Preference Moore Machines for Neural Fuzzy Integration

Stefan Wermter
The Informatics Center
School of Computing, Engineering & Technology
University of Sunderland
St. Peter's Way, Sunderland SR6 0DD, United Kingdom
Email: stefan.wermter@sunderland.ac.uk

Abstract

This paper describes multidimensional neural preference classes and preference Moore machines as a principle for integrating different neural and/or symbolic knowledge sources. We relate neural preferences to multidimensional fuzzy set representations. Furthermore, we introduce neural preference Moore machines and relate traditional symbolic transducers with simple recurrent networks by using neural preference Moore machines. Finally, we demonstrate how the concepts of preference classes and preference Moore machines can be used to integrate knowledge from different neural and/or symbolic machines. We argue that our new concepts for preference Moore machines contribute a new potential approach towards general principles of neural symbolic integration.

1 Introduction

In previous years there has been a fair amount of interest in hybrid and connectionist systems, that is, in systems which integrate symbolic, neural and/or statistical knowledge for solving difficult real-world tasks [Reilly and Sharkey, 1992; Miikkulainen, 1993; Yager, 1994; Medsker, 1995; Dorffner, 1997; Cleeremans and Destrebecqz, 1997]. Much work on hybrid systems has been guided by the particular tasks at hand [Dyer, 1991; Honavar and Uhr, 1994; Sun and Bookman, 1995] and only little work has concentrated towards more general rigorous models of neural interpretation (for an early exception see [Smolensky, 1988; Sharkey and Jackson, 1995]). This lack of fundamental principles of hybrid neural/symbolic integration was also identified at a recent international workshop on hybrid intelligent systems [Wermter and Sun, 1998] and this paper addresses this current issue.

In this paper we want to start with focusing on the integration of simple symbolic machines, fuzzy representations and recurrent neural networks. First, we introduce the general concept of multidimensional neural preferences. Then, we relate multidimensional neural preferences to multidimensional fuzzy set representations and show that the corner preference order on preference classes is a partial order. This allows us to rank different neural preferences and provides a basic link between neural preferences and symbolic fuzzy representations at the *preference class level*.

Then, we introduce neural preference Moore machines and relate traditional symbolic transducers with simple recurrent networks by using neural preference Moore machines. Preference Moore machines provide a link between simple recurrent networks and symbolic transducers at the *preference Moore machine level*.

Finally, we demonstrate how the concepts of preference classes and preference Moore machines can be used to integrate knowledge from different neural and/or symbolic machines. We introduce operations like intersection and union on preference classes and show that these operations on preference classes are commutative, associative, and monotonic. These operations provide a link between several neural or symbolic modules at the *system architecture level*.

2 Neural Preferences, their Order, and their Preference Value

Typically, artificial neural networks receive analogous input from a number of network units (input layer) and they produce output for a number of network units (output layer). While the actual processing within different networks may be very different, their external interface may be modeled by a general multidimensional preference for input and/or output.

Definition 1 (Preference) *A preference is an analog representation which is represented by an m-dimensional vector $p \in [0,1]^m$.*

Definition 2 (Preference Mapping)
A preference mapping is a mapping between preferences: $[0,1]^n \to [0,1]^m$, with n, m positive integers.

Such a preference mapping could be a transformation of the input or a prediction of the next input based on the current input. If we want to rank preferences according to their strength, we need to specify an order for m-dimensional preferences in $[0,1]^m$. Within this m-dimensional space, we will consider a preference as being large if the values of the individual vector elements are close to 1 or 0. In contrast, we will consider a preference as being small if the values of the individual vector elements are close to 0.5. This is our *goal criterion* for determining a partial order on preferences. In general, we define a preference a as being larger than another preference b, if a has a smaller distance to a reference.

Definition 3 (Reference Order) *Let $a = (a_1, \cdots, a_m)$, $b = (b_1, \cdots, b_m)$ be two m-dimensional preferences and a reference $r = (r_1, \cdots, r_m)$ from $[0,1]^m$. Then the reference order $\geq_r$ is defined as: $(a_1, \cdots, a_m) \geq_r (b_1, \cdots, b_m)$, if $|r-a| \leq |r-b|$. Here $\leq$ is the usual order on real numbers and $|r-a| = \sqrt{(r_1-a_1)^2 + \cdots + (r_m-a_m)^2}$ is the Euclidean distance of the preference a to the reference r.*

It is possible to determine multiple references, for instance, the corner references from $\{0,1\}^n$. These corner references are particularly interesting since they allow a direct symbolic interpretation of preferences for input and output.

Definition 4 (Corner Reference Order) *If r is a corner reference $r = (r_1, \cdots, r_m) \in \{0,1\}^m$, then the reference order $\geq_r$ is based on the distance of two preferences from this corner reference. We call this special form of the reference order the corner reference order.*

That is, referring to a corner reference r, a preference a is greater than or equal to a preference b, if the distance of a to r is smaller than or equal to the distance of b to r. The corner reference can be interpreted as a strict, sharp preference. Below, we will specify that $r(a)$ is the next corner reference with minimal distance to a currently considered preference a. We define in detail:

Definition 5 (Next Corner Reference) *The next corner reference $r(a) \in \{0,1\}^m$, which is closest to $a \in [0,1]^m$, is determined for $i \in \{1, \cdots, m\}$ as:*

$$r(a)_i = 0, if\ a_i < 0.5$$

$$r(a)_i = 1, if\ a_i \geq 0.5$$

We consider an example: (0.9 0.1) and (0.6 0.4) are comparable and (0.9 0.1) $\geq_r$ (0.6 0.4), because (0.9 0.1) is closer to the next corner reference (1 0) than (0.6 0.4) to (1 0). Furthermore, it holds that: (0.9 0.8) $\geq_r$ (0.5 0.6), because the distance of (0.9 0.8) to (1 1) is smaller than the distance of (0.5 0.6) to a corner reference. The closer a preference is to a corner reference, the greater the preference. This can be defined more formally by assigning a preference value from the interval $[0,1]$ to each preference a related to its next corner reference r in the m-dimensional space:

Definition 6 (Preference Value of a Preference) *Let $r(a)$ be the next corner reference for a preference a in m-dimensional space. Let $distance(a, r(a))$ be the Euclidean distance between a and $r(a)$. Then we define the preference value of a preference a with respect to $r(a)$ as:*

$$pref_{r(a)}(a) = 1 - \frac{distance(a, r(a))}{\frac{\sqrt{m}}{2}}$$

$\sqrt{m}/2$ is the maximum distance in m-dimensional space to the next corner reference, that is the distance from the center to the corner references. Therefore, the values of $pref_{r(a)}(a)$ are between 0 and 1. If a is close to its next corner reference $r(a)$, then $pref_{r(a)}(a)$ is close to 1. If a is close to the center reference $(0.5, \cdots, 0.5)$, then $pref_{r(a)}(a)$ is close to 0.
Figure 1 shows the preference values for the two-dimensional space. For each two-dimensional preference $(x\ y)$ the corresponding preference value z is shown. In general, the value $pref_{r(a)}(a)$ has been given as the preference value of a preference a referring to a reference $r(a)$. For instance, a preference value for a categorization would specify how strong a certain category assignment would be.

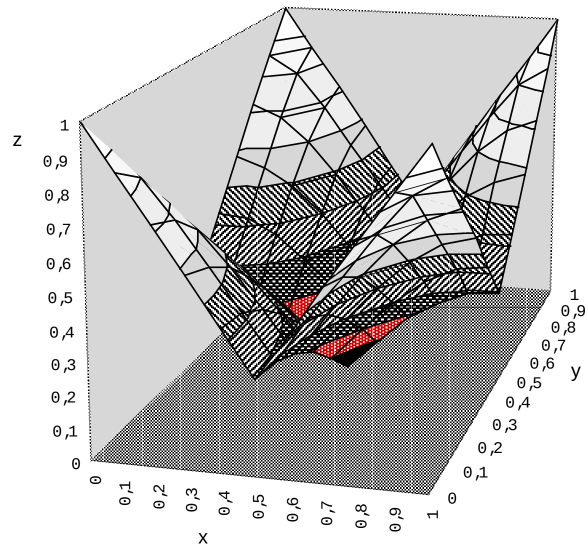

Figure 1: Preference values z of two-dimensional preferences (x y)

3 Neural Preferences as Multidimensional Fuzzy Set Representations

An m-dimensional preference can be seen as an m-dimensional vector of a neural network as well as an m-dimensional fuzzy set. For a *neural interpretation*, the preference value is a measure of how far away a neural preference is from a discrete symbolic corner vector, which represents the corner reference. For a *fuzzy interpretation*, the preference value is a measure of how far away a fuzzy set is from the corresponding symbolic sharp set which represents the corner reference.

For each preference in m-dimensional space, we can specify a preference value in $[0, 1]$. Because of the definition of the corner reference order and the definition of the preference value, only preferences with the same corner reference can be compared. This property is useful, since the preferences (0.9 0.3) and (0.3 0.9) for the different references (1 0) and (0 1) would provide the same preference value $pref_{(1\ 0)}(0.9\ 0.3)$ and $pref_{(0\ 1)}(0.3\ 0.9)$; however, it cannot be decided whether (0.9 0.3) or (0.3 0.9) are greater, since these preferences belong to different corner references. It is only possible to compare preferences which have the same corner reference. Those preferences which have the same distance to the same corner reference are judged as equal, for instance (0.9 0.8) and (0.8 0.9), because $pref_{(1\ 1)}(0.9\ 0.8) = pref_{(1\ 1)}(0.8\ 0.9)$. It is not possible to determine which of these preferences is greater and closer to the corner reference (1 1).

Our previous definition of the corner reference order is not yet a partial order. However, a partial order is a minimum requirement for the definition of all fuzzy sets with multi-dimensional goal domains [Klir and Folger, 1988]. The corner reference order is already transitive and reflexive, but it is not antisymmetric. For antisymmetry it must hold: if $x \geq_r y$ and $y \geq_r x$ then $x =_r y$. However, $(0.8\ 0.9) \geq_r (0.9\ 0.8)$ and $(0.9\ 0.8) \geq_r (0.8\ 0.9)$, but both preferences are different. Therefore, we cluster those preferences which belong to the same next corner reference into one class. We want to define the corner reference order based on these classes.

Definition 7 (Class of Preferences) *Let* $a = (a_1, \cdots, a_m)$ *be a preference and* $r(a) = (r_1, \cdots, r_m) \in \{0, 1\}^m$ *is next corner reference. Then the class of preferences of* a *is called* $c(a)$ *and contains all those preferences for next corner reference* $r(a)$*, which have the same distance from* $r(a)$ *as* a*.*

Definition 8 (Order on Preference Classes) *Let* $a = (a_1, \cdots, a_m)$, $b = (b_1, \cdots, b_m)$ *be two preferences and their common next corner reference* $r = (r_1, \cdots, r_m)$*. Then the corner reference order on classes of preferences* $\geq_{rc}$ *is defined as follows:* $c(a) \geq_{rc} c(b)$, *if* $|r - a| \leq |r - b|$*. Here* $\leq$ *is the usual order for real numbers and* $|r - a| = \sqrt{(r_1 - a_1)^2 + \cdots + (r_m - a_m)^2}$ *is the distance of the preference* a *from reference* r*. We say that preference class* $c(a)$ *is greater than or equal to the preference class* $c(b)$*.*

Definition 9 (Preference Value of a Class) *The preference value of a preference class* $c(a)$ *is the preference value of an arbitrary preference which belongs to this class.*

Theorem 1 *The corner reference order for preference classes is a partial ordering.*

Sketch of Proof:
Let $a = (a_1, \cdots, a_m)$, $b = (b_1, \cdots, b_m)$ be two preferences with their corresponding preference classes $c(a)$ and $c(b)$. Let $r = (r_1, \cdots, r_m) \in \{0, 1\}^m$ be their common next corner reference. Then it is straightforward to show reflexivity, antisymmetry and transitivity for the preference classes.

The corner reference order for classes of preferences is a partial order which meets the particular requirements for a neural interpretation of preferences (multidimensional and uncertain close to 0.5) but also the general requirements for a fuzzy interpretation of preferences (at least partial order in the goal domain) and also the general requirements of neural and symbolic integration (symbolic corner reference as a reference for classes of neural preferences). The preference value of a class of output preferences of a neural network can be understood as the membership degree of these output preferences for an m-dimensional fuzzy set which represents a reference (for instance a corner reference) in m-dimensional space. Figure 2 shows examples of four preference classes which have the same distance to their corresponding corner reference.

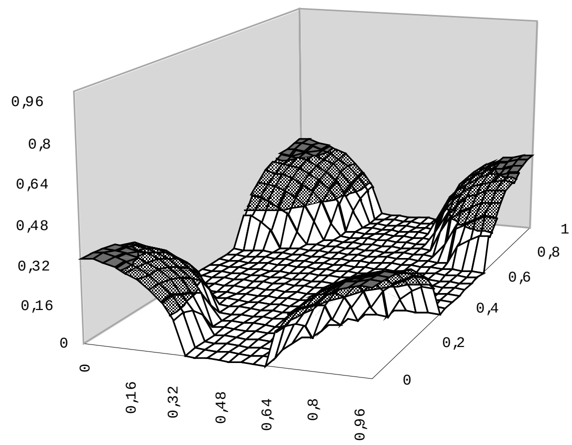

Figure 2: Classes of preferences in three-dimensional space

Another reason for the use of classes of preferences is based on symbolic processing. If a preference value for $(Feature1, Feature2)$ has to be specified, a single value, e.g. 0.8, can be given. This preference value corresponds to all those preferences which have the same corresponding distance from the specified corner reference. Therefore a class of preferences also supports the integration of symbolic and neural representations. A class of preferences represents a high-dimensional hypersphere of an unlimited number of preferences with the same distance from the specified corner reference.

4 Interpretation of Dynamic Preference Mappings

So far we have concentrated on static preferences and preference classes. Our next step is to focus on dynamic preference mappings which can be associated with certain sequential machines. As one possibility for relating principles of symbolic computational representations and neural representations by means of preferences, we consider a so-called neural preference Moore machine. We have chosen this type of machine since they are simple and widely applicable.

Definition 10 (Preference Moore Machine)
A preference Moore machine PM is a synchronous sequential machine, which is characterized by a 4-tuple $PM = (I, O, S, f_p)$, with I, O and S non-empty sets of inputs, outputs and states. $f_p : I \times S \to O \times S$ is the sequential preference mapping and contains the state transition function f_s and the output function f_o. Here I, O and S are n-, m- and l-dimensional preferences with values from $[0,1]^n$, $[0,1]^m$ and $[0,1]^l$, respectively.

A general version of a preference Moore machine is shown to the left of figure 3. The preference Moore machine realizes a sequential preference mapping, which uses the current state preference S and the input preference I to assign an output preference O and a new state preference.

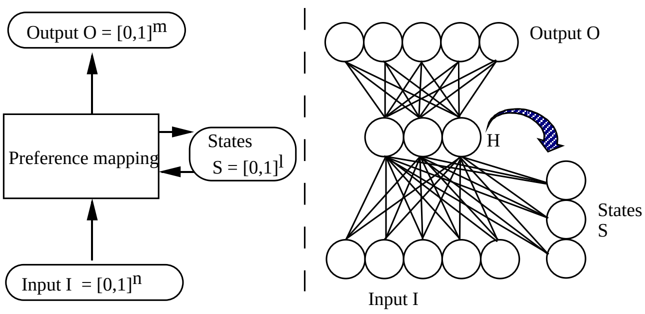

Figure 3: Neural preference Moore machine and its relationship to a simple recurrent neural network

Simple recurrent networks (also called SRN) [Elman, 1990] have the potential to learn a sequential preference mapping $f_p : I \times S \to O \times S$ automatically based on input and output examples (see figure 3), while traditional Moore machines or Fuzzy-Sequential-Functions [Santos, 1973] use manual encodings.
Such a simple recurrent neural network constitutes a neural preference Moore machine which generates a sequence of output preferences for a sequence of input preferences. Here, internal state preferences are used as local memory. A feedforward network represents a neural preference Moore machine with a degenerated sequential memory, since there is no possibility to have an influence from previous patterns.
On the one hand, we can associate a neural preference Moore machine in a preference space with its symbolic interpretation. On the other hand, we can represent a symbolic transducer in a neural representation. Using the symbolic m-dimensional preferences and the corner reference order, it is possible to interpret neural preferences symbolically and to integrate symbolic preferences with neural preferences.
Each preference of a neural trajectory is a representative of its preference class and it is possible to assign a symbolic description as a corner reference together with a preference value. In this way, neural preferences can be interpreted symbolically. On the other hand, symbolic knowledge can be integrated with neural knowledge by associating a preference value with a symbolic corner reference. This preference value of the symbolic reference determines which neural preference class is associated with the symbolic reference.

5 Combination of Symbolic/Neural Preferences

An integration of symbolic Moore machines and neural preference Moore machines has a number of advantages. Known knowledge can be represented as manually coded symbolic Moore machines. Unknown knowledge can be learned in neural preference Moore machines. Symbolic regular relations can be understood as a top-down specification for symbolic Moore machines. Alternatively, a training set can be viewed as a bottom-up specification for neural preference Moore machines.

5.1 A Connection via Preference Classes

For one training set there can be several different neural preference Moore machines which realize the training set and which differ in their connections and weights. Symbolic Moore machines represent knowledge at a higher discrete abstraction level compared to neural preference Moore machines. Therefore we want to examine a possible integration of symbolic and neural Moore machines. We will consider a single neural or symbolic Moore machine as a unit, whose input and output should be integrated. We suggest that a preference class could be a suitable connection between different symbolic and/or neural Moore machines (see also section 3). Below we will focus on operations on preference classes.

5.2 Operations on Preference Classes

Notation for Preference Classes
Let $[0,1]^n \to [0,1]^m$ be a mapping which associates input preferences with output preferences. For preference classes from $[0,1]^m$ we can define the operations for intersection and union.
Let $a = (a_1, \cdots, a_m)$, $b = (b_1, \cdots, b_m)$ be two preferences from $[0,1]^m$, with their corresponding preference classes $class_{r(a)}(a)$ and $class_{r(b)}(b)$. Let $pref_{r(a)}(a)$ be preference value of a preference a for a reference $r(a)$; similarly this holds for $pref_{r(b)}(b)$. If it is clear that the reference of a preference is the next corner reference $r(a) \in \{0,1\}^m$, we say $p(a)$ rather than $pref_{r(a)}(a)$ and $c(a)$ rather than $class_{r(a)}(a)$.
For simplicity, we will consider a preference class in a slightly modified compact notation as a pair of reference and preference value: $preference\ class = (reference, preference\ value)$. For instance, $((0\ 1), 0.3)$

is a preference class in the two-dimensional space which contains all preferences which have the preference value 0.3 for the reference (0 1).

Definition 11 (Union of Preference Classes) *The union of two preference classes $(r(a), p(a))$ and $(r(b), p(b))$ is a preference class with the reference $(X(r(a)_1, r(b)_1), \cdots X(r(a)_m, r(b)_m))$ and preference value $X(p(a), p(b))$. X is the maXimum function.*

$$PU(((r(a)_1, \cdots, r(a)_m), p(a)), ((r(b)_1, \cdots r(b)_m), p(b)))$$

$$= ((X(r(a)_1, r(b)_1) \cdots X(r(a)_m, r(b)_m)), X(p(a), p(b)))$$

In general, this definition can be described as follows: if the (symbolically interpretable) reference of two preference classes is equal, then the reference will be kept and the union provides the preference class with the larger preference value. If the reference of two preference classes is different then the union is extended to the references. Thus, the union provides the preference class with the larger preference value.

Definition 12 (Intersection of Preference Classes) *The intersection of two preference classes $(r(a), p(a))$ and $(r(b), p(b))$ is a preference class which has the reference $(N(r(a)_1, r(b)_1), \cdots N(r(a)_m, r(b)_m))$ and the preference value $N(p(a), p(b))$. N is the miNimum function.*

$$PI(((r(a)_1, \cdots r(a)_m), p(a)), ((r(b)_1, \cdots r(b)_m), p(b)))$$

$$= ((N(r(a)_1, r(b)_1) \cdots N(r(a)_m, r(b)_m)), N(p(a), p(b)))$$

That is, if the (symbolically interpretable) reference of two preference classes is equal, then the reference will be kept and the intersection provides the preference class with the smaller preference value. If the reference of two preference classes is different, then the intersection is extended to the references. Thus, the intersection provides the intersected preference class with the smaller preference value.

5.3 Relationship of m-dimensional Preference Classes to Fuzzy Sets

We will now examine whether our operations for preference classes fulfill axioms which are a basic precondition for a relationship of preference classes to fuzzy sets. The axioms are: 1) generalization of sharp sets, 2) commutativity, 3) monotonicity, and 4) associativity. The following theorems can be proven:

Theorem 2 *Let $\geq_{rc}$ be the partial ordering for the m-dimensional preference space $[0,1]^m$. Then the union PU on preference classes fulfills the axioms generalization of sharp sets, commutativity, monotonicity, and associativity.*

Theorem 3 *Let $\geq_{rc}$ be the partial ordering for the m-dimensional preference space $[0,1]^m$. Then the intersection PI on preference classes fulfills the axioms generalization of sharp sets, commutativity, monotonicity, and associativity.*

Now we will illustrate the use of the union and intersection for preference classes. We consider the 2-dimensional space. In the following illustration, we refer to a small preference value with "S" and to a large with "L". Then $((00), S)$ is the preference class which contains those preferences which have a preference value S with respect to the reference (00).

If the preference classes have the same corner reference, they are directly comparable with their preference values. The preference class $PU((r(a), p(a)), (r(b), p(b)))$ is the preference class with the largest preference, that is $PU(((00), S), ((00), L)) = ((00), L)$. On the other hand, $PI((r(a), p(a)), (r(b), p(b)))$ is the preference class with the smallest preference, that is $PI(((00), S), ((00), L)) = ((00), S)$.

If the preferences classes have a different corner reference, the preference classes cannot be judged only by their preference value. In this case, the preference classes $PU((r(a), p(a)), (r(b), p(b)))$ and $PI((r(a), p(a)), (r(b), p(b)))$ are a generalization of the standard union and intersection. Therefore, it holds that for instance $PU(((00), S), ((01), S)) = ((01), S)$ but $PI(((00), S), ((01), S)) = ((00), S)$. This is based on the following motivation: For instance, if there is a preference for (*no noun*, *no verb*) and at the same time a preference for (*no noun*, *verb*), then PU provides the optimistic integration, namely (*no noun*, *verb*) and PI provides the pessimistic integration, namely (*no noun*, *no verb*). The preference value of the intersection of preference classes is the minimum of the preference values of the arguments, and the preference value of the union of preference classes is the maximum of the preference values of the arguments.

We have illustrated that the union and intersection on preference classes and fuzzy sets fulfill equivalent axioms. Furthermore, fuzzy sets and preference classes represent uncertainty by a fuzzy value and a preference value, respectively. Therefore there is a tight relationship between fuzzy sets and preference classes if we interpret them as points in m-dimensional space.

6 Discussion and Conclusion

We have introduced multidimensional neural preference classes and preference Moore machines as one general principle for integrating different neural and/or symbolic knowledge sources. For ranking different preferences, we introduced a new reference order and showed that the corner preference order on preference classes is a partial order. This allowed us to rank different neural preferences and provides preference classes as a basic link between neural preferences and symbolic fuzzy representations at the *preference level*. We introduced neural preference Moore machines which provide a link between simple recurrent networks and symbolic transducers at the *preference Moore machine level*. Finally, we demonstrated how the concepts of preference classes and preference Moore machines can be used to integrate knowledge from different neural and/or symbolic machines.

We used operations like intersection and union on preference classes and proved that these operations on preference classes are commutative, associative and monotonic. These operations provide a link between several neural or symbolic modules at the *system architecture level.*

Recently, the question as to whether simple recurrent networks can emulate each symbolic Moore machine and each finite automaton has been examined [Kremer, 1996]. On the other hand it has been shown [Goudreau and Giles, 1995; Goudreau *et al.*, 1994] that a recurrent network with only one input layer, one context layer and one output layer (so-called Single-Layer-First-Order-Network) is not sufficient for realizing arbitrary finite automata. Therefore, the link with our preferences between simple recurrent networks and symbolic Moore machines is particularly important.

Here we focused on Moore machines because they are relatively simple and efficient. So far, it could be shown that simple recurrent networks can emulate certain restricted properties of a pushdown automaton, in particular the recursive representation of structures up to a limited depth [Elman, 1990; Wiles and Elman, 1996]. In the future, more complex machines, like different pushdown automata with explicit unlimited memory, may be further candidates for additional principles of neural symbolic integration. Neural preference Moore machines like simple recurrent networks can support such properties but can also benefit from the integration with known symbolic heuristics.

References

[Cleeremans and Destrebecqz, 1997] A. Cleeremans and A. Destrebecqz. Incremental sequence learning. In *Proceedings of the 19th Conference of the Cognitive Science Society*, pages 119–124, Stanford, 1997.

[Dorffner, 1997] G. Dorffner. *Neural Networks and a New AI.* Chapman and Hall, London, UK, 1997.

[Dyer, 1991] M. G. Dyer. Symbolic neuroengineering for natural language processing: a multilevel research approach. In J. A. Barnden and J. B. Pollack, editors, *Advances in Connectionist and Neural Computation Theory, Vol.1: High Level Connectionist Models*, pages 32–86. Ablex Publishing Corporation, Norwood, NJ, 1991.

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

[Goudreau and Giles, 1995] M. W. Goudreau and C. L. Giles. On recurrent neural networks and representing finite-state recognizers. In *Proceedings of the Third International Conference on Neural Networks*, pages 51–55, 1995.

[Goudreau *et al.*, 1994] M. W. Goudreau, C. L. Giles, S. T. Chakradhar, and D. Chen. First-order vs. second-order single layer recurrent neural networks. *IEEE Transactions on Neural Networks*, 5(3):511–513, 1994.

[Honavar and Uhr, 1994] V. Honavar and L. Uhr. *Artificial Intelligence and Neural Networks: Steps Toward Principled Integration.* Academic Press, Cambridge, MA, 1994.

[Klir and Folger, 1988] G. J. Klir and T. A. Folger. *Fuzzy Sets, Uncertainty and Information.* Prentice Hall, 1988.

[Kremer, 1996] S. C. Kremer. A theory of grammatical induction in the connectionist paradigm. Technical Report PhD dissertation, Dept. of Computing Science, University of Alberta, Edmonton, 1996.

[Medsker, 1995] L. R. Medsker. *Hybrid Intelligent Systems.* Kluwer Academic Publishers, Boston, 1995.

[Miikkulainen, 1993] R. Miikkulainen. *Subsymbolic Natural Language Processing.* MIT Press, Cambridge, MA, 1993.

[Reilly and Sharkey, 1992] R. G. Reilly and N. E. Sharkey. *Connectionist Approaches to Natural Language Processing.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1992.

[Santos, 1973] E. S. Santos. Fuzzy sequential functions. *Journal of Cybernetics*, 3(3):15–31, 1973.

[Sharkey and Jackson, 1995] N. E. Sharkey and S. A. Jackson. An internal report for connectionists. In R. Sun and L. A. Bookman, editors, *Computational Architectures integrating Neural and Symbolic Processes*, pages 223–244. Kluwer, Boston, 1995.

[Smolensky, 1988] P. Smolensky. On the proper treatment of connectionism. *Behavioral and Brain Sciences*, 11:1–74, 1988.

[Sun and Bookman, 1995] R. Sun and L.A. Bookman. *Computational Architectures Integrating Neural and Symbolic Processes.* Kluwer Academic Publishers, Boston, MA, 1995.

[Wermter and Sun, 1998] S. Wermter and R. Sun. *Nips Workshop on Hybrid Neural Symbolic Integration.* Nips, Breckenridge, CO, 1998.

[Wiles and Elman, 1996] J. Wiles and J. Elman. Learning to count without a counter: A case study of dynamics and activation landscapes in recurrent networks. In *Proceedings of the AAAI Workshop on Computational Cognitive Modeling: Source of the Power*, Portland, Oregon, 1996.

[Yager, 1994] R. R. Yager. Modeling and formulating fuzzy knowledge bases using neural networks. *Neural Networks*, 7(8):1273–1283, 1994.