Coarse Coding In Value Unit Networks:

Subsymbolic Implications Of Nonmonotonic PDP Networks

C. Darren Piercey (dpiercey@bcp.psych.ualberta.ca)
Department of Psychology; University of Alberta
Edmonton, Alberta, CANADA T6G 2E9

Michael R.W. Dawson (mike@bcp.psych.ualberta.ca)
Department of Psychology, University of Alberta
Edmonton, Alberta, CANADA T6G 2E9

Abstract

PDP networks that use nonmonotonic activation functions often produce hidden unit regularities that permit the internal structure of these networks to be interpreted (Berkeley et al, 1995; Dawson, 1998; McCaughan, 1997). In some cases, these regularities are associated with local interpretations (Dawson, Medler & Berkeley, 1997). Berkeley has used this observation to suggest that there are fewer differences between symbols and subsymbols than one might expect (Berkeley, 1997). We suggest below that this kind of conclusion is premature, because it ignores the fact that regardless of their content, the local features of these networks are not combined symbolically. We illustrate this point with the interpretation of a network trained on a variant of Hinton's (1986) kinship problem, and show how the network's behavior depends on the coarse coding of information represented by hidden unit bands, even when these bands have local interpretations. We conclude that nonmonotonic PDP networks actually provide an excellent example of the differences between symbolic and subsymbolic processing.

Introduction

Networks of value units are a PDP architecture whose processors use a Gaussian activation function, and whose connection weights are trained using a variation of the generalized delta rule (Dawson & Schopflocher, 1992).

One property that emerges from this PDP architecture is a marked "banding" of its hidden unit activities (Berkeley et al., 1995; Dawson, 1998; Dawson et al., 1997). This banding is revealed when the responses of hidden units to each of a set of training patterns are plotted in a type of one-dimensional scatter plot called a jittered density plot (Chambers, Cleveland, Kleiner, & Tukey, 1983). One jittered density plot is drawn for each hidden unit in a network. For each pattern in a training set, a dot is added to the density plot. The x-position of the dot indicates the activity produced in that hidden unit by an input pattern. The y-position of the dot is randomly selected to reduce the overlap of different points. For the hidden units of a value unit network, the dots in a jittered density plot are not "smeared" uniformly across the graph. Instead, the plot is typically organized into a set of distinct bands or stripes (see Figure 1).

This banding phenomenon is important, because the bands often enable a researcher to determine the algorithm that is used by a trained network to accomplish a particular pattern recognition task. Training patterns that fall into the same band in a hidden unit do so because they share one or more properties, called *definite features* (Berkeley et al., 1995). By identifying the definite features in a layer of hidden units, and by determining how they are combined by a layer of output units, one can specify in great detail how a network of value units accomplishes a mapping from inputs to outputs.

For example, a network of value units has been trained on a set of logical problems devised by Bechtel and Abrahamsen (1991). When this network was analyzed, its hidden units were highly banded, and bands were associated with specific local features (e.g., type of logical connective, relations among variables in the logic problems). The network combined these local features in such a way that its internal structure represented many of the traditional rules of logic, such as *modus ponens* (Dawson et al., 1997).

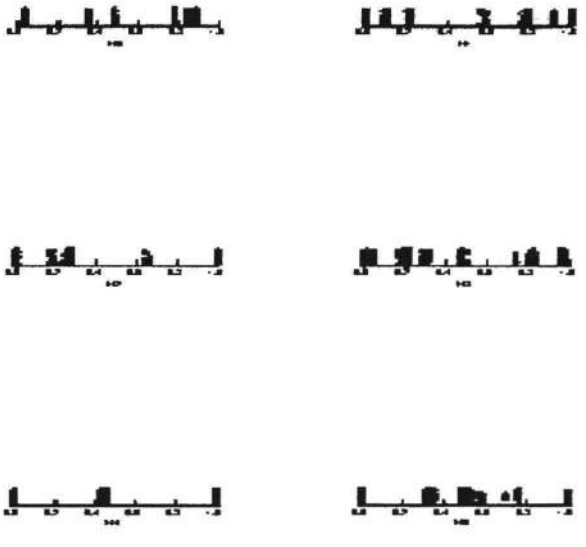

Figure 1. Jittered density plots for the kinship network described below. Each plot is for one of the 6 hidden units in that network. Each plot is comprised from 312 different points, which are organized into distinct bands for each hidden unit

Bands, Symbols, And Subsymbols

One major debate in cognitive science concerns potential differences (and similarities) between symbolic models and connectionist networks (Dawson, 1998). For example, Smolensky has argued that, in contrast to symbolic theories, PDP networks are *subsymbolic* (Smolensky, 1988). To say that a network is subsymbolic is to say that the activation values of its individual hidden units do not represent interpretable features that could be represented as individual symbols. Instead, each hidden unit is viewed as indicating the presence of a *microfeature*. Individually, a microfeature is unintelligible, because its "interpretation" depends crucially upon its context (i.e., the set of other microfeatures which are simultaneously present (Clark, 1993)). However, a collection of microfeatures represented by a number of different hidden units can represent a concept that could be represented by a symbol in a classical model.

It has recently been argued that the banding phenomenon found in value units is relevant to understanding the subsymbolic nature of PDP networks (Berkeley, 1997). This argument is based on an interpretation of a network trained to solve the logic problem (see also Dawson et al., 1997). Berkeley noted that the bands in the logic network are associated with a local interpretation (e.g., some bands represent which connective is present in a stimulus problem, while other bands represent relationships between specific variables in a stimulus problem, such as "Sentence 1 variable 2 is equal to the variable in the conclusion"). Berkeley also noted how such local features become interpretable (as symbols) only after considering a collection of individual hidden unit activations (i.e., a collection of individual dots which in turn produce a band of the sort depicted in Figure 1). Berkeley concluded that the fact that the bands in this network could be construed as being symbols under a liberal interpretation of the term (Vera & Simon, 1993), and suggested that the differences between symbols and subsymbols was smaller than one would believe from the extant literature.

Unfortunately, this conclusion is premature. This is because the "symbolic" nature of the bands in the logic network (i.e., a local interpretation denoting or representing a specific component of the logic problem, along the lines explored by Vera and Simon, 1993) are actually rarely seen in value unit networks. When most other examples of such networks are interpreted with the banding technique, we find that individual bands do not typically denote entities that would be represented as symbols in a classical theory. Instead, the bands themselves seem much more akin to subsymbols, and the "symbolic" interpretation of a network's internal structure only emerges after considering combinations of bands distributed over a number of different hidden units. Furthermore, even when the representations at the level of hidden units are local (e.g., in the logic network, when individual bands of activity could be assigned local interpretation, and as a result each hidden unit represented a collection of different local features), these local features are not combined into a more global response using symbolic operations.

To illustrate these points, let us consider the interpretation of a different value unit network, one which has been trained to solve a variation of a kinship problem originally reported by Hinton (1986) in the context of interpreting internal network representations.

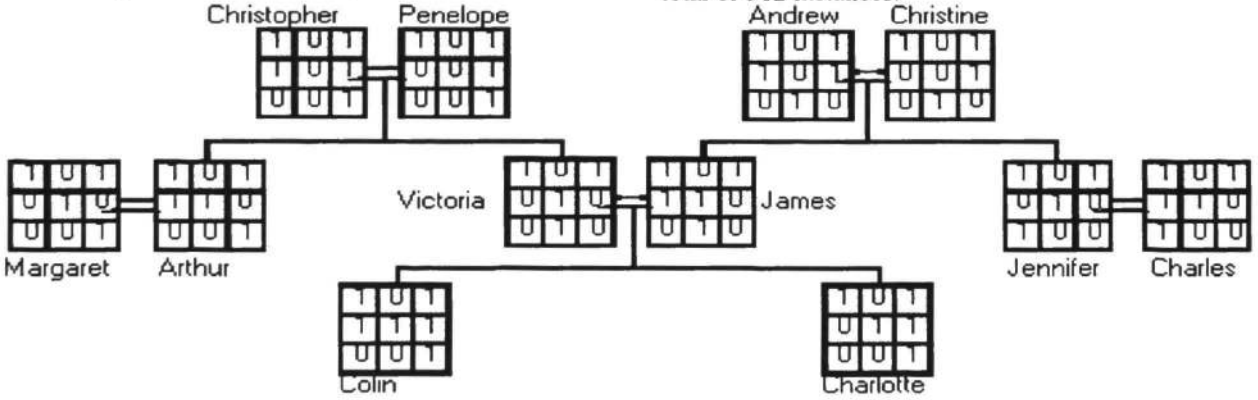

Figure 2. One of the 6 family trees modeled after Hinton (1986). The 9 bits at each node represent the binary code used to represent an individual in the network. The top 3 represent the family, the next three represent gender (white bit) and generation (gray bits), and the bottom 3 represent a code to distinguish individuals of the same generation (see text for more details).

Simulation

Problem Representation

In Hinton's kinship problem (Hinton, 1986), a network was given an individual's name and a relationship (e.g., "James, father"). This input represented a question about a person (i.e., "Who is James' father?"). The network's task was to generate the name or names representing the correct answer to the question (i.e., "Andrew").

In Hinton's original version of the problem, a network was trained on 100 of the 104 possible relationships in two different family trees of identical structure (i.e., the structure illustrated in Figure 2). In our version of this problem, we used six different versions of this family tree (i.e., six different families with the identical family tree structure), training the network on 52 relationships in each tree, for a total of 312 instances.

The network had 21 input units. The first 9 represented a person's name using the following coding scheme: The first

three bits indicated which of the six families the individual belonged to (001 = family 1, 010 = family 2, 011 = family 3, 100 = family 4, 101 = family 5, 110 = family 6). The fourth bit indicated whether the individual was male (activity = 1) or female (activity = 0). The fifth and sixth bits indicated the generation within the family tree to which the person belonged (01 = first generation, 10 = second generation, 11 = third generation). The seventh, eighth, and ninth bits were local codes that, in combination with gender bit 4, individuated different people belonging to the same generation of the family tree (see Figure 2). The advantage of the local code in these final bits is that the network could generate two names by turning two of these bits on, which is necessary when asked to name the aunts or uncles of Generation 3 children.

The remaining 12 input units of the network represented a relationship using Hinton's local coding scheme (Hinton, 1986). A relationship was encoded by turning one of these 12 units on and by turning the other 11 off. In order from input unit 10 to input unit 21 the represented relations were nephew, niece, aunt, uncle, brother, sister, father, mother, daughter, son, wife, and husband.

The network had 6 hidden units and 9 output units, all of which were value units. The 9 output units encoded an individual's name using the same coding scheme that was used to represent names in the input units.

In each family tree, there is a total of 52 different relationships that can be queried (4 nephew, 4 niece, 2 aunt, 2 uncle, 3 brother, 3 sister, 6 father, 6 mother, 6 daughter, 6 son, 5 wife, 5 husband). Note that there are only 2 aunt and 2 uncle queries because each of these queries results in the network generating a name output that represents two different individuals by turning two of the "local bits" on. Because we trained the network on these 52 relationships for 6 different family trees there was a total of 312 patterns in the training set.

Network Training

The network biases and connections were randomly selected from the range [-0.1,0.1], and the network was trained using a variation of the generalized delta rule developed for value unit networks (Dawson & Schopflocher, 1992) with a learning rate of 0.001 and a momentum of 0. Connection weights and biases were updated after every pattern presentation. During one sweep of training, each of the 312 training patterns was presented to the network. The order of pattern presentation was randomized before every sweep.

The network was said to have converged on a solution to the problem when a "hit" was recorded for the output unit for every pattern presented during the epoch. A "hit" was defined as output unit activity of 0.9 or greater when the desired output was 1.0, or as output unit activity of 0.1 or less when the desired output was 0.0. Convergence was achieved after 2734 sweeps.

Results

Network Interpretation

The jittered density plots that were presented in Figure 1 were actually plots for each of the 6 hidden units in the converged kinship network. It is apparent from these diagrams that there is marked banding in all six of these units. The interpretation of these bands was accomplished by using descriptive statistics to identify the definite unary and binary features in each of these bands in accordance with previously published methods (Berkeley et al., 1995). The interpretations of the definite features that were found are presented in Table 1.

From Table 1, it can be seen that two of the hidden units are completely devoted to representing which of the six possible family trees is being queried. Each of the six bands observed in hidden unit 0 is composed of stimulus questions about only one of the six families. For example, Band A contains all of the questions about family 3 (see Table 1 for more details). Similarly, each non-zero band in Hidden unit 4 contains questions about a specific family.

The network's discovery that some of the input bits correspond to family name is important, because the remaining hidden units can be used to represent regularities *within* the family tree structure. These regularities can be applied to all six of the family trees. Therefore, the regularities represented in the bands of the remaining four hidden units ignore the first three bits of any input name. Table 1 indicates that all four of the remaining hidden units have bands associated with specific definite features, all of which pertain to structure within the family tree, and which ignore the family feature.

Given the Table 1 account of the bands for the hidden units in this network, how does it solve the kinship problem? Qualitatively speaking, the network's algorithm appears to involve two different tasks. When asked a question like "*Who is person X's mother?*", the network uses two of its hidden units (i.e., units 0 and 4) to identify the family name that is required in the answer, and to write this family name into the first three output units by activating them appropriately. There does not appear to be much of a mystery about how this "writing" is done: hidden units 0 and 4 act as the bottleneck in a 3-2-3 encoder network. In such a network, the values of 3 input units are compressed into a 2-hidden unit representation; the hidden unit activity is then uncompressed to produce a copy of the input bits into the 3 output units.

The second task for the network is to identify the individual's name, and to "write" this into the remaining six output units. How this task is accomplished is much more mysterious, though, because the kind of definite features listed in Table 1 appear to refer to groups of people, and do *not* refer to individuals. How does the network utilise these general features to represent the identity of the individual whose name is to be "written" into the output units?

The answer to this question is that the network uses *coarse coding* to represent individuals (or more specifically, particular nodes in the family tree) using the Table 1 features. In general, coarse coding means that an individual processor is sensitive to a broad range of features, or at least

Table 1: Definite features for each band in each hidden unit. Beside each band label is the number of patterns that belong to that band. Key for definite features: F = father, M = mother, B = brother, Sr = sister, Sn = son, D = Daughter, W = wife, H = husband, Nc = niece, Np = nephew, U = uncle, A = aunt, G = generation, P = person, FG = female of generation, MG = male of generation.

UNIT	BAND	DEFINITE FEATURES
Hidden Unit 0	A *N=52*	Family 3
	B *N=52*	Family 1
	C *N=52*	Family 2
	D *N=52*	Family 5
	E *N=52*	Family 6
	F *N=52*	Family 4
Hidden Unit 1	A *N=156*	Not A and Not U
	B *N=36*	(Sn of G01 P001) or (A or U of G11 P001)
	C *N=18*	(H of FG10 P001) or (W of MG10 P001) or (B of F G10 P010)
	D *N=24*	(D of G01 P001) or (Sr of G10 P001) or (B of G10 P100)
	E *N=30*	(D of G01 P010) or (Sr or W or H of G10 P010) or (Sr or W or H of G10 P100)
	F *N=12*	Sn of G01 P010
	G *N=36*	(F or M or W or H of G10 P010) or (F or M of G11 P001
Hidden Unit 2	A *N=240*	No definite features
	B *N=12*	(M of FG10 P010) or (F of FG10 P010)
	C *N=24*	(H of FG01 P001) or (W of MG01 P001) or (M or F of MG10 P001)
	D *N=12*	(F of FG10 P100) or (M of FG10 P100)
	E *N=24*	(H or W of G01 P010) or (F or M of G10 P010)
Hidden Unit 3	A *N=66*	Not Np and Not Nc and Not U and Not Sn
	B *N=96*	Np or Nc or B or Sr or D
	C *N=24*	(Sn of G01 P001) or (Sn of G10 P010)
	D *N=36*	(W or H of G01 P010) or (F or M of G10 P010)
	E *N=6*	B of FG10 P010
	F *N=24*	(Sn of G01 P010) or (W or H of G10 P001)
	G *N=60*	(F or M or W or H of P001) or (F or M or W or H of P010)
Hidden Unit 4	A *N=156*	Family 2 or Family 3 or Family 4
	B *N=52*	Family 1
	C *N=52*	Family 6
	D *N=52*	Family 5
Hidden Unit 5	A *N=156*	Np or U or B or F or Sn or H
	B *N=72*	Nc or Sr or D or W
	C *N=12*	D of G01 P010
	D *N=6*	Sr of MG10 P010
	E *N=12*	(W of MG01 P001) or (W of MG P010)
	F *N=24*	(M of G10 P001) or (M of G11 P001) or (M of G01 P100)
	G *N=6*	W of MG01 P010
	H *N=12*	M of G01 P010
	I *N=12*	A of G11 P001

to a broad range of values of an individual feature (e.g., Churchland & Sejnowski, 1992, pp. 178-179). As a result, individual processors are not particularly useful or accurate feature detectors. However, if different processors have overlapping sensitivities, then their outputs can be pooled, which can result in a highly useful and accurate representation of a specific feature. Indeed, the pooling of activities of coarse-coded neurons is the generally accepted account of hyperacuity, in which the accuracy of a perceptual system is substantially greater than the accuracy of any of its individual components (e.g., Churchland & Sejnowski, 1992, pp. 221-233).

In the trained kinship network, each of the four hidden units that is not involved in representing a particular family tree is instead involved with the coarse coding of a particular node within a family tree. The network can pick out an individual node in the family tree by pooling (or combining, or intersecting) the coarse coded representation of the four hidden units.

To illustrate this, let us imagine that for any one of the family trees, we asked the network "*Who is the father of the female Person 2 Generation 2?*" (e.g., for the family tree given in Figure 2, the network would be asked "Who is Victoria's father?"). Ignoring hidden units 0 and 4 (which are concerned with picking out family trees, and not concerned with picking out relations within the tree structure), this query will produce activity that falls in Band A of hidden unit 1, Band B of hidden unit 2, Band D of hidden unit 3, and Band A of hidden unit 5.

Importantly, none of these bands picks out an individual node in the family tree by itself, as is revealed in Table 1. Hidden unit 1 Band A picks out 156 different individuals (across family trees) who are not aunts and not uncles. Hidden unit 2 Band B picks out 12 different individuals who are either the mother or the father of the female person 010 in the second generation. Hidden unit 3 Band D picks out the 36 different individuals who are the wife or husband of person 010 in generation 1, or who are the father or mother of person 010 in generation 2. Band A of hidden unit 5 picks out the 156 different individuals who are either nephews, uncles, brothers, fathers, sons, or husbands (i.e., any individual who is male).

While none of the bands by themselves pick out an individual, the *intersection* of the nodes picked out by each of these four bands selects the appropriate individual within the family tree: the only node pointed to by every one of these bands is the male Person 1 in Generation 1. This is the essence of coarse coding -- the overlap of the receptive fields of broadly tuned detectors can be used to represent finely detailed information.

Likewise, we could ask the network a very similar question: "*Who is the mother of the female Person 2 Generation 2?*" This question will produce the identical band activity in the network as was produced in the previous example, with one exception: it will produce activity in hidden unit 5 that falls in Band H, and not in Band A. Because of this change, the result of intersecting the subsets of nodes pointed to by all the bands changes: now, the only node pointed to by all of the bands is the female Person 1 in Generation 1.

Finally, let us consider the two hidden units that detect which of the 6 family trees is being queried. As was noted earlier, and as can be observed in Table 1, the bands for both of these units have very specific local interpretations. However, it is important to realize that their activities must also be pooled in order to "write" the correct family name into the appropriate output units. For instance, when a network is asked about a relationship for a person in Family 5, this will produce activity that falls in Band D of hidden unit 0 and that falls in Band D of hidden unit 4. Both of these bands must be active for the correct family output to be generated. For instance, if hidden unit 0 was ablated from the network, and the network was asked a question about Family 5, the activity of hidden unit 4 by itself would not produce the correct output in the network, even though the local interpretation of hidden unit 4's activity is "Family 5". For the network, the complete representation of family is a result of a distributed representation -- a combination of hidden unit 0 and hidden unit 4 activities.

Discussion

According to Smolensky (1988), subsymbols are constituents of traditional symbols. "Entities that are typically represented in the symbolic paradigm are typically represented in the subsymbolic paradigm by a large number of subsymbols" (p. 3). As a result, " it is often important to analyze connectionist models at a higher level; to amalgamate, so to speak, the subsymbols into symbols".

The analysis of the kinship network that was reported above is completely consistent with this view. To summarize this analysis, the following discoveries were made. First, the jittered density plots revealed a great deal of structure (i.e., bands). Second, the definite features of most of these bands did *not* correspond to a particular local concept (e.g., an individual's name, or the name of a particular relationship). Instead, the bands usually corresponded to disjunctions of general features that picked out sets of individuals (e.g., Hidden Unit 3 Band D), or in some cases a single feature shared by a large number of individuals (e.g., Hidden Unit 5 Band A's detection of "male"). Third, an account of how the network uses such broadly tuned representations to identify particular individuals relies on the notion of coarse coding. Specifically, the intersection of the sets of individuals represented in all of the bands in which the activity of an input pattern falls picks out a single individual, permitting the network to correctly respond to an input question. In short, the bands illustrated in Figure 1 appear to be acting as subsymbols, and the "symbolic" behavior of the network (i.e., its generation of an individual's name in its output units) depends upon the ability of the output units to combine -- to intersect -- the subsymbolic representations.

Results like these are relevant to the comparison between classical and connectionist architectures. Consider a recent attempt to incorporate situated action theories (including connectionism) into classical cognitive science. Vera and Simon (1993) argued that any situation-action pairing can be represented either as a single production in a production system, or (for complicated situations) as a set of produc-

tions. "Productions provide an essentially neutral language for describing the linkages between information and action at any desired (sufficiently high) level of aggregation" (p. 42).

Greeno and Moore (1993) take the middle road in their analysis of ALVINN, suggesting that "some of the processes are symbolic and some are not" (p. 54). Disagreements about what counts as a symbol are clearly at the heart of the debate that Vera and Simon initiated (Vera & Simon, 1994).

The problem with Vera and Simon's notion of what defines a symbol is that it focuses exclusively on the content that the symbol represents, and ignores the operations that are used to manipulate this information (e.g. symbolic concatenation, or the parsing of a string into symbolic constituents). The definition of a subsymbol in Smolensky's (1988) terms not only depends on content (i.e., what subsymbols might represent), but also upon the mechanisms for processing this content. Smolensky (p.3) notes that networks "participate in numerical not symbolic - computation." Similarly, Fodor and Pylyshyn (1988) have pointed out that "even on the assumption that concepts are distributed over microfeatures, '+ has-a-handle' is not a constituent of CUP in anything like the sense that 'Mary' (the word) is a constituent of (the sentence) 'John loves Mary'" (p. 21). This is exactly the position of connectionist critics who believe that Vera and Simon's (1993) definition of "symbol" is too liberal. For example, Touretzky and Pomerleau (1994) argue against Vera and Simon's symbolic reconstrual of a particular network, ALVINN, by noting that its internal features "are not arbitrarily shaped symbols, and they are not combinatorial. Its hidden unit feature detectors are tuned filters" (p. 348). (But for responses to this view, see also Greeno & Moore, 1993; Vera & Simon, 1994).

The coarse coding interpretation of the kinship network is a case study in the nonsymbolic processing of subsymbols, and thus illustrates an important difference between subsymbolic and symbolic accounts. Importantly, this processing difference holds true for value unit networks even when the content associated with bands is local (i.e., the family units discussed above, or the units of the logic network discussed by (Dawson et al., 1997)). When Berkeley used value unit bands to argue for similarities between symbols and subsymbols, he mistakenly focussed on the content of the bands themselves (Berkeley, 1997). As we have shown above, when one considers value unit banding in terms of represented content as well as the processes required to exploit this content, value unit banding provides an excellent example of Smolensky's (1988) subsymbolic level.

Acknowledgments

This work was supported by an NSERC Research Grant awarded to the second author.

References

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

Berkeley, I. S. N. (1997). What the #$*%! is a subsymbol? . Paper presented at the 1997 meeting of the Society For Exact Philosophy: Web version available at http://www.ucs.usl.edu/~isb9112/dept/phil341/subsymbol/subsymbol.html.

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.

Chambers, J. M., Cleveland, W. S., Kleiner, B., & Tukey, P. A. (1983). *Graphic methods for data analysis*. Belmont, CA: Wadsworth International Group.

Churchland, P. S., & Sejnowski, T. J. (1992). *The computational brain*. Cambridge, MA: MIT Press.

Clark, A. (1993). *Associative engines*. Cambridge, MA: MIT Press.

Dawson, M. R. W. (1998). *Understanding Cognitive Science*. Oxford, UK: Blackwell.

Dawson, M. R. W., Medler, D. A., & Berkeley, I. S. N. (1997). PDP networks can provide models that are not mere implementations of classical theories. *Philosophical Psychology, 10*, 25-40.

Dawson, M. R. W., & Schopflocher, D. P. (1992). Modifying the generalized delta rule to train networks of non-monotonic processors for pattern classification. *Connection Science, 4*, 19-31.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture. *Cognition, 28*, 3-71.

Greeno, J. G., & Moore, J. L. (1993). Situativity and symbols: Response to Vera and Simon. *Cognitive Science, 17*, 49-59.

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.

McCaughan, D. B. (1997, June 9-12). *On the properties of periodic perceptrons*. Paper presented at the IEEE/INNS International Conference on Neural Networks (ICNN'97), Houston, TX.

Smolensky, P. (1988). On the proper treatment of connectionism. *Behavioural and Brain Sciences, 11*, 1-74.

Touretzky, D. S., & Pomerleau, D. A. (1994). Reconstructing physical symbol systems. *Cognitive Science, 18*, 345-353.

Vera, A. H., & Simon, H. A. (1993). Situated action: A symbolic interpretation. *Cognitive Science, 17*, 7-48.

Vera, A. H., & Simon, H. A. (1994). Reply to Touretzky and Pomerlau: Reconstructing physical symbol systems. *Cognitive Science, 18*, 355-360.