On the Subsymbolic Nature of a PDP Architecture that Uses a Nonmonotonic Activation Function

MICHAEL R.W. DAWSON and C. DARREN PIERCEY
Biological Computation Project, Department of Psychology, University of Alberta, Edmonton, Alberta, Canada; E-mail: mike@bcp.psych.ualberta.ca

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; McCaughan, 1997; Dawson, 1998). In particular, when the responses of hidden units to a set of patterns are graphed using jittered density plots, these plots organize themselves into a set of discrete stripes or bands. In some cases, each band is associated with a local interpretation. On the basis of these observations, Berkeley (2000) has suggested that these bands are both subsymbolic and symbolic in nature, and has used the analysis of one network to support the claim that there are fewer differences between symbols and subsymbols than one might expect. We suggest below that this conclusion is premature. First, in many cases the local interpretation of each band is difficult to relate to the interpretation of a network's response; a more appropriate relationship only emerges when a band associated with one hidden unit is considered in the context of other bands associated with other hidden units (i.e., interpretations of distributed representations are more useful than interpretations of local representations). Second, the content that a band designates to an external observer (i.e., the interpretation assigned to a band by the researcher) can be quite different from the content that a band designates to the output units of the network itself.. We use two different network simulations – including the one described by Berkeley (2000) – to illustrate these points. We conclude that current evidence involving interpretations of nonmonotonic PDP networks actually illustrates the differences between symbolic and subsymbolic processing.

Key words: connectionism, PDP, representations, subsymbols, symbols

1. Introduction

One major debate in cognitive science concerns potential differences (and similarities) between symbolic models and connectionist networks (e.g., Dawson, 1998). For example, it has been argued that, in contrast to symbolic theories, parallel distributed processing (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.

Minds and Machines **11:** 197–218, 2001.

Smolensky's (1988) notion of "subsymbolic" processing was introduced as an alternative to the classical notion that the mind is a product of a physical symbol system (Newell, 1980). However, some researchers have argued that architectures that appear to be subsymbolic are actually symbolic in nature, and can be quite comfortably absorbed into a physical symbol system account. For example, consider a recent attempt to incorporate situated action theories (including connectionism) into classical cognitive science. Vera and Simon (1993) have 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 productions. "[Situated action] systems are symbolic systems" (p. 8).

Much of the position championed by Vera and Simon (1993) depends upon what some would call a fairly liberal definition of the term symbol. For Vera and Simon, the first major property of a symbol is that it is a pattern. This pattern can be compared to other symbols/patterns to be identified as being the same or different, and a physical symbol system's behavior depends on the outcome of this comparison. The second major property of a symbol is that it designates or denotes. This means that the symbol references some object (e.g., other symbols, patterns of sensory stimuli, motor actions); the physical symbol system can gain access to the referenced object via the designating symbol.

Disagreements about what counts as a symbol are at the heart of the reaction to Vera and Simon's (1994) position. Some researchers have called for a more restrictive definition of the term symbol. 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 and Moore, 1993; Vera and Simon, 1994)). Other researchers have sought a compromise between these views. For instance, 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). It appears the relationship between symbols and subsymbols is controversial, and is an issue that deserves further investigation.

1.1. Bands, symbols, and subsymbols

Recently, Berkeley (1997) has used a property of one type of PDP network, called networks of value units, to investigate the relationship between subsymbolic and symbolic descriptions. 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 and Schopflocher, 1992). One property that emerges from this PDP architecture is a marked "banding" of its hidden unit activities (Berkeley et al., 1995; Dawson et al., 1997; Dawson, 1998). 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

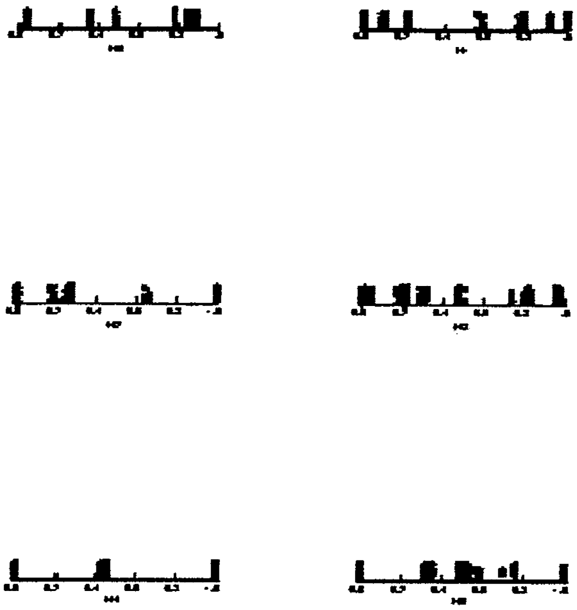


Figure 1. An example of banding injittered density plots of the hidden units of a value unit network. These particular plots are for the network that was trained in the first simulation which is reported later in the paper.

jittered density plot (Chamberset et al., 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 ajittered 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 can be important, because the bands sometimes 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 such features are exploited by output units, one can specify in great detail the kinds of features to which a particular hidden unit is sensitive.

For example, Berkeley et al. (1995) trained a network of value units to categorize a set of logic problems devised by Bechtel and Abrahamsen (1991). When this network (called L1O) 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).

As will be described in more detail below, Berkeley (2000) used the interpretation of the logic network to argue that subsymbolic and symbolic accounts are more similar in nature than one would expect from reading the existing literature:

> "However, if a broader definition of 'symbol', closer to Vera and Simon's (1993) conception is ultimately judged to be the most appropriate, then the evidence discussed above suggests that the difference between symbols and subsymbols may not be as great as has previously been supposed. Hopefully, future evidence from the analysis of trained connectionist networks will serve to provide evidence which will clarify the issues in the debate over the appropriate definition for symbols. In the meantime, it appears that the Berkeley et al. (1995) logic network analysis provides suggestive evidence for a *prima facie* case to be made that connectionist networks are, in fact, symbolic systems."

Our position is that such a conclusion is premature, for two general reasons.

First, the "symbolic" nature of the bands in the logic network (i.e., a local interpretation denoting or representing a specific content) is not often seen in value unit networks. When most other examples of such networks are interpreted with the banding technique, 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, much in the manner originally suggested by Smolensky (1988).

Second, the notion of designation used to assign content (definite features) to bands is not the same as the notion of designation used by Vera and Simon (1993) to define symbolhood. For Vera and Simon, designation is a property defined within the context of the operations of a physical symbol system. When a symbol designates an entity, it does so for the symbol system – the system accesses that entity (e.g., carries out some action) in virtue of the symbol's designation. In contrast, the interpretation of definite features relies on content associated with bands in such a way that the content is designated for some external observer of the system, and not for the system itself. These can lead to the situation in which the content assigned to the band by the external viewer does not really correspond to content that can be used to predict the network's behavior.

The purpose of this paper is to explore these matters in more detail, by considering two different PDP simulations. The first is a network of value units which has been trained to solve a variation of a kinship problem originally reported by Hinton (1986). The second is the network discussed by Berkeley (2000), which was originally reported by Berkeley et al. (1995). Both of these networks reveal distinct bands in their hidden value units. However, the interpretation of how the information revealed by such bands is used by both networks is highly context dependent – the functional role of the feature associated with one band depends

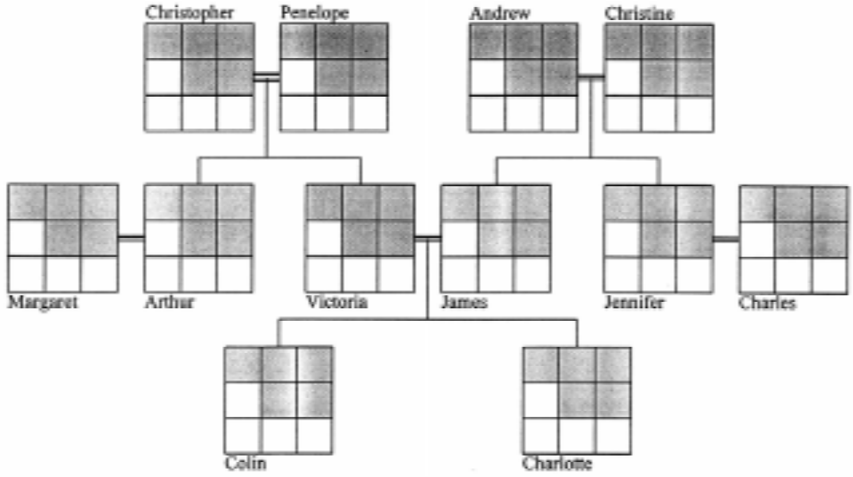


Figure 2. The structure of the kinship problem that was presented to the network.

strongly upon what other activation values are present in other hidden units in the network at the same time. In other words, the interpretation of network behavior requires considering definite features from one hidden unit in the context of definite features from others. With respect to the "cognitive science" of artificial neural networks, this does not support Berkeley's (2000) claim that the subsymbols of a PDP network seem to be symbolic in nature. Instead, interpretations of nonmonotonic PDP networks actually provide an excellent example of the *differences* between symbolic and subsymbolic processing.

2. Simulation 1: The Kinship Problem

2.1. METHOD

2.1.1. *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 tuming 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.

2.1.2. *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 leaming 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 randomised 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.

Table I. 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 *M*=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 601 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 F G10 P010) or (F of F G10 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 M G10 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

2.2. Results

2.2.1. *Network Interpretation*

The jittered density plots that were previously 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 can be 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 I.

From Table I, 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 I 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 I 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 I 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 I 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

I features. In general, coarse coding means that an individual processor is sensitive to a broad range of features, or at least to a broad range of values of an individual feature (e.g., Churchland and 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 highiy 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 and 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 I. 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 I. 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.

2.3. 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 sub symbolic representations realized as activation values of the hidden units.

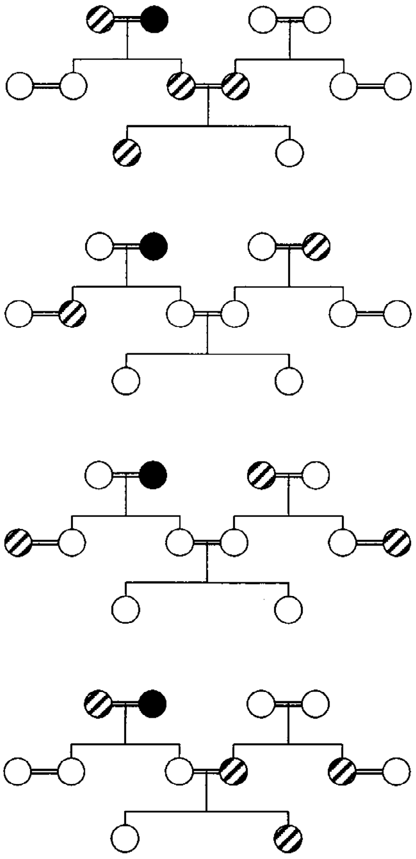


Figure 3. Coarse coding in the kinship network. Each plot indicates which individuals within a family tree belonged to a particular band in one of the hidden units. Note that each plot represents a set of individuals associated with each band. The intersection of these graphs indicates that only one individual is associated with each of the bands in question. See text for details.

3. Simulation 2: Lesioning the LIO Network for the Logic Problem

The preceding study described a network that had learned to solve one version of Hinton's (1986) kinship problem. The hidden value units of this network demonstrated marked banding, and the bands were associated with definite features. However, an examination of these definite features indicated that they revealed subsymbolic characteristics, and did *not* appear to be particularly symbolic in nature. This result provides an illustration of the subsymbolic nature of networks of value units.

Other networks of value units that have been interpreted support this general view – hidden value unit bands represent subsymbolic content, and symbolic interpretations of network behavior require considering the content of one band in the context of other bands in other hidden units. For example, Dawson et al., (2000) used a training technique called extra output learning to insert a symbolic theory into a network of value units. When the network was interpreted (to verify theory insertion), a precise relationship between the network and the symbolic theory only emerged when cluster analysis was performed on the basis of activations across all the hidden units in the network Similarly, Leighton (1999) trained a network of value units to solve the Wason (1966) card selection task, in which a subject must select cards to be examined to test a logical argument. In one version of the network trained to generate the logically correct responses to the task, pairs of hidden units cooperated to activate an output unit (i.e., to select a card for examination). The network behavior could not be understood by examining individual hidden units. Finally, Zimmerman (1999) trained a network of value units to generate solutions to the balance scale task used to study cognitive development (Inhelder and Piaget, 1958). In this task, the network is presented a configuration of weights on either side of a balance scale, and has to judge whether the scale would tip to the left, tip to the right, or balance. Zimmerman found that the network solved this task by performing a function approximation that required a coarse-coding combination of activities from all of the network's hidden units. Importantly, in all three of these examples, banding was found for the hidden units of the networks. However, the local interpretations of these bands could not adequately explain the network's behavior.

Results like those above are consistent with the view that hidden value unit bands are subsymbolic in nature. However, these results do not rule out the possibility that a network of value units can solve a problem by creating an internal representation in which its subsymbolic states (as revealed by banding analysis) are essentially symbolic in nature. lndeed, Berkeley (2000) has argued that the network of value units that he calls Lb, and which has been previously reported (Berkeley et al., 1995; Dawson et al., 1997) is one such network.

For example, two of the hidden units in L1O (units 6 and 8) reveal a striking set of bands that indicate that these two units are responsible for L10's detection of the connective in any presented logic problem (Berkeley et al., 1995, Figure 2). Hidden unit 6 has two distinct bands, one near 0 activity, the other near activity of 1. For logic problems that fall into the first band, the only definite feature shared is that the connective is *not* OR. All of the logic problems that fall into the second band share the definite feature that the connective is OR. So, it would appear that hidden unit 6 is an "OR detector". Hidden unit 8 has 3 distinct bands. The first falls around activity of 0; all of the logic problems that fall into this band share the definite feature that the connective is "OR". The second falls around activity of 0.11; all of the logic problems that fall into this band share the definite feature that the connective is "IF...THEN". The third falls around activity of 0.82; all of the logic problems that fall into this band share the definite feature that the connective is "NOT BOTH....AND". Berkeley (2000) points out that "The detailed analysis and interpretation of the bands of hidden unit 8 of the network makes it very clear that the function of this unit within the network was to act as a connective detector."

The existence of bands like those in hidden unit 8 of L10 provides the evidence that Berkeley (2000) uses to argue for similarities between subsymbolic and symbolic descriptions. After describing the bands of hidden unit 8, Berkeley states that "This must then be what Smolensky had in mind when he talked of subsymbols." However, the fact that these bands are associated with specific content suggests that they are also symbolic:

> "Moreover, these dots also fairly unambiguously satisfy Vera and Simon's (1993) condition of symbolhood. Individual dots in the jittered density plot of hidden unit 8 (the unit described earlier as a 'connective detector') 'designate or denote' particular main connectives in the problem set. This being the case, subsymbols fall within the class of 'representations', as Touretzky and Pomerleau (1993) use the term, but they fail to satisfy their conditions for symbolhood. This is not the appropriate place to attempt to adjudicate between the alternative definitions of 'symbol' proposed by Vera and Simon (1993), and Touretzky and Pomerleau (1993). However, if a broader definition of 'symbol', closer to Vera and Simon's (1993) conception is ultimately judged to be the most appropriate, then the evidence discussed above suggests that the difference between symbols and subsymbols may not be as great as has previously been supposed."

Importantly, the force of this argument depends on a specific interpretation of the notion of "designation" used by Vera and Simon (1993). For Berkeley (2000), a band designates a particular content if one can provide an interpretation to the definite feature or features associated with this band. We will call this notion "designation for the external observer", because this definition of describes what content a band designates to an observer examining the network from the outside.

However, other notions of designation are perhaps more plausible. For Vera and Simon (1993), designation is not from the perspective of an external observer, but is instead from the perspective of the information processing system that contains the symbols: "We call patterns symbols when they can designate or denote. An information system can take a symbol token as input and use it to gain access to a referenced object in order to affect it or be affected by it in some way" (p. 9). In this quote, it is clear that designation concerns what the information processing system itself can gain access to via its symbols.

The analysis reported by Berkeley et al. (1995) provides an account of what bands designate for the external observer. But what do the bands of units like hidden unit 6 and 8 denote for the L10 network itself? To answer this question, we took the original L10 network, and examined its behavior after lesioning its internal structure.

3.1. METHOD

3.1.1. *Problem Description*

Each pattern in Bechtel and Abrabmsen's (1991) 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 (MP), modus tollens (MT), alternative syllogism (AS), and disjunctive syllogism (DS)); there were two different versions of each AS and DS problem type.

Each argument was represented as pattern of on/off activity in a set of 14 input units 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 letters (A,B,C,D). A variable letter could also be negated (e.g., Not A). 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.

3.1.2. *Network Structure*

We studied the L10 network that was originally reported by Berkeley et al. (1995). This was a network of value units with 14 input units, 10 hidden value units, and 3 output value units that was trained to solve the Bechtel and Abrahamsen (1991) logic problem using a backpropagation procedure developed by Dawson and Schopflocher (1992). We took the connection weights from the L10 network and used them to construct a Microsoft Excel spreadsheet that could be used to observe L10's behavior when any of the logic problems were presented to it. Table 2 provides the connection weights and values for ji that were used to create this spreadsheet.

3.1.3. *Lesioning Procedure*

In all of the studies described below, at least one hidden unit was ablated from the original L10 network by "cutting" the connections from the hidden unit(s) to each of the three output units. This was accomplished by setting the weight of a "cut" connection to a value of 0. Three different lesioned networks were studied. In the first, hidden unit 6 was removed from the L10 network. In the second, hidden unit 8 was removed from the L10 network. In the third, both hidden units 6 and 8 were removed from the L10 network. We chose these two units for study because they had been previously identified as providing L10's ability to detect the connective in a presented logic problem (Berkeley et al., 1995).

Table II. The structure of (Berkeley et al., 1995) L10 network. (A) Bias of each hidden unit (μ) along with the connection weight to each hidden unit from the 14 different input units. (B) Bias of each output unit along with the connection weight to each output unit from the 10 different hidden units

Source	H0	H1	H2	H3	H4	H5	H6	H7	H8	H9
μ	−4.17E−02	3.74E−02	−1.12E−01	−2.21E−01	−2.39E−01	2.65E−01	4.91E−02	3.90E−02	1.91E−02	−2.45E−02
Input 0	9.18E−02	5.20E+00	−6.36E+00	4.04E+00	3.28E+00	−1.08E−01	4.22E−02	−3.08E−03	−1.08E−03	3.63E+00
Input 1	−3.34E+00	7.63E−01	2.09E+00	−2.97E+00	2.98E+00	1.15E+00	−6.85−02	2.33E−02	−4.27E−02	−1.29E+00
Input 2	−1.42E+00	−2.75E−01	1.50E+00	−1.62E+00	2.30E+00	1.84E+00	−4.48E−02	−4.72E−02	−8.84E−02	−2.87E−01
Input 3	2.17E−01	−3.94E+00	−4.14E−02	−7.13E−02	−5.61E−02	−7.36E−02	−1.48E+00	−5.20E−03	−2.20E−01	3.55E+00
Input 4	5.07E+00	7.68E−01	4.19E−01	−3.97E+00	2.54E-01	5.04E+00	8.60E−02	4.98E−01	1.08E+00	−2.41E+00
Input 5	−5.42E+00	5.97E−01	−5.41E−05	−1.12E−01	−4.23E−01	−4.66E+00	−1.65E−02	−1.02E−04	3.96E−03	4.21E−02
Input 6	3.38E+00	−5.15E−01	−2.45E+00	2.67E+00	−3.14E+00	−1.14E+00	−5.87E−02	1.92E−02	−4.58E−02	1.15E+00
Input 7	1.33E+00	3.71E−01	−1.20E+00	1.31E+00	−2.25E+00	−1.90E+00	−6.67E−02	−3.88E−02	−9.43E−02	7.94E−02
Input 8	1.66E−02	3.55E+00	5.36E+00	−1.68E−01	−4.31E−01	−4.66E+00	−7.61E−02	−8.89E−01	4.83E−03	−1.19E+00
Input 9	3.34E+00	5.34E−01	−2.10E+00	−2.69E+00	3.14E+00	1.13E+00	6.00E−02	−1.38E−02	3.86E−02	−1.12E+00
Input 10	1.41E+00	−3.29E−01	−1.49E+00	−1.34E+00	2.24E+00	1.90E+00	6.56E−02	3.79E−02	8.96E−02	1.97E−01
Input 11	−5.42E+00	−1.52E+00	−3.13E−03	3.90E+00	−3.29E+00	−1.16E−01	−4.73E−02	−6.07E−04	−7.50E−03	−4.89E+00
Input 12	−3.37E+00	−7.71E−01	2.44E+00	2.97E+00	−2.99E+00	−1.15E+00	6.70E−02	−1.92E−02	3.75E−02	1.26E+00
Input 13	−1.33E+00	3.16E−01	1.17E+00	1.60E+00	−2.32E+00	−1.85E+00	3.87E−02	4.24E−02	8.62E−02	−1.08E03

A

Source	Out 0	Out 1	Out 2
μ	2.32E−01	−1.31E−01	9.82E−02
H0	7.06E−01	6.15E−01	2.14E+00
H1	4.94E−04	−1.34E−02	4.14E+00
H2	−1.24E+00	−1.06E+00	−8.87E−01
H3	2.77E−03	1.92E−03	1.29E+00
H4	−7.04E−01	−8.61E−01	1.66E+00
H5	7.17E−01	8.94E−01	−1.70E+00
H6	5.55E−01	−3.29E−O1	−4.02E-01
H7	−4.84E−01	2.44E−01	9.12E−01
H8	−2.18E+00	2.32E+00	7.54E−02
H9	−3.27E−02	4.72E−03	1.21E+00

B

After a lesioned network had been produced by cutting the appropriate connections, each of the 576 logic problems was presented to the network, and the response of the network to each problem was recorded. These responses were then used to categorize network responses into eight different categories: invalid disjunctive syllogism (DSI), valid disjunctive syllogism (DSV), invalid modus ponens (MPI), valid modus ponens (MPV), invalid modus tollens (MTI), valid modus tollens (MTV), invalid alternative syllogism (ASI) and valid alternative syllogism (ASV). This was done by thresholding the observed activations of each output unit. If an activation was greater than 0.5, then it was assigned a value of 1. Otherwise, it was assigned a value of 0. Once the output values had been thresholded in this way, the categorized response of the network could be compared to desired response.

3.2. RESULTS

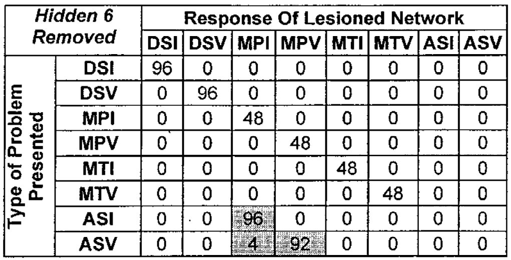
The behavior of each of the three lesioned versions of the L10 network was represented in an 8×8 confusion matrix (see Table III). Each row of the confusion matrix is associated with the known category of a problem presented to the network. Each column of the confusion matrix is associated with the observed category of the network's response. Each numerical entry in the matrix represents the number of times that a pattern of the type associated with the entry's row is identified as being a pattern of the type associated with the entry's column. Correct responses are represented in the entries along the diagonal of the matrix. Incorrect responses are represented in the off-diagonal entries of the matrix.

When hidden unit 6 is ablated from the L10 network, it becomes unable to correctly identify either invalid or valid alternative syllogisms, but correctly classifies all other problem types (see Table IIIA). This is consistent with the interpretation of this unit serving as an "OR detector", because OR is the connective used to define an alternative syllogism.

When hidden unit 8 is ablated from the L10 network, it has marked difficulty classifying disjunctive syllogisms, although it still correctly classifies some of these problems (see Table IIIB). 25 of the 96 DSI problems are correctly classified (26% accuracy), and 24 of the 96 DSV problems are correctly classified (25% accuracy). Interestingly, for all of the problems that are misclassified by this lesioned network, all of the mistakes are made with respect to connective type. The network never makes a mistake on problem validity. For instance, while it frequently mistakes a DSI problem as being an MPI problem, it never classifies a DSI problem as being of type MPV. This network never misclassified any of the other six types of logic problems.

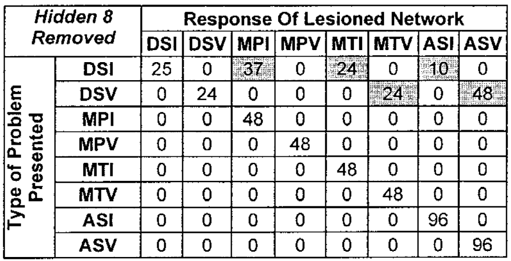
When both hidden units 6 and 8 are ablated from the network (Table IIIC), the results are almost identical to summing the effects of the individual lesions (Table IIIA and IIIB). These lesions cause the network to misclassify about 75% of the disjunctive syllogisms, while maintaining a correct judgement about the validity of these problems. The network never misclassifies a modus ponens or a modus

Table III. Confusion matrices representing response of the L10 logic network after (A) hidden unit 6 has been removed, (B) hidden unit 8 has been removed, and (C) both hidden units 6 and 8 have been removed. The eight response categories reflect problem type and validity (DS = disjunctive syllogism, MP = modus ponens, MT = modus tollens, AS = alternative syllogism, I = invalid, V = valid). Each number in the table indicates the frequency of times that an input problem of the type corresponding with the cell's row was classified as being of the type corresponding with the cellŠs column. Off-diagonal numbers that are greater than 0 (shaded cells) reflect errors generated by lesioning the network



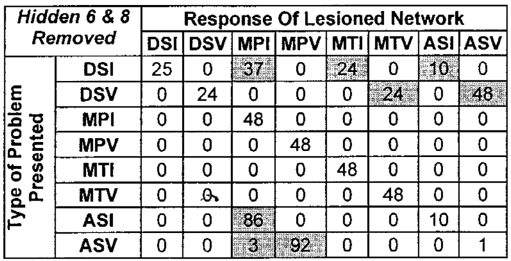
Hidden 6 Removed		Response Of Lesioned Network							
		DSI	DSV	MPI	MPV	MTI	MTV	ASI	ASV
Type of Problem Presented	DSI	96	0	0	0	0	0	0	0
	DSV	0	96	0	0	0	0	0	0
	MPI	0	0	48	0	0	0	0	0
	MPV	0	0	0	48	0	0	0	0
	MTI	0	0	0	0	48	0	0	0
	MTV	0	0	0	0	0	48	0	0
	ASI	0	0	96	0	0	0	0	0
	ASV	0	0	4	92	0	0	0	0

A



Hidden 8 Removed		Response Of Lesioned Network							
		DSI	DSV	MPI	MPV	MTI	MTV	ASI	ASV
Type of Problem Presented	DSI	25	0	37	0	24	0	10	0
	DSV	0	24	0	0	0	24	0	48
	MPI	0	0	48	0	0	0	0	0
	MPV	0	0	0	48	0	0	0	0
	MTI	0	0	0	0	48	0	0	0
	MTV	0	0	0	0	0	48	0	0
	ASI	0	0	0	0	0	0	96	0
	ASV	0	0	0	0	0	0	0	96

B



Hidden 6 & 8 Removed		Response Of Lesioned Network							
		DSI	DSV	MPI	MPV	MTI	MTV	ASI	ASV
Type of Problem Presented	DSI	25	0	37	0	24	0	10	0
	DSV	0	24	0	0	0	24	0	48
	MPI	0	0	48	0	0	0	0	0
	MPV	0	0	0	48	0	0	0	0
	MTI	0	0	0	0	48	0	0	0
	MTV	0	0	0	0	0	48	0	0
	ASI	0	0	86	0	0	0	10	0
	ASV	0	0	3	92	0	0	0	1

C

tollens problem. Finally, the network has great difficulty classifying alternative syllogisms. Interestingly, though, when both hidden units 6 and 8 are lesioned, the performance of this network on alternative syllogisms actually improves compared to the case of when only hidden unit 6 is removed. With only one hidden unit ablated, the network had 0% accuracy on both ASI and ASV problems. However, when hidden unit 8 is also removed, it correctly classifies 10 of the 96 ASI problems (10% accuracy) and 1 of the 96 ASV problems (1% accuracy).

4. Discussion

At first glance, the behavior of the lesioned versions of the L10 network are consistent with the expectations derived from assigning interpretations to the bands of hidden units 6 and 8. Ablating these units causes the network to have difficulty classifying particular types of problems, and these difficulties appear to be related to an inability to identify the connective in a presented logic problem. However, a closer examination of the confusion matrices presented in Table III suggests that connective detecting is actually more complex for L10 than the local interpretations of the bands would predict. In short, what these bands designate for the external observer is actually quite different from what they designate to the network itself.

First, consider hidden unit 6. As was noted earlier, the interpretations of its bands indicate that this unit serves as an “OR detector”. When this unit is removed from L10, it became unable to correctly identify either valid or invalid alternative syllogisms, which are the only problem types in the training set that use the connective “OR”. However, “OR detection” – as far as the network is concerned – cannot only be mediated by hidden unit 6. This is because when both hidden unit 6 and hidden unit 8 are absent from L10, its ability to categorize alternative syllogisms actually improves! This suggests that the ability to detect “OR” is not simply housed in hidden unit 6.

Second, consider hidden unit 8. The interpretations of its bands suggest, to the observer that it is a connective detector. It basically adopts three different levels of activity, and each level designates one of the three connectives used in the logic problems that comprise the training set. However, one would arrive at a completely different interpretation of its functional role on the basis of the errors produced by L10 when hidden 8 is removed. On the one hand, the network only has problems with disjunctive syllogisms, indicating that instead of being a general connective detector, hidden unit 8 is a “NOT BOTH...AND detector”. On the other hand, this interpretation cannot be completely correct, because the network is still able to correctly identify approximately 25% of both the DSI and the DSV problems. At the very least, this behavioral evidence suggests that this unit is involved in detecting “NOT BOTH... AND” for some, but not all, of the disjunctive syllogisms. This kind of interpretation is quite a bit different from the one arrived at by examining the three bands of this unit – there is no behavioral evidence from the lesioning experiments that hidden unit 8 is responsible for designating each of the three connective types to the network.

Third, consider one aspect of L10's performance that is not affected by these lesions: its categorization of modus ponens and modus tollens problems. From the interpretation of hidden unit 8's bands, one would predict that one of its functions is to detect the connective "IF...THEN". However, lesioning this unit, with or without lesioning hidden unit 6, produces absolutely no change in the network's ability to process logic problems that use this connective. Clearly, hidden unit 8 is not – as far as the network is concerned – a general connective detector, and the network's ability to detect "IF...THEN" must be mediated by other processors than the two that were lesioned in this study.

As a matter of fact, there is a good deal of evidence in the L10 network that functions like connective detecting are distributed across a number of different hidden units, and that a symbolic account of L10's behavior cannot be determined by providing interpretations to individual bands, but instead requires identifying regularities distributed throughout the network.

For instance, Berkeley et al. (1995) provided definite features for most of the bands identified in the L10 network (Table II). We have already reviewed the interpretation of the bands for hidden units 6 and 8, which are associated with the different connectives involved in the logic problems. Importantly, these connectives also emerge as definite features in several other bands in the other hidden units of this network. For example, the presence of the connective "IF...THEN" is one of the definite features associated with hidden unit 0 bands B and C, and with hidden unit 4 band B. As well, the presence of the connective "NOT BOTH...AND" is one of the definite features associated with hidden unit 2 band C, hidden unit 4 band C, hidden unit 5 band B, and hidden unit 7 bands A and C. Similarly, the presence of the connective "OR" is one of the definite features associated with hidden unit 3 band B, and with hidden unit 4 band D. In most of these cases, the presence of a connective is only one of the definite features associated with the band; usually other features involving properties of the variables are also present. This evidence suggests that the bands of L10 are actually picking out complicated subsets of logic problems, much in the same way that the bands in the kinship network were picking out subsets of individuals within a family tree.

If this is the case, then one would predict that a symbolic account of L10 would emerge by seeking regularities distributed across bands in different hidden units, which would be completely consistent with Smolensky's (1988) notion of the relationship between subsymbols and symbols This is exactly what happens with Lb. When the internal states of the L10 network are examined across hidden units, one can identify a small number of rules for identifying valid instances of the different problems (Berkeley et al., 1995; Berkeley, 2000; Dawson et al., 1997). The majority of these rules are classical in nature. In other words, even the L10 network supports the claim that the bands of hidden value units are subsymbolic in nature, and that symbolic properties of these networks are represented as patterns distributed across bands in different units.

5. General Discussion

This paper has described the results of two different simulation studies that have used network interpretation as a tool with which to explore the relationship between subsymbols and symbols. In the first simulation, the jittered density plots of hidden units of a network trained to solve a kinship problem revealed distinct bands. These bands were associated with definite features. However, the definite features that could be assigned to the bands were very difficult to relate to the desired output of the network. Instead of describing particular individuals, these features were characteristics of sets of individuals. Their role in determining how the network solved the problem – that is, how the network picked the name of an individual – only emerged after considering them from as components of a distributed representation. The functional role of one hidden unit's activity depended crucially on the activities that were simultaneously present in the other hidden units.

In the second simulation, we examined the effects of removing "connective detecting" units from a network trained to solve a particular logic problem. These units had a known local interpretation that was derived by using a previously derived analysis of hidden unit bands (Berkeley et al., 1995). The question of interest was the degree of correspondence between the interpretation of the units based on this technique, versus the interpretation that would be derived on the basis of observed responses in the damaged network. The behavior of the lesioned network did not show strong agreement with the local interpretation, suggesting that as far as the network's outputs were concerned, it was not correct to view these units as detecting specific connectives. Instead, the representation of connective in the logic problem appears to be distributed across a number of different hidden units.

Together, in combination with other results (Dawson et al., 1997; Leighton, 1999; Zimmerman, 1999; Dawson et al., 2000), these findings are consistent with the notion that the bands that are often revealed in jittered density plots of value units are subsymbolic in nature. If a network of value units can be described using a symbolic vocabulary, then its symbolic regularities emerge from a distributed representation – that is, by combining bands from different hidden units. The notion of a symbolic account emerging from the combination of subsymbolic information that is distributed throughout a PDP network is completely consistent with Smolensky's (1988) account of subsymbolic processing. These results are not consistent with Berkeley's (2000) notion that individual bands of a value unit are both subsymbolic and symbolic.

There are two methodological implications of these results. The first concerns the examination of subsymbolic representations in PDP networks. If value unit bands are indeed subsymbolic, then the analysis of these bands using the Berkeley et al. (1995) technique can provide a great deal of information about what kind of subsymbolic features are being detected by hidden units. The second concerns the examination of symbolic behavior in PDP networks. If the value unit bands are subsymbolic, then the local analysis of individual bands is not an appropriate technique to use in an attempt to generate symbolic descriptions. Instead, an attempt must be made to identify regularities distributed across hidden units. One technique that has been used to do this with some success involves the cluster analysis of vectors of hidden unit activities, coupled with the identification of definite features associated with each cluster (Dawson et al., 2000).

Acknowledgements

This research was supported by NSERC Research Grant A2038 awarded to the first author. Address correspondence to Dr. Michael Dawson, Biological Computation Project, Department of Psychology, University of Alberta, Edmonton, Alberta, Canada T6G 2P9.

References

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

Berkeley, I.S.N. (2000), What the #$*%! is a subsymbol?. *Minds and Machines*, 10, pp. 1–14.

Berkeley, I.S.N., Dawson, M.R.W., Medler, D.A., Schopflocher, D.P. and Hornsby, L., (1995). Density plots of hidden value unit activations reveal interpretable bands. *Connection Science* 7 pp. 167–186.

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

Churchland, P.S. and 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. and 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., Medler, D.A., McCaughan, D.B., Willson, L. and Carbonaro, M. (2000), Using extra output learning to insert a symbolic theory into a connectionist network. *Minds And Machines*, 10, pp. 171–201.

Dawson, M.R.W. and Schopflocher, D.P. (1992), Modifying the generalized delta rule to train networks of nonmonotonic processors for pattern classification. *Connection Science* 4, pp. 19–31.

Greeno, J.G. and 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.

Inhelder, B. and Piaget, J. (1958). The Growth Of Logical Thinking From Childhood To Adolescence. New York, NY: Basic Books.

Leighton, J.P. (1999), An alternate approach to understanding formal reasoning: Thinking according to the inductive-coherence model. Unpublished Ph.D., University of Alberta, Edmonton.

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.

Newell, A. (1980). Physical symbol systems. Cognitive Science, 4, 135–183.

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

Touretzky, D.S. and Pomerleau, D.A. (1994), Reconstructing physical symbol systems. Cognitive Science. 18, pp. 345–353.

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

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

Wason, P.C. (1966). *Reasoning*. New York: Penguin.

Zimmerman, C.L. (1999), *A network interpretation approach to the balance scale task*. Unpublished Ph.D., University of Alberta, Edmonton.