network connections were updated after every pattern presentation, and pattern presentation was randomized every epoch. The network was trained until a 'hit' was recorded for every output unit for every pattern in the training set. We operationalized a hit as being an

activation of 0.9 or greater when the desired output was 1, and as being an activation of 0.1 or less when the desired output was 0. Convergence was achieved in 5793 epochs.

5. Results

After the network was trained to convergence, the stimulus set was presented once again, and the activity produced in each hidden unit was recorded. This information was then used to create jittered density plots for each hidden unit.

Figure 2 depicts the density plots for the 10 hidden units of the value unit network. First, it shows that all but one of the hidden units (hidden unit 1) produced density plots that were markedly banded. Second, all of the density plots show a very high density of patterns, producing zero or near zero levels of activation, as is revealed by the narrow dark line on the left side of each plot. Third, most of the density plots have three or more definite bands (hidden units 0, 2, 4, 5, 7 and 8).

5.1. Why Do Value Units Produce Bands?

Why did bands appear in the value unit network? Unlike standard units that employ a sigmoid-shaped activation function, value units—because of the bell-shape of their activation function—will only respond to a limited range of net inputs. This places constraints on the patterns of connectivity that a value unit network can use to solve a pattern classification problem. As a result, value units frequently produce balanced connection weights that fan into the same hidden unit. When this occurs, one connection weight will have the value x and will be balanced by another connection weight that has the value $-x$.

This balancing of connection weights appears to be responsible for banding. When balancing of weights occurs, one can activate many different patterns of input activity that produce the same net input, and fall into the same band of hidden unit activity, because the balancing cancels out different input signals. As Figure 3 shows, units that demonstrate very nice banding are characterized by a great deal of connection weight balancing, while units that do not demonstrate banding do not exhibit this balancing. This raises the important possibility that other PDP architectures that use tuned activation functions, such as RBF networks (e.g. Moody & Darken, 1989), may also exhibit banding when density plots of hidden unit activities are constructed.

5.2. Identifying Definite Features

The fact that the value unit plots are very dense at zero activation, and the fact that they are banded, were consistent with our original predictions. However, we were quite surprised when density plots revealed three or more distinct bands. To investigate this phenomenon further, for each hidden unit we identified the members of the training set that comprised each band in the density plot. Then, we attempted to identify common attributes of patterns belonging to the same band. These common attributes, which we call 'definite features', were identified by computing correlations among the 14 binary features for the patterns that fell into a particular band. A definite unity feature was defined as an input bit that had

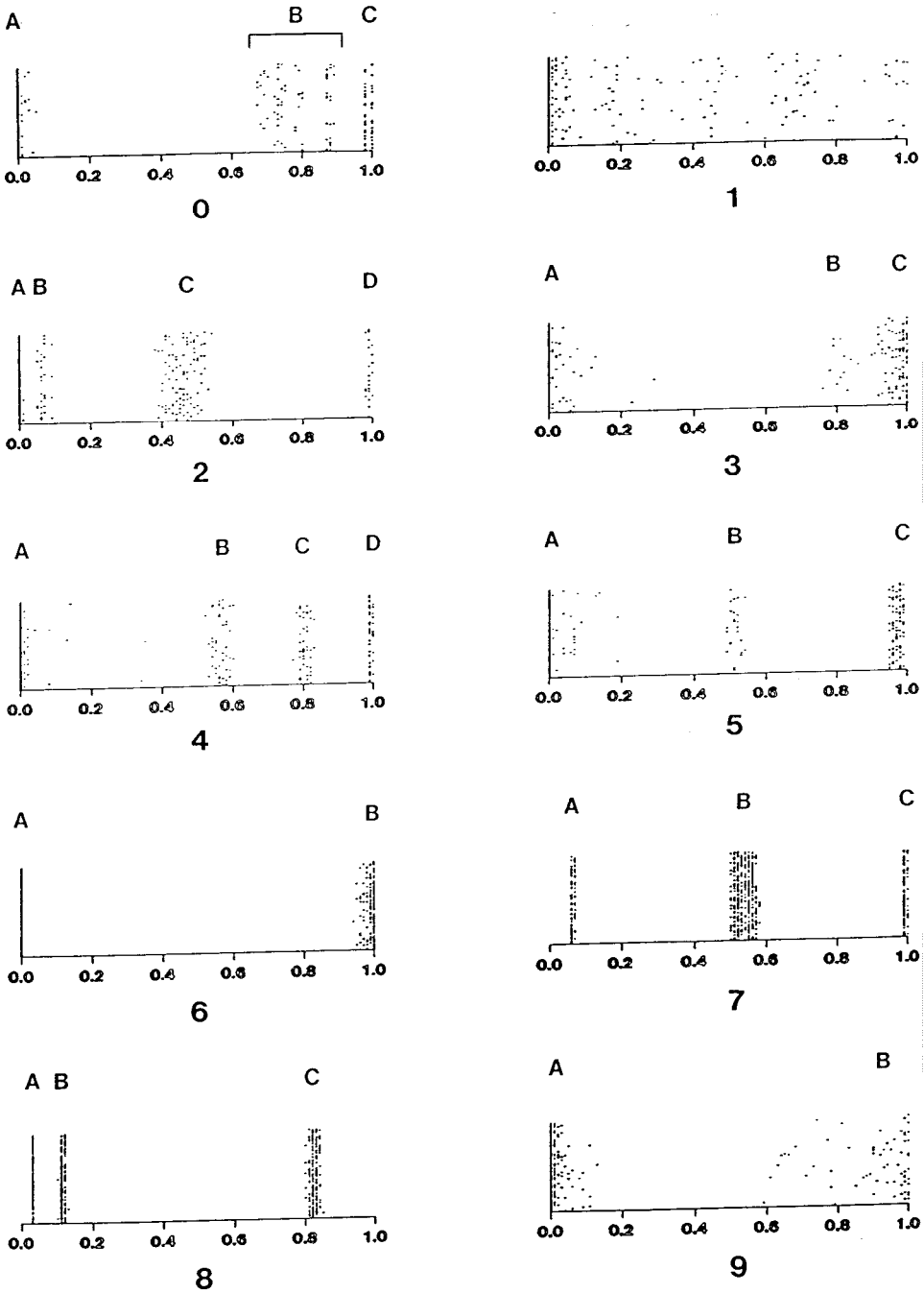

Figure 2. The jittered density plots for each of the 10 hidden units obtained for the value unit network after it was trained to convergence. Letters are used as labels for the observed bands, and correspond to the labels used in Table II.

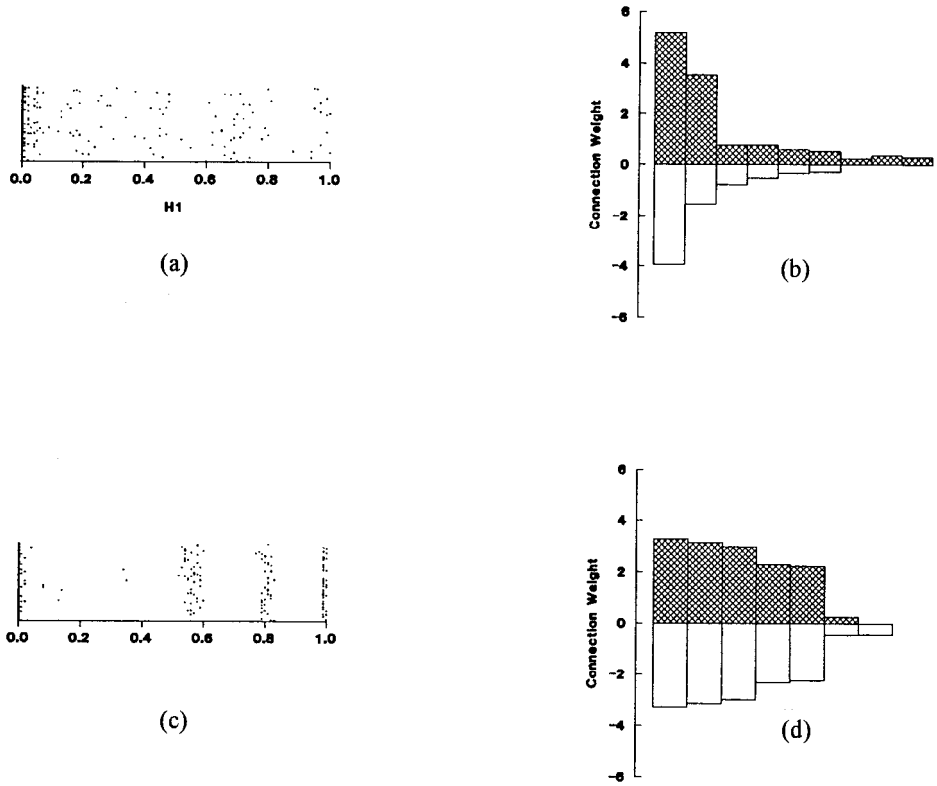

Figure 3. Balancing of the connection weights into a hidden value unit produce banding. (a) The density plot for hidden value unit 1, which did not produce much banding. (b) Histogram of the connection weights leading into this unit. Each hashed bar represents the weight of a single positive connection, and each white bar represents the weight of a single negative connection. The asymmetry in this figure shows a lack of balance between positive and negative connections. (c) The density plot for hidden value unit 4, which demonstrated excellent banding. (d) Histogram of the connection weights leading into hidden value unit 4 show excellent balancing, as indicated by a much greater degree of symmetry than in (b). In general, value units that demonstrate banding have a great deal of balancing between positive and negative connection weights.

a constant value for all patterns within the band. A definite binary feature was defined as a perfect negative or perfect positive correlation between pairs of binary features, the former representing the fact that two bits were always opposite in value, the latter representing the fact that two bits were always equal in value. Remarkably, most of the bands illustrated in Figure 2 produced definite features defined using these objective, quantitative definitions. Furthermore, because each input bit is associated with a known interpretation, each of the bands that revealed definite features had a very basic and elegant interpretation.

For example, consider the density plot for hidden unit 8 in Figure 2, which is composed of three different bands. All of the patterns that fall into band A in the figure (activation = 0.03) contain the connective OR. All of the patterns that fall into band B (activation from 0.10 to 0.13) contain the connective IF ... THEN.

All of the patterns that fall into band C (activation from 0.80 to 0.85) contain the connective NOT BOTH ... AND. In general, then, this unit is a connective detector, and adopts three distinct levels of activity to signal that one of three different connective types has been detected. Table II provides the interpretations of the bands for each of the hidden units.

5.3. Using Bands To Predict Network Behaviour

The preceding analyses have shown that the activations of hidden value units can be organized into identifiable bands, which are in turn associated with interpretable definite features. However, an important question remains: Do these bands play any role in explaining how the network solves the problem? For example, it is possible that these bands are merely epiphenomenal, and that in order to account for network performance one must ignore the bands and still rely on specific activity values produced by patterns.

In order to address this issue, we described the patterns presented to the network in terms of the bands of activity that they produced for each hidden unit, instead of describing the patterns in terms of their input features. We found that if one only knows the median activity of each hidden unit band that a particular pattern belongs to, then one can still make very accurate predictions about network outputs.

For example, when described in terms of activity bands, every valid modus ponens problem produced a single pattern of hidden unit activity in the network: [0-A, I-A, 2-B, 3-A, 4-A, 5-A, 6-A, 7-B, 8-B, 9-A], where 0-A means 'produced activity in Band A of hidden unit 0'. Using Table II, we can replace each of these bands with its median level of activity, and represent this activity pattern as the vector [0, 0, 0.46, 0, 0, 0, 0, 0.54, 0.11, 0]. The vector of weights from the 10 hidden units to output unit 0 is [0.71, 0, −1.24, 0, −0.70, 0.72, 0.55, −0.48, −2.18, −0.03]. The dot product of this weight vector with the hidden unit activity vector produces a net input of −1.0964, which in turn will produce an activity of 0.005 in output unit 0 because it has $\mu = 0.23$ in its Gaussian equation. The vector of weights from the 10 hidden units to output unit 1 is [0.62, −0.01, −1.06, 0.00, −0.86, 0.89, −0.33, 0.24, 2.32, 0.01]. The dot product of this weight vector with the activity vector produces a net input of −0.1028, which in turn will produce an activity of 0.998 in output unit 1 because it has $\mu = -0.13$. Finally, the vector of weights from the 10 hidden units to output unit 2 is [2.14, 4.14, −0.89, 1.29, 1.66, −1.70, −0.40, 0.91, 0.08, 1.21]. The dot product of this weight vector with the activity vector produces a net input of 0.0908, which in turn will produce an activity of 1.000 in output unit 2 because it has $\mu = 0.10$.

To summarize this example, knowing only the medians of the bands of activity that valid modus ponens problems fall into, we predict that the network's response to any of these problems will be [0.005, 0.998, 1.00]; the desired network outputs [0, 1, 1]. Similar accounts can be provided for banded patterns of hidden unit activity provided by other valid problem types. In short, our knowledge of the hidden unit activity bands provides an excellent predictor of network output for these problems, indicating that the bands play an important role in explaining network behaviour, and that they are not merely epiphenomenal.

Table II. Interpretations of bands in the hidden unit density plots

Unit number	Band	Number of patterns	Median activity	Interpretation of definite features
0	A	456	0.00	No definite features
	B	72	0.77	S1(V1) is the same letter as S2
				S1(V2) is the same letter as C
				The connective is not IF ... THEN
	C	48	0.99	S1(V1) is the same letter as S2
				S1(V1) is opposite in sign to S2
				S1(V2) is the same letters as C
				S1(V2) is opposite in sign to C
				The connective is IF ... THEN
1	A	576	0.00	No definite features
2	A	456	0.00	No definite features
	B	96	0.46	S1(V1) is the same letter as S2
				S1(V1) is the same sign as S2
				S1(V2) is the same letter as C
				The connective is not NOT BOTH ... AND
	C	24	0.99	S1(V1) is the same letter as S2
				S1(V1) and S2 are not negated
				S1(V2) is the same letter as C
				S1(V2) is opposite in sign to C
				The connective is NOT BOTH ... AND
3	A	456	0.00	No definite features
	B	12	0.81	S1(V1) is negated
				S1(V1) is the same letter as C
				S2 and C are not negated
				S1(V2) is the same letter as S2
				The connective is OR
	C	86	0.99	S1(V1) is the same letter as C
				S1(V2) is the same letter as S2
4	A	431	0.00	No definite features
	B	48	0.56	S1(V1) is the same letter and sign as C
				S1(V2) is the same letter and sign as S2
				The connective is IF ... THEN
	C	48	0.81	S1(V1) is the same letter and sign as C
				S1(V2) is the same letter as S2
				S1(V2) is opposite in sign to S2
				The connective is NOT BOTH ... AND
	D	48	0.99	S1(V1) is the same letter and sign as C
				S1(V2) is the same letter as S2
				S1(V2) is opposite in sign to S2
				The connective is OR
5	A	456	0.00	No definite features
	B	24	0.51	S1(V1) is the same letter as C
				S1(V1) is opposite in sign to C
				S1(V2) is the same letter as S2
				S1(V2) and S2 are not negated
				The connective is NOT BOTH ... AND
	C	96	0.97	S1(V1) is the same letter as C
				S1(V2) is the same letter as S2
				S1(V2) is opposite in sign to S2
				The connective is not NOT BOTH ... AND
6	A	384	0.00	The connective is not OR
	B	192	1.00	The connective is OR
7	A	96	0.06	S2 is negated
				The connective is NOT BOTH ... AND
	B	384	0.54	The connective is not NOT BOTH ... AND
	C	96	0.99	S2 is positive
				The connective is NOT BOTH ... AND
8	A	192	0.03	The connective is OR
	B	192	0.11	The connettive is IF ... THEN
	C	192	0.82	The connective is NOT BOTH ... AND
9	A	512	0.00	No definite features
	B	64	0.95	No definite features

5.4. The 'Rules in the Network's Head'

Another advantage of representing patterns as the set of hidden unit activity bands to which they belong is that these patterns can be viewed as a 'rule' that the network uses to make judgements about different types of logic problems. This is because this pattern represents a set of features which, when combined, dictate the network's response to the pattern. Importantly, by identifying the 'logical rules' in the network's head' we can raise interesting empirical questions about how humans might learn to deal with these logic problems.

Table III presents the traditional rules of inference (in our notation) of all of the valid problem types, as well as the bands of activity produced in the network by these problems. By providing the interpretation for each of these bands from Table II, one can construct the rules that the network uses for this task (see Table III). There are several important points that arise from studying this table. First, the network has a very small number of rules for identifying valid problem types: it has one rule for modus ponens, one rule for modus tollens, two rules for alternative syllogisms and three rules for disjunctive syllogisms. Second, some of the rules in the network are equivalent to the traditional rules of inference used in natural deduction systems (cf. Bergmann *et al.*, 1990). The network has learned the traditional rules of inference for modus ponens. modus tollens, and one version of alternative syllogism. Given this, we would expect that the network's performance would generalize well to patterns that it had not been trained on (e.g. by defining new variables by using continuous inputs to the relevant input units.) Third, the network learned a number of rules of inference which are significantly different from the traditional ones. One of the rules for alternative syllogisms and one of the rules for disjunctive syllogisms are what might be termed 'default' rules. Default rules work by identifying the main connective, and require that no other definite features are present. The other two rules for disjunctive syllogisms are similar to the traditional rules, but require in addition that S2 is not negated.

This last point is important for the psychological relevance of this type of model. It has long been known that the formal rules of inference do not always provide good accounts of how humans solve logic problems (e.g. Johnson-Laird, 1983). Instead, humans appear to deal with these problems by building *ad hoc* mental models that generally lead to correct solutions, though the models themselves have little resemblance to logical formalisms. Some of the rules learned by the network have this *ad hoc* appearance. For example, one of the rules for a valid alternative syllogism (the default rule) can be described as 'If the connective is OR, and no other features that 1 know about are present, then it must be a valid AS'. It is easy to imagine that this kind of rule might be used by a student who is learning about this type of problem, but is still unsure about its formal characterization.

In point of fact, the results in Table III raise some interesting empirical questions that could be addressed by a cognitive psychologist. If human subjects were to learn to solve these logical questions, would they tend to develop *ad hoc* mental models for the AS and DS problems, but not to do so for MP and MT? Would human subjects treat DS problems with negated S2 differently than DS problems with non-negated S2? Would human subjects develop a relatively small number of rules, paying attention to the same features as the model? Note that the fact that these questions can be raised depends on two things. The first is that unlike symbolic models of logic (e.g. Rips, 1994) in which the logical rules are

Table III. Interpretation of the rules that the network uses to identify valid instances of each problem type. Braces are used to indicate the equivalence between network rules and traditional formal rules of inference. Legend: S1(V1) and S1(V2) are the first and second variables in sentence 1; S2 is the variable in sentence 2; C is the variable in the conclusion; SIGN refers to whether a variable is negated or not negated.

Problem type	Formal definition of rule for valid problem type		Hidden unit bands produced by problem type	Interpretation of hidden unit bands		Notes about network rules
Valid modus ponens (MP)	S1(V1) = S2 S1(V2) = C SIGN S1(V1) = SIGN S2 SIGN S1(V2) = SIGN C CONNECTIVE: IF...THEN	ü ï ý ï þ	[0-A,1-A,2-B,3-A,4-A, 5-A,6-A,7-B,8-B,9-A]	S1(V1) = S2 S1(V2) = C SIGN S1(V1) = SIGN S2 CONNECTIVE: IF...THEN	ü î ì þ	The network rule is the same as the formal rule except that the network does not pay attention to the signs of S1(V2) and C. Due to the nature of the training set, though, this is not necessary.
Valid modus tollens (MT)	S1(V1) = C S1(V2) = S2 SIGN S1(V1) ≠ SIGN C SIGN S1(V2) ≠ S1GN S2 CONNECTIVE: IF ... THEN	ü ï ý ï þ	[0-A,1-A,2-A,3-C,4-A, 5-C,6-A,7-B,8-B,9-A]	S1(V1) = C S1(V2) = s2 SIGN S1(V2) ≠ SIGN S2 CONNECTIVE: IF...THEN	ü î ì þ	Although the network does not pay attention to the signs of S1(V1) or C, this is not significant due to the nature of the training set.
Valid Alternative syllogism (AS) (There are two versions of AS in the training set)	S1(V1) = S2 S1(V2) = C SIGN S1(V1) ≠ SIGN S2 SIGN S1(V2) = SIGN C CONNECTIVE: OR		[0-A,l-A,2-A,3-A,4-A, 5-A,6-B,7-B,8-A,9-A]	CONNECTIVE: OR		This is a default rule. Provided the connective is OR and no other definite features are true of the pattern, then the problem must be a valid AS.
	S1(V1) = C S1(V2) = S2 SIGN S1(V1) = SIGN C SIGN S1(V2) ≠ SIGN S2 CONNECTIVE: OR	ü ï ý ï þ	[0-A,1-A,2-A,3-A,4-D, 5-C,6-B,7-B,8-A,9-A]	S1(V1) = C S1(V2) = S2 SIGN S1(V1) = SIGN C SIGN S1(V2) ≠ SIGN S2 CONNECTIVE: OR	ü ï ý ï þ	Here the network employs exactly the classical rule

Table III. Continued.

Valid disjunctive syllogisms (DS) (There are two versions of DS in the training set)	S1(V1) = S2 S1(V2) = C SIGN S1(V1) = SIGN S2 SIGN S1(V2) ≠ SIGN S2 CONNECTIVE: NOT BOTH ... AND	[0-A,1-A,2-A,3-A,4-A, 5-A,6-A,7-A,8-C,9-A]	S2 IS NEGATED CONNECTIVE: NOT BOTH ... AND	This is another default rule. Provided that S2 is negated, the connective is NOT BOTH ... AND, and no other definite features are present, then the problem must be a valid DS.
	S1(V1) = C S1(V2) = S2 SIGN S1(V1) ≠ SIGN C SIGN S1(V2) ≠ SIGN S2 CONNECTIVE: NOT BOTH ... AND	[0-A,1-A,2-A,3-A,4-A, 5-B,6-A,7-C,8-C,9-A]	S1(V1) = C S1(V2) = S2 SIGN S1(V2) ≠ SIGN C SIGN S1(V2) = SIGN S1(V2) IS NOT NEGATED S2 IS NOT NEGATED CONNECTIVE: NOT BOTH ... AND	This network rule is the same as the second formal rule for DS apart from the additional stipulation that S2 and S1(V2) are not negated.
		[0-A,1-A,2-C,3-A,4-A, 5-A,6-A,7-C,8-C,9-A]	S1(V1) = S2 S1(V2) = C SIGN S1(V1) = SIGN S2 SIGN S1(V2) ≠ SIGN C S1(V1) IS NOT NEGATED S2 IS NOT NEGATED CONNECTIVE: NOT BOTH ... AND	This network rule is the same as the first formal rule for DS apart from the additional stipulation that S2 and S1(V2) are not negated.

programmed in, the connectionist network has to learn how to solve the problems, raising the possibility that surprising new rules will be discovered. The second is that the discovery of these rules in the network depends upon our ability to interpret its structure.

5.5. Relation to Other Interpretive Techniques

While some researchers have questioned the usefulness of connectionist models because they are difficult to interpret (e.g. McCloskey, 1991; Robinson, 1992), this does not mean that connectionists themselves do not attempt to analyze the internal structure of their networks. In this section, we briefly consider the relationship between some other methods of network interpretation and our approach to interpreting hidden value unit activity bands.

Some researchers have successfully interpreted network structure by examining the relative sizes of connection weights that feed into a network (e.g. Hinton, 1986). This approach has the advantage of being very simple to do, because it typically involves depicting connection weights in some pictorial fashion that is easy to interpret (e.g. a Hinton diagram). However, this method is not without limitations. For example, in order to interpret a unit's function by looking at the connection weights that feed into it, one must know what feature is associated with the connection. Unfortunately, such labels might be difficult to assign in networks with multiple layers of hidden units. This is because one may not know with certainty what features are encoded in the connections between adjacent layers of hidden units, for these features in turn depend upon interpretations of other hidden units (see also Hanson & Burr, 1990, Section 5.7). In general, these kinds of problems will always arise when interpretations are based upon the analysis of network structure (i.e. the 'network space') instead of the analysis of stimulus features (i.e. the 'pattern space').

We believe that the interpretation of hidden value unit activity bands has advantages over the examination of connection weights. First, it is just as simple to accomplish, as it only requires one to graph density plots of hidden unit activities in order to identify bands. Second, it is less subjective: once bands are identified, definite features are discovered by computing correlations among pattern features. Third, the analysis of bands is always done in pattern space, looking for features shared by patterns that produce the same band of activity in a hidden unit. As a result. the interpretation of bands can still be accomplished in networks that have more than one layer of hidden units.

Other researchers have approached the problem of network interpretation by applying multivariate statistics to connection weights or to hidden unit activations (for a review, see Hanson & Burr, 1990). While these techniques are very powerful, they are not without their drawbacks. First, they assume linear relationships among variables: in general, such relationships are not true of PDP networks. Second, many of the design decisions required to, say, factor analyze some aspect of a trained network are as complex as the design decisions used to construct the original network. In particular, when doing factor analysis one has to decide how many factors to extract, what kinds of factors to extract, whether to rotate the factor structure and (if desired) what kind of rotation to perform (for an introduction to such issues, see Cattell, 1978). Third, because of the complexity of the design decisions underlying the multivariate analysis, there is no guarantee that the features that are revealed by this analysis are actually valid (e.g. Eysenck, 1967).

Different researchers who, for example, choose different criteria for factor rotation, would propose completely different interpretations of the same network.

In comparison, our approach to interpreting bands of hidden value unit activations appears to have several advantages over traditional multivariate analytical techniques. First, the bands are determined after hidden unit activations have been calculated, and as a result incorporate the non-linear transformation of the net input provided by the hidden unit. Second, the bands that are interpreted are provided by the network itself, and are not an artefact of additional assumptions required by an analytic technique. In our approach, the network itself provides a natural clustering or factoring of the input patterns, which we then analyze.

5.6. Does Banding Occur in Other Problems?

To this point, we have concentrated on a single example to make the point that hidden value units can organize their responses to a pattern set into interpretable bands. However, it is also important to show that the banding phenomenon is not limited to a particular problem.

Indeed, we have found hidden value unit banding in a variety of other problems. Figure 4 illustrates some sample density plots for all of the hidden units in networks trained on three different problems: 7-bit majority, 6-bit parity and a variation of Hinton's (1986) kinship problem. As can be seen from inspecting this figure, the density plots typically are organized into a distinct pattern of bands. However, the mere presence of bands is not sufficient to make them of interest to us. Are the bands illustrated in Figure 4 associated with specific interpretations, as was the case with the logic network described earlier? Again, this question can be answered affirmatively.

Figure 4(a) illustrates the density plot for the single hidden unit that a value unit network requires to determine whether more than half of 7 input units have been activated. (The non-monotonic nature of the value unit's activation unit results in a value unit network requiring a hidden unit for this linearly separable problem; for more details see Shamanski *et al.*, 1994.) The density plot is organized into six distinct bands. With the exception of the leftmost band, each band contains all patterns that have a specific number of 'on' bits (indicated by the number above each band in the figure). The leftmost band includes all patterns that have 5, 6 or 7 bits turned on. In general, this hidden unit is a 'minority' detector, generating a response of 0.2 or greater to those patterns in which a minority of the bits are activated; the response of this hidden unit is then inverted by the network's output unit.

Figure 4(b) presents density plots for the two hidden units that are required by a value unit network to detect odd parity in 6-bit input patterns. Again, each band is associated with a definite feature. Both hidden units organize the input bits into two groups. Group P consists of input bits 0 and 5; both of these bits have positive connection weights to each hidden unit (equal to 0.82 for hidden unit 0 and to 0.49 for hidden unit 1). Group N consists of the other four input bits, which have weights to the hidden units that are equal in magnitude but opposite in sign to those associated with the group P bits (i.e. weights of -0.82 for hidden unit 0 and of -0.49 for hidden unit 1). Let us define the balance of a pattern as the difference between the number of group N bits that it has turned on and the number of group P bits that it has turned on. Note that balance, defined in this way, is correlated with parity: if balance is odd, then parity is odd, and if balance

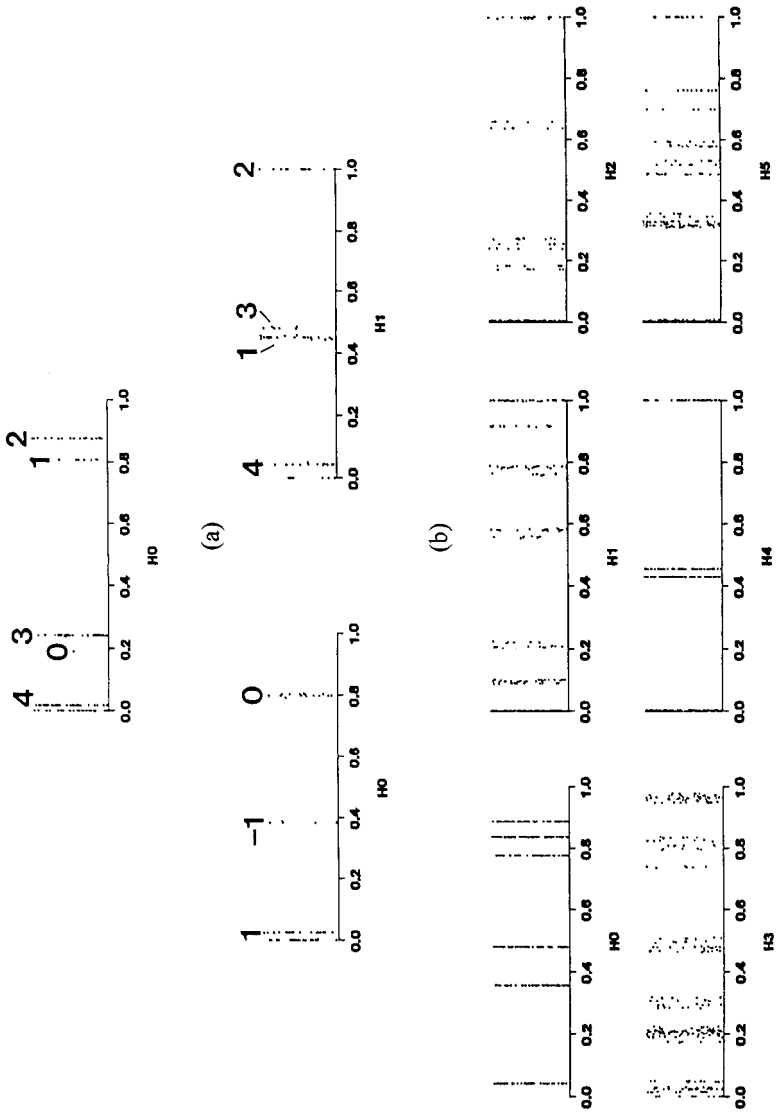

Figure 4. Jittered density plots for all hidden value units from networks trained on different problems. (a) 7-bit majority problem. The number associated with each band is the sum of the 'on' bits for each pattern in the band. (b) 6-bit parity problem. The number associated with each band is the balance calculated for each pattern in the band as described in the text. (c) Expanded version of the Hinton kinship problem. See text for details.

is even, then parity is even. The bands in the two Figure 4(b) density plots mediate parity detection by encoding balance: each band is associated with patterns that have a specific balance value, as indicated in the figure.

A concept's representation is distributed if it 'is represented by a pattern of activation across an ensemble or set of units; by design no single unit can convey that concept on its own' (Bechtel & Abrahamsen, 1991, p. 51). Clearly, parity is an excellent example of a distributed concept because one must consider all input units together to determine whether an odd or even number of them are activated, and no single input unit can signal the parity of the entire pattern. Thus, the results in Figure 4(b) are important because they demonstrate that value unit networks can produce bands for distributed representations; banding does not require a local encoding of input information.

Figure 4(c) provides the density plots for the six hidden value units trained on a version of Hinton's (1986) kinship problem. In the original problem, a complex network (36 input units, three layers of 6, 12 and 6 hidden units, 24 output units) was trained to encode kinship relationships in two (identical) family trees (104 training patterns). Purely local encoding was used to present the name of a person and a relationship as input to the network. The network was trained to generate the names of the person or persons that completed the relationship. For example, when provided the pattern that represented 'James has-father', the network was trained to generate the pattern representing James' father, 'Andrew'. In our version of the problem, we have used a distributed encoding that has permitted us to train a smaller network (22 input units, 6 hidden units, 10 output units) to accomplish this task for six family trees (312 training patterns). A complete interpretation of the resulting network is beyond the scope of the current paper. Let us simply note some of the highlights:

(1) Each band in hidden unit 0 and hidden unit 4 is associated with a specific family.
(2) Each band above 0.0 in hidden unit 1 picks out a specific relationship that points to a second generation person.
(3) Each band above 0.0 in hidden unit 2 picks out a specific relationship to a first generation person.
(4) Each band in hidden unit 3 is associated with an input person from a particular location in the family tree (e.g. 'the first person in generation 2').
(5) Each of the non-zero bands for hidden unit 5 captures a specific feminine relationship (e.g. 'has-mother', 'has-sister').

The results reported in this section clearly show that the production of interpretable bands in the density plots of hidden value units is not an artefact of the Bechtel and Abrahamsen (1991) logic problem. It is important to note, however, that we have not found banding for every problem that we have studied. For example, bands were not observed in a network that was trained to differentiate Alzheimer's patients from control subjects on the basis of single positron emission computed tomography measures (Dawson *et al.*, 1994). Early on, we suspected that this was because the inputs for this problem were continuous, and we hypothesized that banding of density plots may only occur when a value unit network is trained to discover mappings from binary inputs to binary outputs. However, pilot results suggest that this hypothesis is incorrect. In one study, we converted the binary patterns for the parity problem into continuous patterns by replacing all zeros with real values randomly selected from the range 0.0 to 0.49,

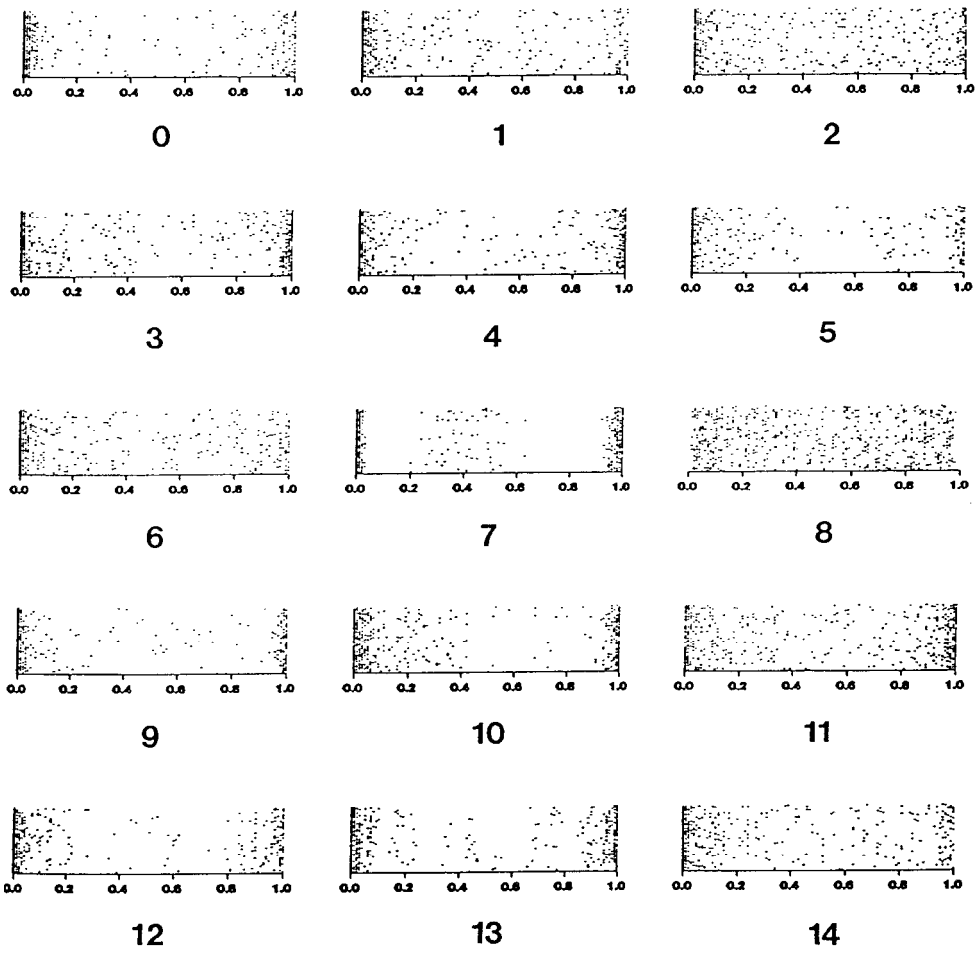

Figure 5. The jittered density plots for each of the 15 hidden units obtained for a standard network after it was trained to convergence. Note the absence of banding.

and by replacing all ones with real values randomly selected from the range 0.50 to 1.0. Banding still occurred. This is an important finding, because it suggests that we will still be able to see banding if additional layers of hidden units are added to the network; note that there is no guarantee that the inputs to these additional layers would be binary. However, this result does not explain why banding has not yet been found in the Alzheimer's data set. Our current research is attempting to determine the necessary and sufficient conditions for the production of banding in value unit density plots.

5.7. Can Bands be Found in Other Architectures?

Earlier, it was suggested that the interpretation of hidden unit activity bands may have certain advantages over other techniques for analyzing network structure. It is therefore important to consider whether these bands might appear when examining the activities of hidden units in other PDP architectures.

The most commonly used feedforward network is one in which there is a single

layer of hidden units, and the units are characterized by a sigmoid-shaped activation function. We have not yet found substantial banding in the hidden units of this type of network. For example, Figure 5 presents the jittered density plots for such a network with 15 hidden units which was successfully trained on the Bechtel and Abrahamsen (1991) logic problem. The most common pattern in these plots was two dense regions of activity near activation of 0 and 1, with an even distribution of activity 'smeared' across the rest of the plot (hidden units 0, 2, 3, 4, 9, 10 and 13). The next most common pattern was a single dense region of activity near 0 activation accompanied by a uniform smear across the rest of the plot (hidden units 1, 5, 6, 12 and 14). The plot for hidden unit 8 was simply a uniform smear across the entire plot, with no evident bands. Only hidden unit 7 demonstrated any distinct banding, with narrow bands at 0 and 1 activations, and a sparse, broad band ranging from an activation of 0.2 to an activation of 0.6.

While we have not found bands for hidden units in the standard architecture, this is not in principle a limitation of units that employ a sigmoid-shaped activation function. First, some circuits in which two units with sigmoid activation functions serve as input units to a third are logically equivalent to a single value unit (Dawson & Schopflocher, 1992). Thus, in a standard network in which two layers of hidden units are used, we would not be surprised to find interpretable bands emerging in the second layer of hidden units. Second, the emergence of bands in our architecture appears to be a consequence of the activation function's tuned nature, which places constraints on the patterns of connectivity that solve pattern recognition problems. We would expect that other architectures that use tuned activation functions, such as RBF networks (e.g. Moody & Darken, 1989), would also reveal interpretable bands of hidden unit activity.

In summary, we do not believe that the value unit architecture is the only one that will produce interpretable bands. Our hope is that researchers interested in other architectures will be able to find this kind of structure in their networks, and as a result be able to provide detailed interpretations of how their networks function.

6. Conclusion

Mozer and Smolensky (1989, p. 3) have noted that "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". Indeed, some researchers believe that this is in principle a limitation of PDP networks that is due to their non-linear activation functions and their development of distributed representations: "We may have to accept the inexplicable nature of mature networks" (Robinson, 1992, p. 655).

In contrast to this view, the results that were presented above suggest that the value unit architecture can produce interpretable structures without requiring the application of multivariate statistics. Simple density plots of hidden value unit activations revealed distinct bands that were assigned coherent interpretations. This indicates that the value unit architecture may be extremely valuable to researchers in cognitive science who are not content to view their networks as unassailable black boxes.

References

Ballard, D.H. (1986) Cortical connections and parallel processing: structure and function. *Behavioural and Brain Sciences*, **9**, 67–120.

Bechtel, W. & Abrahamsen, A. (1991) *Connectionism and the Mind*. Cambridge, MA: Basil Blackwell.

Bergmann, M., Moor, J. & Nelson, J. (1990) *The Logic Book*. New York: McGraw-Hill.

Cattell, R.B. (1978) *The Scientific Use of Factor Analysis in Behavioral and Life Sciences*. New York: Plenum Press.

Chambers, J.M., Cleveland, W.S., Kleiner, B. & Tukey, P.A. (1983) *Graphical Methods for Data Analysis*. Belmont, CA: Wadsworth International Group.

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.

Dawson, M.R.W. & Shamanski, K.S. (1994). Connectionism, confusion, and cognitive science. *Journal of Intelligent Systems*, **4**, 215–262.

Dawson, M.R.W., Schopflocher, D.P., Kidd, J. & Shamanski, K.S. (1992) Training networks of value units. *Proceedings of the Ninth Canadian Artificial Intelligence Conference*, 244–750.

Dawson, M.R.W., Shamanski, K.S. & Medler, D.A. (1993) From connectionism to cognitive science. In L. Goldfarb (Ed.) *Proceedings of the Fifth University of New Brunswick Artificial Intelligence Symposium*. Fredericton, NB: UNB Press.

Dawson, M.R.W., Dobbs, A., Hooper, H.R., McEwan, A.J.B., Triscott, J. & Cooney, J. (1994). Artificial neural networks that use SPECT to identify patients with probable Alzheimers disease. *European Journal of Nuclear Medicine*, under editorial review.

Eysenck, H.J. (1967) The logical basis of factor analysis. In D. N. Jackson & S. Messick (Eds), *Problems in Human Assessment*. New York: McGraw-Hill, 288–299.

Hanson, S.J. & Burr, D.J. (1990) What connectionist models learn: Learning and representation in connectionist networks. *Behavioural and Brain Sciences*, **13**, 471–518.

Hartman, E. & Keeler, J.D. (1991) Predicting the future: advantages of semilocal units. *Neural Computation*, **3**, 566–578.

Hinton, G.E. (1986) Learning distributed representations of concepts. *Proceedings of the 8th Annual Meeting of the Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1–12.

Johnson-Laird, P. (1983) *Mental Models.* Cambridge, MA: Harvard University Press.

McCloskey, M. (1991) Networks and theories: the place of connectionism in cognitive science. *Psychological Science*, **2**, 387–395.

Medler, D.A. & Dawson, M.R.W. (1994) Training redundant artificial neural networks: imposing biology on technology. *Psychological Research*, **57**, 54–62.

Moody, J. & Darken, C.J. (1989) Fast learning in networks of locally-tuned processing units. *Neural Computation*, **1**, 281–294.

Mozer, M.C. & Smolensky, P. (1989) Using relevance to reduce network size automatically. *Connection Science*, **1**, 3–16.

Rips, L.J. (1994) *The Psychology of Proof.* Cambridge, MA: MIT Press.

Robinson, D.A. (1992) Implications of neural networks for how we think about brain function. *Behavioral and Brain Sciences*, **15**, 644–655.

Rumelhart, D.E., Hinton, G.E. & Williams, R.J. (1986) Learning representations by back-propagating errors. *Nature*, **323**, 533–536.

Schneider, W. (1987) Connectionism: is it a paradigm shift for psychology? *Behaviour Research Methods, Instruments and Computers*, **19**, 73–83.

Shamanski, K.S., Dawson, M.R.W. & Berkeley, I.S.N. (1994) The effect of linear separability on learning speed and generalization in monotonic and nonmonotonic PDP networks. *Connection Science*, under editorial review.

Reviewers for *Connection Science* Volume 6, 1994

In addition to the associate editors and members of the editorial board, the following persons kindly helped in the journal's reviewing process for Volume 6.

J. Bach	University of Copenhagen, Denmark
M.I. Bellgard	University of Western Australia, Australia
J. Bengaard	University of Copenhagen, Denmark
M. Casey	MIT Media Laboratory, USA
M. Gasser	University of Indiana, USA
R. Golden	University of Dallas, USA
M. Greenhough	University of Wales, UK
N. Griffith	University of Exeter, UK
S. Harnard	Princetown University, USA
S. Jackson	University of Sheffield, USA
B. Kamgar-Parsi	Naval Research Laboratory, Washington, DC, USA
B. Katz	University of Sussex, UK
E. Large	University of Ohio, USA
J. Murre	MRC APU, Cambridge, UK
M. Page	MRC APU, Cambridge, UK
A. Robins	University of Otago, New Zealand
B. Selman	AT&T Bell Laboratories, New Jersey, USA
S. Smoliar	National University of Singapore
C. Stevens	University of Queensland, Australia
I. Taylor	University of Wales, UK
A. Weigend	University of Colorado, USA

0954-009 1/95/020187-01