Hybrid Sequential Machines based on Neuroscience

Stefan Wermter, Christo Panchev
University of Sunderland
Informatics Centre, SCET
St. Peter's Way
Sunderland SR6 0DD, United Kingdom
www.his.sunderland.ac.uk

Abstract

In the past a variety of computational problems have been tackled with different neural network approaches. However, very little research has been done on a framework which connects neuroscience-oriented models with connectionist models and higher level symbolic processing. In this paper, we outline a framework which focuses on a hybrid integration of various neural and symbolic preference techniques in order to shed more light on how we may process higher level concepts, for instance for language processing based on concepts from neuroscience. It is a first hybrid framework which allows a link between various levels from neuroscience, connectionist Preference Moore machines and symbolic machines.

1 Introduction

Our existing computational methods lack the flexibility and reliability of cognitive information processing in the brain. Although a great deal is known about the construction of the brain, this knowledge has had little impact on main stream computing. Since 1999, the computational neuroscience network EmerNet has explored emerging computational neural network architectures based on neuroscience [Wermter et al., 1999] (http://www.his.sunderland.ac.uk/emernet). This paper is based on this context and attempts to explore hybrid neural architectures based on neuroscience, in particular for language processing.

Recently, there has been some preliminary work integrating principles from neuroscience into computational models e.g. [Maass and Bishop, 1999, Thorpe et al., 1996, Wermter et al., 1999, Taylor, 1999, Denham and Denham, 1999]. Although neuroscience principles have helped to develop new computational models, the problems they address are still restricted, and they indicate that new evidence from cognitive neuroscience may help build more realistic brain-inspired computational frameworks.

In many ways there is a challenging distance between lower cognitive neuroscience and higher concepts like in language processing. However, long-term progress needs cognitive science and neuroscience to be taken more seriously by computer scientists for high-level processes like language understanding. Our approach attempts to go beyond the neural approaches that are normally utilized. Our approach is based on the processing found in the brain, integrates sequential machines at diverse levels both vertically and horizontally, and exploits recurrent and pulse neural networks for more neuron-like processing. It is envisaged that tasks like

auditory processing, robust syntactic analysis or semantic classification could benefit especially from such processing.
The focus of this new approach here is to explore hybrid neural architectures, including techniques from cognitive neuroscience and neural computation in order to produce realistic computational neural models of language processes and complex cognitive operations. These models require the integration and consideration of significant amounts of knowledge on brain structure and information processing that has been collected by neuroscientists and cognitive scientists. Furthermore, it is important to determine which principles are critical for higher level functions like language processing. These models, while being able to perform complex language processing operations, can also create general notions on language and the brain, and identify the information requirements for extended models. However, we would like to note that our presented framework is not about interpretation of biological neural networks or neurobiological modeling. Our goal is rather to extend the scope of hybrid approaches to neuroscience-inspired computational models of artificial neural networks.

2 Preference Moore Machines

First, we want to describe a synchronous sequential preference Moore machine which transforms sequential input preferences to sequential output preferences. Later, we will show (1) how symbolic and neural knowledge can be integrated quite naturally using preference Moore machines and (2) how preference Moore machines can be linked to more realistic neural modeling based on neuroscience evidence.

A substantial part of the information being processed in artificial and biological neural networks is encoded in a distributed manner and is transferred, or sometimes temporally stored, as pulsed signals between neurons. Within a given time window, neurons fire, indicating activity with the density or with the particular temporal location of the spikes. Reading such information from real systems or manipulating it in artificial systems is a complex task that addresses many processing and representational problems. In previous work we have introduced preference-based processing [Wermter, 1999, Wermter, pear] and an interpretation of firing rate and pulse coding schemes [Panchev and Wermter, 2000]. Here we would like to extend this work substantially towards an interpretation of some more complex neural network representations of cognitive events. While usually the processing and representation in the brain are believed to be task-dependent, a common neural/symbolic interpretation of spatio-temporal neural code is possible and crucial for hybrid systems.

Definition 1 (Complex Preference, briefly C-Preference) *A complex preference of level l is represented by an $l \times m$-dimensional matrix $a \in [0,1]^{l \times m}$.*

The special case of a c-preference of level one is called simple neural preference, or just preference. The intuition for a simple preference is a concept consisting of features which are present to various degrees. In [Panchev and Wermter, 2000], we showed that some single coding schemes can be interpreted as c-preferences where the simple preference at each level represents a given internal state of the code. Furthermore, we showed that multiple coding concepts can be integrated and can be simultaneously processed in a c-preference where each level (or several levels) represent a single coding scheme. In the following sections, we will use this previous work and the definitions to interpret some more complex neural representations as c-preferences of m-dimensional analog vectors in $[0,1]^m$ or preference Moore machines using c-preferences.

Definition 2 (Next Corner Reference) *The next corner reference $r(a) \in \{0,1\}^{l \times m}$ of the c-preference $a \in [0,1]^{l \times m}$ is determined for $i \in \{1,\ldots,l\}$ and $j \in \{1,\ldots,m\}$ as:*

$$r_{ij}(a) = \begin{cases} 0 & \text{if } a_{ij} < 0.5 \\ 1 & \text{if } a_{ij} \geq 0.5 \end{cases}$$

The introduction of the next corner reference allows us to associate each c-preference with a particular corner of the $[0,1]^{l \times m}$ hypercube, i.e. a discrete symbolic representation.

Definition 3 (Preference Value of a C-Preference) *A preference value of a c-preference $a \in [0,1]^{l \times m}$ with respect to its next corner reference $r(a)$ is defined as:*

$$pref(a) = 1 - \frac{distance(a, r(a))}{\frac{\sqrt{lm}}{2}} \quad \text{, where } \; distance(a, r(a)) = \sqrt{\sum_{i,j}(a_{ij} - r_{ij}(a))^2}$$

is the distance between the c-preference a and its next corner reference.

$\sqrt{lm}/2$ is the maximum distance in the $l \times m$-dimensional c-preference space, that is the distance from the center of the hypercube to any corner. If the c-preference a is close to its next corner reference then its preference value $pref(a)$ will be close to 1 and if it is close to the center, then $pref(a)$ will be close to 0.

Definition 4 (C-Preference Class) *Let $a \in [0,1]^{l \times m}$ be a c-preference with next corner reference $r(a) \in \{0,1\}^{l \times m}$. Then the class of complex preferences of a is called c-preference class $c(a)$ and contains all those c-preferences with next corner reference $r(a)$, which have the same distance from $r(a)$ as a.*

The *preference value of a class of c-preferences* is the preference value of an arbitrary c-preference which belongs to this class. This follows directly from the definitions of c-preference classes and the preference value. Figure 1 shows the preference values for the two-dimensional space.
A class of preferences represents a high-dimensional hypersphere of an unlimited number of preferences with the same distance from the specified corner reference. Figure 2 shows examples of four preference classes which have the same distance to their corresponding corner reference [Wermter, 1999].
Finally we would like to define a preference Moore machine as a device of sequential processing with c-preferences. For some input and state, a new state and output is computed. Input, output and state are multidimensional preferences.

Definition 5 (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 being non-empty sets of inputs, outputs and states. $f_p : I \times S \rightarrow O \times S$ 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.*

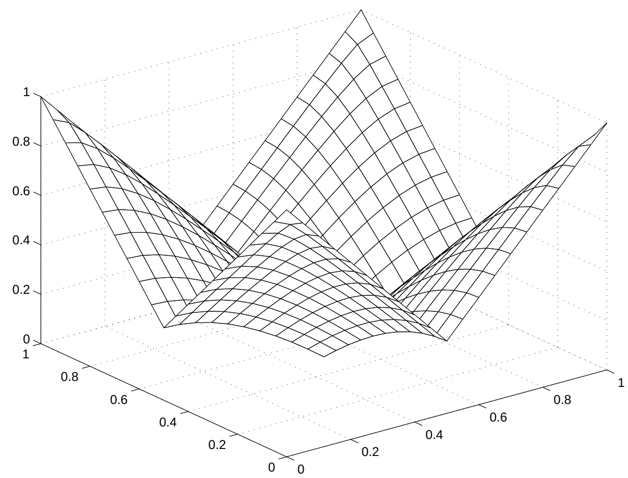

Figure 1: Preference values z of two-dimensional preferences (x y)

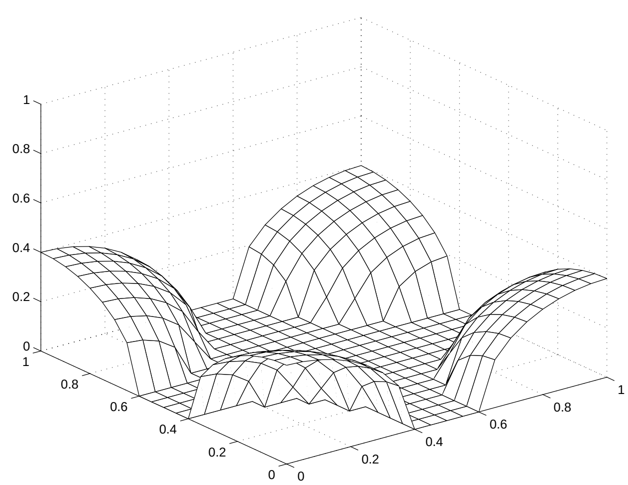

Figure 2: Classes of preferences in three-dimensional space

3 Using Preferences at Symbolic, Connectionist, and Neuroscience Levels

A Preference Moore machine can be seen as a computational machine which has possible links to higher level symbolic machines or lower level neuroscience-inspired concepts. For instance, a SRN network, a form of sequential connectionist network [Elman et al., 1996], is one type of a preference Moore machine and can be interpreted symbolically as finite state machines. It has been shown [Wermter, 2000], that symbolic transducers can be extracted from SRNs using our preference framework. Each state and each output within this preference Moore machine was mapped towards the references of an n-dimensional space. That way, a symbolic transducer represented a higher, more abstract representation of the more detailed connectionist preference Moore machine.

In the remaining part of this paper we will also illustrate some links of the preference framework to the neuroscience level. While in [Panchev and Wermter, 2000] we presented the concept of neural preferences on a single neuron level, here we would like to concentrate on more complex cortical functional structures associated with cognitive functions in the brain: cell assemblies and synfire chains.

3.1 Cell Assemblies as C-Preferences

The concept of a cell assembly was introduced as a functional and structural model for cortical processes and neuronal representations of external events [Hebb, 1949]. Hebb presented the idea that complex objects and stimuli, as well as more abstract entities like concepts, ideas and contextual relations in the brain are represented as simultaneous activation of large groups of neurons. Single cells can belong to different assemblies and the cells in one assembly are not necessarily close to each other. If, as a result of an external event, a sufficiently large subset of the cells in the assembly are stimulated, the whole assembly becomes active and may sustain activity for some period of time even when the external event has disappeared.

Cell assemblies are a widely accepted paradigm for feature binding mechanisms in the brain. In many artificial neural networks, cell assemblies are explored as a model of associative memories [Palm, 1986, Fransén et al., 1994]. Different interpretations of the paradigm can serve as a concept of short or long term memory models. The concept of neural assemblies in combination with activity-dependent (spatio-temporal) Hebbian learning provides a paradigm for long term memory [Wennekers and Palm, 1999].

Many artificial neural network models of cell assemblies use a simple neuron as the elementary computational unit of the network. However, there are several models of associative memories with spiking neurons that use models of cortical columns as functional units [Fransén and Lansner, 1998]. Although in both approaches a neuron or column represents a single feature, there are different interpretations of the behavior of that unit.

3.1.1 Cortical column as a threshold gate

In the first interpretation, a column is considered to behave as a threshold gate, that is, if a sufficient number of excitatory neurons fire, the column is said to be active and the respective feature present. If there are not enough firing neurons, the column is said to be non-activated and so is the feature it represents.

The neural preferences interpretation of a model with cortical columns acting as threshold gates is analogous to the case of single neurons and synchrony code. Let us consider a model of synchronously firing cell assemblies, with Δt being a time interval in which all spikes would be considered as synchronous. A sequence of synchronously firing assemblies will be defined in a sequence of intervals $\Delta t_1, \Delta t_2, \Delta t_3, \ldots$, where the s^{th} interval is defined as $\Delta t_s = \{\, t \mid t'_s < t < t''_s \,\}$, t'_s and t''_s are the beginning and the end of the interval, and $|\Delta t_s| = t''_s - t'_s$ is the length of the interval. In some implementations of spiking neurons, the sequence of intervals might represent a continuous time set, i.e. $t''_s = t'_{s+1}$, while in others there might be an explicit time shift between the separate intervals of synchronous firing, i.e. $t''_s < t'_{s+1}$. For each interval we can define $\bar{t}_s$ as the mean time of the spikes in Δt_s. Examining the spikes from time $|\Delta t_s|$ before t_s and $|\Delta t_s|$ after t_s, that is in interval $2|\Delta t_s|$ around t_s, we can define a spike time preference of a neuron (threshold gate column) in the interval Δt_s as:

$$a_s^i = \begin{cases} 1 - \frac{|t_s^i - \bar{t}_s|}{\Delta t_s} & \text{if neuron (column) } i \text{ has fired in the time window } 2\Delta t_s \\ 0 & \text{if neuron (column) } i \text{ has not fired in the time window } 2\Delta t_s \end{cases}$$

Here, t_s^i denotes the firing time of neuron (column) i. Then the vector $a_s = (a_s^1, a_s^2, \ldots, a_s^N)$ is the *c-preference vector of cell assemblies of single neurons or threshold gate columns* in the time interval Δt_s. According to the above definition of a_s, a higher density in the synchronous firing in the assembly will lead to values in the preference vector close to 1. Alternatively, lower density of the spikes inside the time window will lead to values close to 0.5. Finally, firing times outside the time window will lead to values close to 0 and therefore rejection of the represented features.

A *neural preference class of cell assemblies of single neurons or threshold gate columns* can be interpreted as a set of all preferences that represent the cell assembly for the same information with equal strength. This interpretation of the classes allows us to abstract from the particular distribution of the synchronous spikes in the time window usually considered as noise in biological systems.

3.1.2 Cortical column as a population of neurons

A second and more interesting interpretation of the behavior of a single column is when a column is considered as a population of neurons representing one particular feature and the level of activation of that feature is determined by the relative number of excitatory neurons that have fired at a particular time, i.e. examining the population code of a single column. Such a concept is a computationally efficient approach of encoding features with analog values. It allows the combination of two different encoding schemes within a single network: graded activation of features as a population code of a single column and binding of features via synchrony firing of cell assemblies.

Let us now consider such a column i with P^i excitatory and Q^i inhibitory neurons. For a particular time interval Δt_s of synchronous firing, the number of excitatory neurons in column i that have fired is denoted as p_s^i, and respectively, the number of inhibitory neurons would be q_s^i. We can define a value representing the activity of the column as:

$$a_s^i \;=\; \frac{1}{2}\left(1 + \frac{p_s^i}{P^i} - \frac{q_s^i}{Q^i}\right)$$

If most of the excitatory neurons in the column have fired and there is no activity of the inhibitory neurons, the activation value will be close to 1 and therefore indicate a strong preference for the feature that the column represents. The opposite situation will have a value close to 0 and would indicate strong suppression of the feature in the time interval. Finally, an activation value close to 0.5 would indicate low activity in the column and therefore no activation or suppression of the represented feature.

The vector constructed from the above defined activation values for all columns in a network with N columns $a_s = (a_s^1, a_s^2, \ldots, a_s^N)$ is the *neural c-preference of cell assemblies of cortical columns using population code.* A particular c-preference would represent the state of the network at a particular time and therefore contain a representation of the complex object (event) activated in the network at that time.

The *class of c-preferences of cell assemblies of cortical columns using population code* will allow us to abstract from the mutual fluctuations in the activity of the features included in a particular object. Such a class will include all c-preferences that represent the same object (event) with equal total activity of the assembly. Furthermore, the corner preference of the class will represent the object (event) as a binary vector and classes with the same corner preference will represent the same entity but with different strength.

3.2 Dynamic representations: Synfire Chains and C-Preferences

After having introduced cell assemblies, we now look into more dynamic representations, i.e. synfire chains. The concept of synfire chains as a model of cortical function was introduced by Abeles [Abeles, 1982, Abeles, 1991]. It explains some phenomena of precise timings in spatio-temporal patterns in frontal areas of the brain. A synfire chain consist of precisely timed repeating sequence of synchronously firing small pools of neurons. The firing time in a chain can spread over a large time period - usually a few hundred milliseconds, or up to one second. The neural pools are linked together in a feedforward chain, so that a wave of activity propagates from pool to pool in the chain. It has been shown that a computationally efficient number of spatio-temporal patterns can be stored in a network constructed of synfire chains where one neuron or cortical column can participate in several chains or several times in the same synfire chain [Herrman et al., 1995].

It is suggested that the activity waves in the synfire chains represent an elementary cognitive event [Bienenstock, 1995]. Synfire chains can be applied as storage elements of an associative memory: long term store for learning, recognition and recall of spatio-temporal patterns and as a possible physical substrate for short term memory [Wennekers and Palm, 1999]. Furthermore, it has been shown that synfire chain are able to regenerate ordered sequences of patterns [Aertsen et al., 1996, Abeles et al., 1993, Bienenstock, 1995].

Synfire chains can be viewed as a possible extension of the associative memories from static spatial patterns to dynamic spatio-temporal ones. Furthermore, there are several properties of the synfire chains that result from their dynamic behavior. For example the firing patterns exhibit cyclic activity, the order of firing of the pools in a chain is believed to be of significance, different chains can share the same pool at different times without crosstalk, etc. We suggest that a synfire chain is best interpreted as a dynamic symbolic representation and present the concept of preference Moore machine as one possible solution.

3.2.1 C-Preferences in the Synfire Chains

In an artificial neural network model, a synfire chain would represent a composite cognitive event. The event consists of several entities which might have explicitly defined semantics. Each pool in the chain would represent a single composite concept (entity). Therefore we can represent the activation of the network of N columns at a given interval Δt_s as a c-preference $a_s = (a_s^1, a_s^2, \ldots, a_s^N)$, where each value in the preference vector equals the population activity of a given column in the network. Such an interpretation is analogous to c-preferences of cell assemblies of a cortical column with population code and the formulas defined above are valid here:

$$a_s^i = \frac{1}{2}\left(1 + \frac{p_s^i}{P^i} - \frac{q_s^i}{Q^i}\right).$$

If the network has well-defined, say N, pools of columns, the c-preference can be constructed based on the population activity of these pools. In this case $P^i = Q^i$ is the total number of columns in pool i, and p_s^i denotes positive activity, i.e. is the number of excitatory activated columns of pool i in interval Δt_s, respectively q_s^i denotes the negative activity in the pool, i.e. the number inhibitory activated columns.

3.2.2 Synfire Chains as a Preference Moore Machine

A c-preference will represent the activity of one or several pools in the chain at particular time. To integrate the sequence of firing of the pools in the synfire chains, we develop a sequence of c-preferences representing the activity at each time step and construct a preference Moore machine that would be able to represent the behavior of synfire chains in the network. A direct interpretation of the representation of a cognitive event in the synfire chain (which is an ordered sequence of synchronous firing of neuronal pools in the chain) would be a final state (or set of states) of a preference Moore machine. If the network has activated only one event, the final state would be the one representing that event. Respectively if the network activates several cognitive events, the preference machine will have multiple final states at the end, each representing a particular event. The intermediate state(s) of the machine represent the history of the firing patterns of the network. A repeated intermediate state(s) sequence would indicate the cycling activity of the network.

4 Initial Case Study: A preference analyzer based on Pulse Neural Networks

We are currently working on a pulsed neural network model for semantic understanding based on feature binding and slot filling. The network consists of three layers of integrate-and-fire spiking neurons and uses a model of a cortical column as a functional unit (Figure 3.). The input to the network is a sequence of role/filler pairs representing a single sentence. The input layer receives a distributed representation of a role and/or a filler at a time. The working (short-term) memory layer has uniformly decaying weights and in the absence of an input, the activation slowly vanishes as the weights become weaker. This layer contains the current context of the sentence. It also retrieves constraints about possible role/filler pairs from the long-term memory layer. The task of the working memory is to combine the current input, context and the constraints from the long-term memory.

When performing a parallel constraint satisfaction, this layer enters into a mode of partial synchrony firing where the features of the assigned role/filler pairs are bound together. To perform this task, while receiving the input, the working memory layer performs an online spatio-temporal correlation-based (Hebbian) learning based on the current input and the constraints from the long-term memory, i.e. the neurons in the working memory have dynamic synapses reflecting the current context of the sentence. In contrast, the synapses in the long-term memory layer change over a much longer time scale. The long-term memory layer should contain possible combinations of role/filler pairs and contexts. During processing, the long-term memory receives the current context from the working memory, and its task is to place constraints on the current context and feed it back to the working memory.

At a higher level of abstraction, such a model can also be viewed as a preference Moore machine (I, O, S, f_p). We can define the input set I to contain all simple preferences representing the network input set using the well-known mean firing rate code. The output set O will contain the interpretation of the working memory in a synchrony code as cell assemblies. The state set S of the machine will contain complex preferences that are an interpretation of the layers of working and long-term memory as cell assemblies. Such a preference Moore machine will perform a mapping from the input preferences which can be symbolically interpreted as roles and words in a sentence and the current state as a complex preference into an output set of role/filler pairs and a new state of the working and long-term memory.

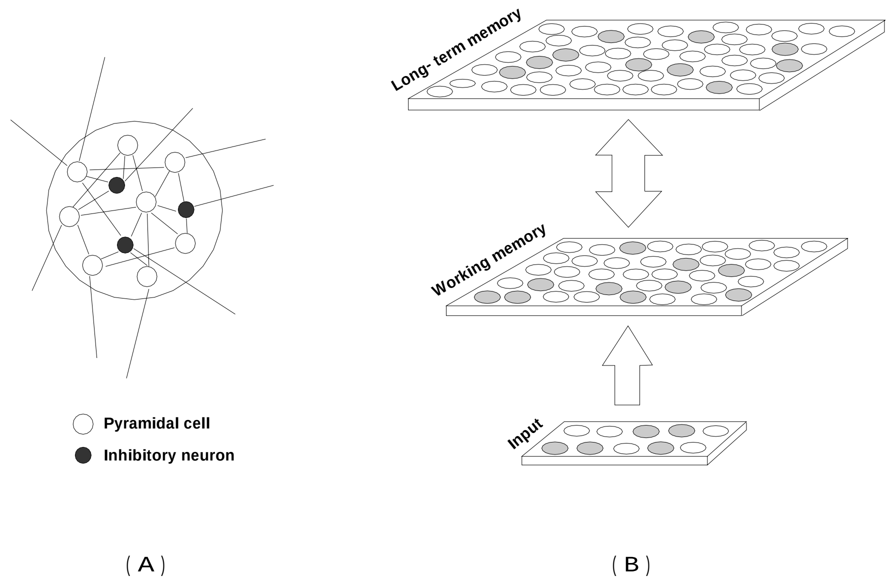

Figure 3: Example of a pulsed neural network model of semantic slot filling based on Hebbian assemblies of cortical columns. (A) Model of a cortical column used in the network. There is dense connectivity between the neurons within the column: inhibitory links from the inhibitory to the pyramidal cell and excitatory links between the pyramidal cells. (B) The three-layered model of the network. Each circle represents a single column from A. There is sparse connectivity between the neurons within a layer. However, on a single column level, the layer is fully connected. Furthermore, there are feedforward connections from the input layer to the working memory layer and from the working memory to the long-term memory. Feedback connections are from the columns of the long-term memory to the working memory.

5 Discussion and Conclusion

In this paper we have explored the use of preference Moore machines at symbolic and connectionist levels but, in particular, we have made some new contributions based on the neuroscience level. We argue that in the long run it will be necessary to understand more of the underlying neuronal processing and that symbolic, connectionist and neuroscience levels are useful levels of abstraction. By considering the neuroscience level, important new insights may be gained for higher symbolic connectionist levels.

In comparison to symbolic structured methods, e.g. symbolic chart parsers and context free grammars, this approach may seem very reductionist from a language processing point of view. However, we have to consider that symbolic Moore machines and connectionist preference Moore machines already support very important general properties of language and they form the basis in the Chomsky hierarchy [Hopcroft and Ullman, 1979].

In contrast to other research work in the area of finite automata and connectionist networks [Manolios and Fanelli, 1994, Omlin and Giles, 1994, Omlin and Giles, 1996], we do not only want to learn an acceptor which learns to accept a correct input sequence but we are interested in building robust learning preference Moore machines which can produce output. Furthermore, an important aspect of our work is that we want to ground these processing mechanisms in constraints which have been known from cognitive and biological neuroscience.

At the symbolic/connectionist level, connectionist preference Moore machines develop internal states which have distributed representations and therefore have very different representations than traditional symbolic automata or statistical HMM models. There is potentially an unlimited number of states in preference Moore machines, different from fuzzy automata or statistical automata [Kosanovic, 1995, Omlin et al., 1995]. Furthermore, if we look at the connectionist level and neuroscience level, preference Moore machines are extended by the time aspect in the representation coding.

Architectural abstractions at different levels are important in order to link higher level cognitive functions like language processing with the neuroscience level. The complexity of cognitive and neurobiological processes makes it seem plausible that several representational levels may be advantageous [Gutknecht, 1992, Sun, 1996]. Furthermore, although the cortex is relatively flatly structured compared to its size [Hubel and Wiesel, 1979], it is structured and far from random. It is partially predetermined at birth but also develops much further in particular in the first years [Bloom, 1993]. We would like to conclude that abstraction levels at symbolic, connectionist and neuroscience levels may be advantageous and that the preference approach is one promising way to link these levels for higher cognitive processing.

Acknowledgments

This research has been partially supported by EPSRC grant GR/M56555. We would like to tank Garen Arevian for his comments on this paper.

References

[Abeles et al., 1993] Abeles, A., Vaadia, E., Bergman, H., Prut, Y., Headman, I., and Slovin, H. (1993). Dynamics of neuronal interactions in the frontal cortex of behaving monkeys. *Concepts in Neuroscience*, 4:131–158.

[Abeles, 1982] Abeles, M. (1982). *Local Cortical Circuits: An Electrophysiological Study.* Springer-Verlag.

[Abeles, 1991] Abeles, M. (1991). *Corticonics: Neural Circuits of the Cerebral Cortex.* Cambridge University Press.

[Aertsen et al., 1996] Aertsen, A., Diesmann, M., and Gewaltig, M. (1996). Propagation of synchronous spiking activity in feedforward neural networks. *Journal of Physiology (Paris)*, 90:243–247.

[Anderson and Rosenfeld, 1988] Anderson, J. and Rosenfeld, E., editors (1988). *Neurocomputing: Foundations of Research.* MIT Press, Cambridge.

[Bienenstock, 1995] Bienenstock, E. (1995). A model of neocortex. *Network*, 6:179–224.

[Bloom, 1993] Bloom, P. (1993). *Language acquisition.* Harvester Wheatsheaf, New York.

[Denham and Denham, 1999] Denham, S. and Denham, M. (1999). An investigation into the role of cortical synaptic depression in auditory processing. In *Proceedings of the EmerNet: International Workshop on Emergent Neural Computational Architectures Based on Neuroscience, ICANN '99*, Edinburgh, Scotland.

[Elman et al., 1996] Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., and Plunkett, K. (1996). *Rethinking Innateness.* MIT Press, Cambridge, MA.

[Fransén and Lansner, 1998] Fransén, E. and Lansner, A. (1998). A model of cortical associative memory based on a horizontal network of connected columns. *Network*, 9(2):235–264.

[Fransén et al., 1994] Fransén, E., Lansner, A., and Liljenström, H. (1994). A model of cortical associative memory based on Hebbian cell assemblies. In Niklasson, L. F. and Bodén, M. B., editors, *Connectionism in a Broad Perspective: Selected papers from the Swedish Conference on Connectionism — 1992*, pages 165–172, Skövde, Sweden. Ellis Horwood.

[Gutknecht, 1992] Gutknecht, M. (1992). The postmodern mind: hybrid models of cognition. *Connection Science*, 4:339–364.

[Hebb, 1949] Hebb, D. (1949). *The Organization of Behavior.* Wiley, New York. Partially reprinted in [Anderson and Rosenfeld, 1988].

[Herrman et al., 1995] Herrman, M., Hertz, J., and Plügel-Bennet, A. (1995). Analysis of synfire chains. *Netowrk: Computation in Neural Systems*, 6(3):403–414.

[Hopcroft and Ullman, 1979] Hopcroft, J. and Ullman, J. (1979). *Introduction to Automata Theory, Languages, and Computation.* Addison Wesley, Reading, MA.

[Hubel and Wiesel, 1979] Hubel, D. H. and Wiesel, T. N. (1979). Brain mechanisms of vision. *Scientific American*, September.

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

[Kosanovic, 1995] Kosanovic, B. R. (1995). Signal and system analysis in fuzzy information space. Technical Report Dissertation, University of Pittsburgh.

[Maass and Bishop, 1999] Maass, W. and Bishop, C. (1999). *Pulsed Neural Networks.* MIT Press.

[Manolios and Fanelli, 1994] Manolios, P. and Fanelli, R. (1994). First-order recurrent neural networks and deterministic finite state automata. *Neural Computation*, 6:1155–1173.

[Omlin and Giles, 1994] Omlin, C. W. and Giles, C. L. (1994). Extraction and insertion of symbolic information in recurrent neural networks. In Honavar, V. and Uhr, L., editors, *Artificial Intelligence and Neural Networks: Steps towards principled Integration*, pages 271–299. Academic Press, San Diego.

[Omlin and Giles, 1996] Omlin, C. W. and Giles, C. L. (1996). Extraction of rules from discrete-time recurrent neural networks. *Neural Networks*, 9(1):41–52.

[Omlin et al., 1995] Omlin, C. W., Thornber, K. K., and Giles, C. L. (1995). Fuzzy finite-state automata can be deterministically encoded into recurrent neural networks. Technical Report CS-TR-3599, University of Maryland, College Park.

[Palm, 1986] Palm, G. (1986). Associative networks and cell assemblies. In Palm, G. and Aertsen, editors, *Brain Theory*. Springer.

[Panchev and Wermter, 2000] Panchev, C. and Wermter, S. (2000). Complex preferences for the integration of neural codes. In *Proceedings of the International Joint Conference on Neural Networks*, page July, Como, Italy.

[Sun, 1996] Sun, R. (1996). Hybrid connectionist-symbolic models: A report from the IJCAI95 workshop on connectionist-symbolic integration. *Artificial Intelligence Magazine*.

[Taylor, 1999] Taylor, J. (1999). Images of the mind: Brain imaging and neural networks. In Wermter, S., Austin, J., and D., W., editors, *Proceedings of the EmerNet: International Workshop on Emergent Neural Computational Architectures Based on Neuroscience, ICANN '99*, Edinburgh, Scotland.

[Thorpe et al., 1996] Thorpe, S., Fize, D., and Marlot, C. (1996). Speed in the processing in the human visual system. *Nature*, 384:520–522.

[Wennekers and Palm, 1999] Wennekers, T. and Palm, G. (1999). Cell assemblies, associative memory and temporal structure in brain signals. In Miller, R., editor, *Time and brain. Conceptual Advances in Brain Research*, volume 2. Harwood Academic Publishers. to appear.

[Wermter, 1999] Wermter, S. (1999). Preference Moore machines for neural fuzzy integration. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 840–845, Stockholm.

[Wermter, 2000] Wermter, S. (2000). Knowledge extraction from transducer neural networks. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Techniques*, 12:27–42.

[Wermter, pear] Wermter, S. (2000 to appear). Neural fuzzy preference integration using neural preference moore machines. *International Journal of Neural Systems*.

[Wermter et al., 1999] Wermter, S., Austin, J., and Willshaw, D. (1999). *Proceedings of the Emernet International Workshop on Emergent Neural Computational Architectures based on Neuroscience*. http://www.his.sunderland.ac.uk/emernet/Publications_frame.html, Edinburgh.