INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center St. • Suite 600 • Berkeley, California 94704-1198 • (510) 643-9153 • FAX (510) 643-7684

A Spatiotemporal Connectionist Model of Algebraic Rule-Learning

Lokendra Shastri* **and Shawn Chang**

TR-99-011

July 1999

Abstract

Recent experiments by Marcus, Vijaya, Rao, and Vishton suggest that infants are capable of extracting and using abstract algebraic rules such as "the first item X is the same as the third item Y". Such an algebraic rule represents a relationship between placeholders or variables for which one can substitute arbitrary values. As Marcus et al. point out, while most neural network models excel at capturing statistical patterns and regularities in data, they have difficulty in extracting algebraic rules that generalize to new items. We describe a connectionist network architecture that can readily acquire algebraic rules. The extracted rules are not tied to features of words used during habituation, and generalize to new words. Furthermore, the network acquires rules from a small number of examples, without using negative evidence, and without any pretraining. A significant aspect of the proposed model is that it identifies a sufficient set of architectural and representational conditions that transform the problem of learning algebraic rules to the much simpler problem of learning to detect coincidences within a spatiotemporal pattern. Two key representational conditions are (i) the existence of nodes that encode serial position within a sequence and (ii) the use of temporal synchrony for expressing bindings between a positional role node and the item that occupies this position in a given sequence. This work suggests that even abstract algebraic rules can be grounded in concrete and basic notions such as spatial and temporal location, and coincidence.

* *E-mail:* {shastri,shawnc}@icsi.berkeley.edu

Introduction

Recent experiments by Marcus et al. [9, 8] suggest that 7-month old infants may be capable of extracting algebraic rules and using them to distinguish between different speech stimuli. In controlled experiments, Marcus et al. habituated each infant to sequences (sentences) of nonsense syllables (words) conforming either to the pattern ABA, or to the pattern ABB (e.g. "ga ti ga" or "ga ti ti"). Subsequently, each infant was presented with additional sentences consisting entirely of new words, with half the sentences conforming to the ABA pattern, and half to the ABB pattern. It was found that infants habituated to the ABB patterns showed a marked preference for the ABA pattern, and infants habituated to the ABA pattern showed a marked preference for the ABB pattern.

The experimental results lead Marcus et al. to conclude that infants are capable of extracting and using abstract — algebraic rules — such as "the first item X is the same as the third item Y"[1]. Such an algebraic rule represents a relationship between placeholders or *variables* for which one can substitute arbitrary values; the rule does not particularly care about the statistical properties and features of the words in a sentence, it only cares that the first and the third words be the same. The experimental results also suggested that infants are able to extract algebraic rules rapidly from small amounts of data.

Marcus et al. point out that while popular neural network models excel at capturing statistical patterns and regularities in data, they are incapable of extracting abstract algebraic rules that generalize to new items[2]. They noted, however, that certain types of connectionist architectures (e.g., [15, 7]) that encoded relationships between variables could capture their findings.

We describe a connectionist model derived from [12, 15] which can readily acquire abstract algebraic rules from a small number of exemplars, and respond in a rule-governed manner to new items.

One significance of the proposed model is that it identifies a sufficient set of architectural and representational conditions that enable a connectionist network to extract and use algebraic rules. Furthermore, the model demonstrates that an appropriate choice of representation can reduce the problem of learning an algebraic rule to one of learning to detect appropriate *coincidences* within a spatiotemporal pattern.

In addition to extracting algebraic rules that do not depend on the features of items occurring in the stimuli, the proposed model can also extract more complex types of algebraic rules which are sensitive to features of the stimuli. For example, in previous work [12], an analogous network architecture was shown to be capable of extracting rules such as:

> "if x walks into y and if x is *animate* and y is *solid* then x gets hurt"

from examples and applying such rule to situations involving novel entities.

In what follows we describe the approach, the network architecture, and simulation results.

[1] The experimental design ruled out other explanations for the infants' ability to differentiate between the two patterns, such as counting syllables, computing transitional probabilities, or counting reduplications (e.g. BB in ABB).

[2] See [11, 10, 3] for a rebuttal and [8] for a reply.

Representational and Architectural Consideration

The model makes the following representational and architectural assumptions:

1. There exist nodes that encode *serial position* within a sequence. We will refer to such nodes as *positional role* nodes. Recent findings suggest that the existence of such nodes is biologically plausible [1].

2. The network can express *bindings* between a positional role node and the item that occupies this position in a given sequence, that is, the network is capable of encoding which item occurs in which position.

3. The bindings are expressed via *temporal synchrony*, that is, the occurrence of an item A in a particular position $P1$ in a sequence is coded by the synchronous activity of the cells coding the item A and the cells coding the positional role $P1$. There is considerable evidence that synchronization of neural activity might underlie the encoding of dynamic bindings [16, 2, 17, 15].

4. Nodes representing positional roles and items are interconnected via recurrent connections mediated by intermediate (or hidden) layers of nodes.

Task Formulation

Our simulation follows the general format of the experiments performed by Marcus et. al. The simulation consists of a habituation (training) phase followed by a testing phase. In the habituation phase, the network is exposed to sentences conforming to the habituation pattern. The network's "task" is to *attend* to these sentences. This is realized by presenting a sentence to the network and executing a gradient descent learning algorithm to modify network link weights such that the activation levels of appropriate nodes in the network match the representation of the sentence.

In the test phase, a sentence is presented to the habituated network and the mismatch between the network's "expectation" and the sentence is computed. It is assumed that this mismatch is a measure of the network's degree of *surprise* (or failure of expectation) upon seeing the test sentence, and hence, it corresponds to the time an infant might attend to the test sentence in the Marcus et al. experiments.

Feature Based Word Encoding

Each word is encoded using twelve binary features as described in [5] (see Table 1). This representation ensures that any feature distinguishing A and B words in the test phase, does not distinguish A and B words during the habituation phase. In other words, any feature contrast in the test words is absent in the habituation words. Furthermore, distinct words are used in the habituation and test phases.

Network Architecture

The network architecture is analogous to that used in [12] for learning relational rules and is shown in Figure 1. The network consists of three positional role nodes $P1$, $P2$, and $P3$,

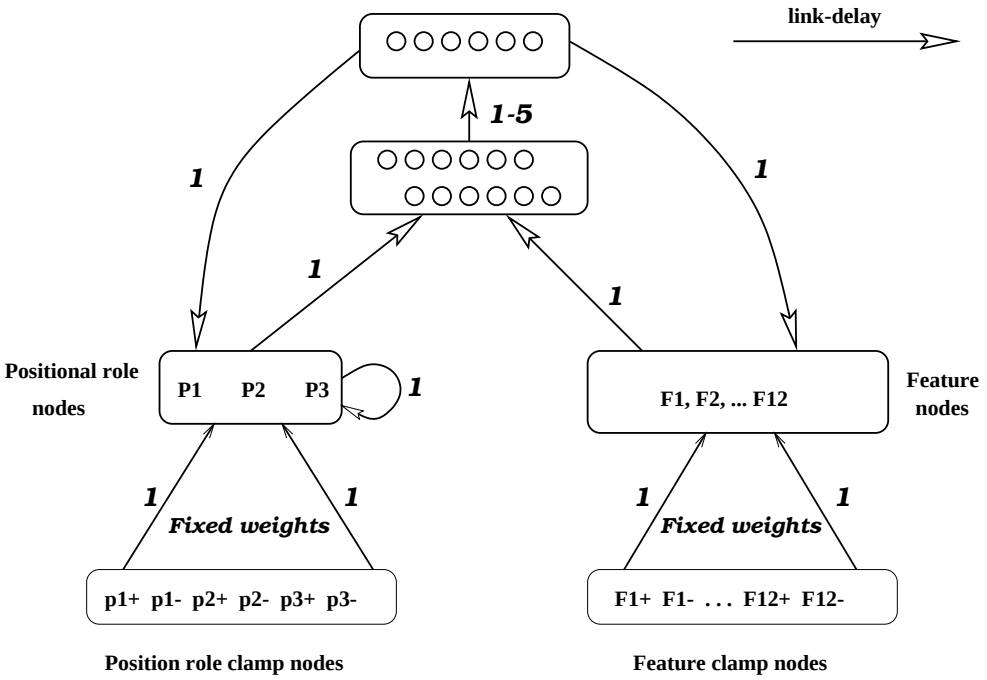

Figure 1: Network Architecture. The numbers along the links indicate link-delays. The network consists of three positional role nodes $P1, P2$, and $P3$, twelve feature nodes $F1, F2, \ldots F12$, and two hidden layers — consisting of twelve and six nodes, respectively. The positional role and feature nodes receive links from *clamp* nodes that serve to convey the stimuli to the network. These links have fixed weights and do not get modified during learning. There exists a pair of clamp nodes (+,−) for each role and each feature. Each pair of clamp nodes (+,−) is connected to the associated role or feature node. The activation of the '+' node indicates activation of the associated role or feature, while the activation of the '−' node indicates the inactivity of the associated role or feature. The inactivity of both '+' and '−' nodes indicates that the activation level of role or feature is unspecified. Each positional role node and each feature node is connected to every node in the first hidden layer by a link of delay 1. Each node in the first hidden layer is connected to each node in the second hidden layer by multiple links having delays of 1 through 5. Each node in the second hidden layer, in turn, connects to each positional role node and each feature node by links having a delay of 1. Finally, positional role nodes have lateral connections to one another with link delays of 1. All nodes have a sigmoidal transfer function.

Syllables (words) used during habituation.

de	-	-	-	-	-	+	-	+	-	+	-	+
di	-	-	-	-	-	+	-	+	-	+	-	-
je	-	-	+	-	-	+	-	+	-	+	-	+
ji	-	-	+	-	-	+	-	+	-	+	-	-
le	-	-	+	+	-	+	-	+	-	+	-	+
li	-	-	+	+	-	+	-	+	-	+	-	-
we	+	+	+	+	-	-	-	+	-	+	-	+
wi	+	+	+	+	-	-	-	+	-	+	-	-

Syllables (words) used during testing.

ba	-	-	-	-	+	-	-	+	+	-	+	+
po	-	-	-	-	+	-	-	-	+	+	-	+
ga	-	-	-	-	-	-	+	+	+	-	+	+
ko	-	-	-	-	-	-	+	-	+	+	-	+

Table 1: Feature representation of syllables (words) used in the training and testing phases.

twelve feature nodes $F1, F2, \ldots F12$, and two hidden layers — consisting of twelve and six nodes, respectively. All these nodes have a sigmoidal transfer function. The positional role and feature nodes receive links from *clamp* nodes that serve to convey the stimuli to the network. These links have fixed weights and do not get modified during learning. There exists a pair of clamp nodes (+,–) for each role and each feature. Each pair of clamp nodes (+,–) is connected to the associated role or feature node. The activation of the '+' node indicates activation of the associated role or feature, while the activation of the '–' node indicates the inactivity of the associated role or feature. The inactivity of both '+' and '–' nodes indicates that the activation level of role or feature is unspecified. It is important to encode such an unspecified state since, in general, we would only like to specify the activity of relevant roles and features and leave the state of all other nodes unspecified.

Each positional role node and each feature node is connected to every node in the first hidden layer by a link of delay 1. Each node in the first hidden layer is connected to each node in the second hidden layer by multiple links having delays of 1 through 5. The use of multiple links with extended delays from the first to the second hidden layer allows nodes in second hidden layer to integrate information over a wider temporal window, and respond to conjunctions of "features" computed by nodes in the first hidden layer (for a detailed discussion of the architecture refer to [12]). Each node in the second hidden layer, in turn, connects to each positional role node and each feature node by links having a delay of 1. Finally, positional role nodes have lateral connections to one another with link delays of 1.

Unlike many popular neural network architectures such as feedforward networks and simple recurrent networks [4], the proposed recurrent architecture does not have designated "output" nodes. Instead the nodes in a network can be classified as observable nodes, clamp nodes, and hidden nodes. Observable nodes correspond to "objects" in the model's

P1	P2	P3	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
+ -	+ -	+ -	+ -	+ -	+ -	+ -	+ -	+ -	+ -	+ -	+ -	+ -	+ -	+ -
1 0	0 1	0 1	0 1	0 1	0 1	0 1	0 1	1 0	0 1	1 0	0 1	1 0	0 1	0 1
1 0	0 1	0 1	0 1	0 1	0 1	0 1	0 1	1 0	0 1	1 0	0 1	1 0	0 1	0 1
0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
0 1	1 0	1 0	0 1	0 1	0 1	0 1	0 1	1 0	0 1	1 0	0 1	1 0	0 1	1 0
0 1	1 0	1 0	0 1	0 1	0 1	0 1	0 1	1 0	0 1	1 0	0 1	1 0	0 1	1 0
0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0	0 0
...														
block repeats 5 times														

Table 2: An example of input encoding showing setting of clamp nodes for the ABB pattern "di de de". Each horizontal line corresponds to a time-slice.

ontology. In the present example, this corresponds to positional roles and word features. Clamp nodes provide external inputs to observable nodes (external being anything outside the network). Hidden nodes are intermediate nodes that serve to encode rules/mappings between observable nodes.

Word and Sentence Presentation to the Network

Each sentence is presented to the network as a pattern of activity wherein the binding between a position in the sentence and the word occurring in that position is expressed via the synchronous activity of the associated positional role node and the feature nodes of the word. In keeping with the temporal synchrony approach for encoding dynamic bindings (Malsburg, 1986; Shastri & Ajjanagadde, 1993), each distinct item (in this case, a word) is assumed to fire in a distinct "phase". It is assumed that this binding is established as a result of sensory processing and the network being modeled is situated downstream from such processing.

The pattern of activity corresponding to a particular ABB pattern "di de de" is illustrated in Table 2. Each horizontal line corresponds to a time-slice and shows the state of all the clamp nodes during that time-slice. In the first line, the first positional role node (P1) is turned on by setting its + and - clamp nodes to 1 and 0 respectively. The positional role nodes for the second and third position (P2 and P3) are turned off by setting the + and – clamp nodes for these roles to 0 and 1, respectively. At the same time, feature nodes F1 ... F12 are turned on or off in accordance with the encoding of "di" by setting their clamp nodes to suitable states. This pattern is repeated in time-slice 2 [3]. This is followed by a time-slice where there is no activity in order to achieve phase-separation between the firing of distinct words. The fourth and the fifth lines encode the activity and positional bindings of the word "de" which occurs in the second and the third position. Here P1 is set off and both P2 and P3 are set on, while feature nodes F1 ... F12 are turned on or off in accordance with the encoding of "de". The above block of inputs is repeated several times

[3]Here the number 2 in not significant. In general, this pattern is repeated a small number of times.

```
P1     P2     P3
------------------
0.99   0.01   0.01
0.99   0.01   0.01
0.01   0.01   0.01
0.01   0.99   0.99
0.01   0.99   0.99
0.01   0.01   0.01
...
block repeats five times
```

Table 3: An example of target encoding. Target activations of positional role nodes for the ABB pattern "di de de". Each horizontal line corresponds to a time-slice. The targets are delayed by seven time-slices to account for the propagation of activity through the hidden structure.

to complete one presentation of the sentence.

During the habituation (i.e., training) phase, a target (i.e., a desired response) is also specified for positional role nodes (see Table 3). This target specifies that the activity of positional role nodes should represent the input sentence. The targets are, however, delayed to account for the propagation time through the hidden layers. The difference between the specified target activity and the actual activity computed by the network drives a gradient descent algorithm which changes link weights in order to minimize this difference.

The network training (habituation) is carried out using the GRADSIM package for training recurrent neural networks with variable delay links (Watrous, 1993). In a typical experiment, habituation took only 34 epochs for the ABB pattern, and only 30 epochs for the ABA pattern (training was terminated when the sum-squared-error between the input and the target fell below 0.001). Each epoch consisted of a single presentation of the 16 habituation sentences listed in (Elman, 1999).

In the test phase, the activity of clamp nodes is set as before to represent a test sentence. The network now computes the activity of positional role nodes. This activity indicates the expectation of the habituated network as to which item should occur in which position. The mismatch between this expectation and the actual input is measured by computing the mean-squared-difference of the network-induced activity of positional role nodes and the purely input-induced activity of these nodes.

Simulation Results

Simulation results are presented in Table 4. The mean-squared-difference between the network's expectation and the actual input is displayed along with standard deviation of this error. The entries along the main diagonal correspond to the situation where the habituation and test patterns are the same, and off-diagonal entries correspond to situations where the two patterns are different. The relatively small errors along the main diagonal indicate that networks exhibit a low degree of surprise when the test and habituation sentences conformed to the same pattern, and a high degree of surprise when they do not. Thus the network

Test Pattern	Habituation Pattern	
	ABB	ABA
ABB	*0.44 (0.18)*	0.77 (0.08)
ABA	0.72 (0.08)	*0.21 (0.05)*

Table 4: Mean-squared-difference between the network's expectation and purely input-induced activity: Mean (Standard Deviation)

Habituation	Cor(P1,P2)	Cor(P1,P3)	Cor(P2,P3)
ABB	0.1667	0.1682	**0.9998**
ABA	0.4319	**0.9993**	0.4330

Table 5: Correlation among link weights incident on different positional role nodes. Row 1 corresponds to the network habituated to the ABB pattern, and Row 2 to the network habituated to the ABA pattern.

response is analogous to that of infants in the Marcus et al. experiments. It follows, that the networks are able to distinguish between the two patterns and generalize this distinction to new sentences.

In order to better understand what the network is learning, we also computed the correlation of link weights for the links from second hidden layer to the positional role nodes. Table 5 shows these correlation coefficients. It is clear from the table that the network habituated to the ABB pattern has learned that the activity of P2 and P3 should be correlated. Similarly, the network habituated to the ABA pattern has leaned that the activity of P1 and P3 should be correlated.

The above confirms that the habituated networks learn to encode the requisite algebraic rules in their link weights. Consider the network habituated with the pattern ABA. This network must learn that "the item in position 1 is the same as item in position 3". The extremely high correlation (0.9993) between the weights of the links from the hidden structure into P1 and P3, respectively, indicates that these links are highly similar. This means that the habituated network induces the same activity in P1 and P3, thereby synchronizing their firing. Since bindings are expressed via synchronous activity, the habituated network is essentially encoding that P1 and P3 should be bound to the *same* entity. It follows that the network has correctly learned that the items in the first and the third position of a sequence should be the same. Similarly, the extremely high correlation (0.9998) in the ABB network between the weights of links from the hidden structure into P2 and P3, respectively, indicates that the ABB network has correctly learned that the items in the second and the third position of the sequence should be the same.

Analogous results for the AAB versus BAB patterns are shown in Tables 6 and 7. This experiment rules out the possibility that our network is able to distinguish between two patterns by simply determining whether the final two syllables are the same or different.[4]

[4]This additional control was suggested by Eimas (1999) and was performed on infants by Marcus and his colleagues. The results were consistent with previous findings reported in (Marcus, 1999b).

Test Pattern	Habituation Pattern	
	AAB	BAB
AAB	*0.41(0.19)*	0.74(0.17)
BAB	1.14(0.54)	*0.21(0.05)*

Table 6: Mean-squared-difference between the network's expectation and purely input-induced activity: Mean (Standard Deviation)

Habituation	Cor(P1,P2)	Cor(P1,P3)	Cor(P2,P3)
AAB	**0.9995**	0.4601	0.4482
BAB	0.4319	**0.9993**	0.4330

Table 7: Correlation among link weights incident on different positional role nodes. Row 1 corresponds to the network habituated to the AAB pattern, and Row 2 to the network habituated to the BAB pattern.

Algebraic Rules and Coincidence Detection

Note that the proposed network architecture — especially, the use of positional role nodes and temporal synchrony binding — reduces the problem of learning an algebraic rule to the problem of learning to detect *coincident* activity. For example, the network learns the ABA pattern by simply learning that nodes coding the first position should fire coincidently with nodes coding the third position.

Beyond Algebraic Rules

One may view much of the *systematic* knowledge underlying cognition and language as a collection of context-sensitive relational rules that have both an algebraic aspect and a *semantic* aspect (Shastri & Ajjanagadde, 1993; Shastri, 1997).[5] While the former governs the systematic relationship between the roles of related situations, the latter serves as a semantic-filter that modulates the strength of this algebraic relationship depending on the features and properties of the entities serving as role fillers in a given situation. Consider the following rule as an illustrative example:

> if x walks-into y and if x is animate and y is solid then x gets hurt.

The algebraic aspect of the above rule says that it is the *agent* (x) of the walk-into event that gets hurt in the resulting hurt event. The semantic aspect of the rule says that for the resulting hurt event to occur, the agent of walk-into must have the feature animate and the patient of walk-into must have the feature solid.

In general, such context-sensitive relational rules are "soft" and apply to a range of conditions with varying degree. Thus the rule about walking-into things and getting hurt may be viewed as:

$$walk\text{-}into(x,y) = (\alpha(x,y)) \Rightarrow hurt(x)$$

[5] In Shastri & Ajjanagadde (1993) these two aspects were labeled *systematicity* and *appropriateness*.

which conveys that "If one walks into something, one might get hurt. However, the degree of hurt in a given situation is determined by α which is a (potentially complex) function of the features of the actual role-fillers x and y in the given situation." For example, α may depend on, among other things, the hardness, the shape, and the relative size and weight of y.

The proposed architecture can be used to solve the more complex problem of extracting and applying context-sensitive relational rules. In previous work (Shastri, 1997) we have shown that an analogous network architecture can learn relational rules from examples, and apply these rules to novel items. Thus given the input "John walked into a wall" the above network could infer that John got hurt, but not so given the input "John walked into the mist".

Conclusion

We have described a connectionist network architecture that can readily acquire abstract algebraic rules. The network acquires such rules with ease and it does so without using negative evidence. The rules generalize to new words and are not tied to features and/or statistical properties of words. In fact, this network architecture can extract the requisite rules even if words are represented as orthogonal feature vectors. It is also noteworthy that the network does not require pretraining of the sort required by the model described in [5].

The model demonstrates that an appropriate architecture and representation can reduce the problem of learning an algebraic rule to the simpler task of learning to detect *coincidences* within a spatiotemporal pattern. This is perhaps, the most interesting aspect of the proposed architecture, for it suggests that the representation of certain types of algebraic rules can be grounded in basic notions of spatial and temporal locations and coincidence.

We believe that this work is a simple demonstration of the power of melding structured connectionist models [6] with the notion of temporal synchrony variable binding, and learning. Other examples may be found in [13, 14, 7].

Acknowledgments

This work was partially funded by ONR grants N00014-93-1-1149 and NSF grant SBR-9720398. Thanks to Gary Marcus, Jerry Feldman and other members of the NTL group for discussions.

References

[1] A.F. Carpenter, A.P. Georgopoulos, and G. Pellizzer. Motor cortical encoding of serial order in a context-recall task. *Science*, 283:1752–1757, 1999.

[2] C. deCharms and M. Merzenich. Primary cortical representation of sounds by the coordination of action-potential timing. *Nature*, 381:610–613, 1996.

[3] P. Eimas. Do infants learn grammar with algebra or statistics? *Science*, 284:436, 1999. Comments on Marcus et. al. 1999.

[4] J.L. Elman. Finding structure in time. *Cognitive Science*, 14:179–211, 1990.

[5] J.L. Elman. Generalization, rules and neural networks: A simulation of marcus et. al. http://crl.ucsd.edu/~elman/Papers/MVRV/simulation.html, 1999.

[6] J.A. Feldman. Neural representation of conceptual knowledge. In L. Nadel, L.A. Cooper, P. Culicover, and R.M. Harnish, editors, *Neural Connections, Mental Computation*. MIT Press, Cambridge, MA, 1989.

[7] J.E. Hummel and K.J. Holyoak. Distributed representations of structure: a theory of analogical access and mapping. *Psychological Review*, 104:427–466, 1997.

[8] G.F. Marcus. Do infants learn grammar with algebra or statistics? *Science*, 284:436–437, 1999. Response to comments.

[9] G.F. Marcus, S. Vijayan, S.B. Rao, and P.M. Vishton. Rule learning in seven month-old infants. *Science*, 283:77–80, 1999.

[10] M. Negishi. Do infants learn grammar with algebra or statistics? *Science*, 284:435, 1999. Comments on Marcus et. al. 1999.

[11] M. S. Seidenberg and J. L. Elman. Do infants learn grammar with algebra or statistics? *Science*, 284:434–435, 1999. Comments on Marcus et. al. 1999.

[12] L. Shastri. Exploiting temporal binding to learn relational rules within a connectionist network. Technical Report TR-97-003, International Computer Science Institute, Berkeley, CA, USA, 1997.

[13] L. Shastri. A model of rapid memory formation in the hippocampal system. In *Proceedings of the Nineteenth Conference of the Cognitive Science Society*, pages 680–685, 1997.

[14] L. Shastri. Advances in SHRUTI — a neurally motivated model of relational knowledge representation and rapid inference using temporal synchrony. *Applied Intelligence*, In Press, 1999.

[15] L. Shastri and V. Ajjanagadde. From simple association to systematic reasoning a connectionist representation of rules, variables and dynamic bindings using temporal synchrony. *Brain and Behavioral Sciences*, 16(3):417–494, 1993.

[16] W. Singer and C.M. Gray. Visual feature integration and the temporal correlation hypothesis. *Annual Review of Neuroscience*, 18:556–586, 1995.

[17] M. Usher and N. Donnelly. Visual synchrony affects binding and segmentation in perception. *Nature*, 394:179–182, 1998.

[18] C. von der Malsburg. The correlation theory of brain function. Technical Report 81-2, Max-Planck Institute for Biophysical Chemistry, Gottingen, Germany, 1981.

[19] R. L. Watrous. *GRADSIM: a connectionist network simulator using gradient optimization techniques*. Siemens Corporate Research, Inc., Princeton, New Jersey., 1993.