Pergamon

S0952-1976(96)00002-4

Engng Applic. Artif. Intell. Vol. 9, No. 2, pp. 121–127, 1996

0952-1976/96 $15.00+0.00

Contributed Paper

Distributed Memory and Localized Memory in Neural Networks*

LIPO WANG
Deakin University, Australia

(Received October 1995)

Two major "categories" of unsupervised learning rules are used in artificial neural networks: (i) competitive learning, which is used in the adaptive resonance theory (ART), the self-organizing map, and the neocognitron; and (ii) Hebbian learning without lateral inhibition, which is used in the Hopfield network. Since the competitive learning is essentially Hebbian learning in the presence of lateral inhibition, the author attempts here to discuss general properties of these unsupervised learning rules in a unified paradigm. As a first effort, this paper presents analytical studies of a performance comparison between a competitive learning neural network (CLNN) and the Hopfield neural network (HNN). Specifically, it discusses their abilities as classifiers after they are trained with noisy patterns. First, the HNN is generalized to perform pattern classification in addition to its well-known capability for pattern completion. The Hopfield formulation of the Hebbian learning rule is generalized to allow the existence of noise in training patterns. It is shown that the performance of the generalized HNN as a classifier decreases as noise in training patterns increases. A parallel study is then carried out for a CLNN. First, a simple CLNN is developed with the same components used by the generalized HNN and features used in existing CLNNs. In contrast, this simple CLNN is shown to be robust with respect to noise in training patterns. These discussions suggest that the reason for this difference in performance between the two types of networks is that in the CLNN each synapse is devoted to only one memory, whereas in the HNN each synapse is responsible for many memories. It is concluded that competitive learning, which leads to localized memory, is superior to Hebbian learning without lateral inhibition, which leads to distributed memory, at tolerating noise in training patterns.

Keywords: Pattern recognition, neural networks, training noise.

1. INTRODUCTION

One of the most fascinating features of neural systems, no matter whether biological or artificial, is their ability to learn from experience. As a result of intensive research activity in recent decades, especially during the last few years, a large body of work on learning theory has been produced. Supervised learning algorithms, such as the least-mean-square learning[1] and the back-error-propagation learning[2] rules, are widely discussed and applied; however, they are computationally expensive and it is unlikely that the brain could use these adaptive algorithms.

In contrast, unsupervised learning rules, which use *local* or *synaptic* level criteria rather than some *global* constraints, are more biologically realistic and are often more computationally efficient. Unsupervised learning algorithms have been discussed mainly under two "categories": competitive learning and Hebbian learning without lateral inhibition.

From a computational point of view, Hebbian learning[3] is that if cell A consistently takes part in firing cell B, or fires simultaneously with cell B, the synaptic strength from cell A to cell B increases. Thus learning occurs with *coincidence* in neuronal activities. There have been many implementations of the Hebbian learning rule (see Ref. 4 for a review). Perhaps the most well-known implementation is the Hopfield neural network (HNN),[5,6] which has been largely credited as one of the main reasons for the recent resurgence of neural-network research.

Much work has been carried out using learning rules

Correspondence should be sent to: Lipo Wang, School of Computing and Mathematics, Deakin University, 662 Blackburn Road, Clayton, Victoria 3168, Australia.

* This work was presented at the International IEEE Symposium on Intelligence in Neural and Biological Systems, 29–31 May 1995, Herndon, Virginia, U.S.A. A shorter version of this paper has appeared in the conference proceedings (N. G. Bourbakis, Chair).

in a seemingly different category, i.e. competitive learning, which includes a competition among some or all neurons.[7] Grossberg,[8] von der Malsberg,[9] Kohonen,[10] and Fukushima[11] are among the best-known pioneers, and have developed remarkable artificial neural-network models using competitive learning, for example, the adaptive resonance theory (ART),[8,12] the self-organizing map,[10] and the neocognitron.[11]

In fact, competitive learning *is* essentially Hebbian *coincidence* learning in the presence of lateral inhibition. Depending on the strength and distribution of lateral inhibition, there should be a continuum of adaptive neural systems. At one extreme of the continuum is the Hopfield prescription of Hebbian learning, where no lateral inhibition exists; whereas at the other extreme are, for instance, the ART[12] and the Rumelhart–Zipser model,[13] which use the "winner-take-all" rule facilitated by maximum lateral inhibition. Kohonen's self-organizing map is an example that is in between these two extremes and uses lateral inhibition of limited range and strength.

As a first effort to discuss the general properties, such as the computational advantages and disadvantages, of these unsupervised learning rules in a *unified* paradigm, this paper presents analytical studies on a performance comparison between a competitive learning neural network (CLNN) and the Hopfield neural network (HNN). Specifically, it discusses the capability of these two types of neural networks as classifiers after they are trained with noisy patterns.

The structure of the paper is as follows. First, the HNN[5] is generalized to perform pattern classification in addition to its well-known capability for pattern completion. The Hopfield formulation of the Hebbian learning rule is generalized to allow for noise in training patterns. The memory capacity of the HNN, which is the total number of pattern categories the network is able to store and classify, is calculated analytically. It is shown that the performance of the generalized HNN as a classifier decreases as the noise in the training patterns increases.

A parallel study is then carried out for a CLNN. First, a simple CLNN is developed, with the same components as those used in the generalized HNN and features of some existing CLNNs, that is, a learning rule proposed by von der Malsburg[9] and algorithms used in the ART model[12] and the Rumelhart–Zipser model.[13] The reason for developing this CLNN rather than directly discussing an existing neural network model is three-fold: (i) it is desired to isolate the effects of the maximum lateral inhibition or the winner-take-all rule; (ii) the intention is to discuss a neural-network model with exactly the same degree of structural and computational complexity as the HNN, so that the performance comparison is a fair one; (iii) the objective of this work is to carry out *analytical* discussions, rather than relying on numerical simulations of limited generality. As will be demonstrated later, the CLNN, in contrast, is robust with respect to noise in training patterns.

These discussions suggest that the reason for this difference in performance between the two types of networks is that, in the CLNN, each synapse is devoted to only one memory, whereas in the HNN each synapse is responsible for many memories. It can thus be concluded that competitive learning, which leads to *synaptically localized* memory, is superior to Hebbian learning without lateral inhibition, which leads to *synaptically distributed* memory, at tolerating noise in training patterns.

2. TRAINING THE GENERALIZED HOPFIELD NEURAL NETWORK WITH NOISY PATTERNS

2.1. The Hopfield neural network

Recently there have been numerous studies of Hopfield-type artificial neural networks.[5,6,14–24] This paper presents an introduction to the HNN, and establishes a basis for the subsequent discussions.

The neural-network model discussed by Hopfield[5] consists of N McCulloch–Pitts neurons[19] that have two states: firing and quiescent, or, $S_i = \pm 1$, where $i = 1, \ldots, N$. Each neuron receives signals from its neighboring neurons, and the signals are transmitted through synaptic weights T_{ij}. The neuron then either fires if the total input h_i exceeds a threshold, or remains quiescent. These model systems, though crude compared to biological neural systems, already display intriguing features, such as a form of learning and recall of associative memory. Another property that makes Hopfield-type models interesting is that one can treat them mathematically, and make analytic statements about them without relying heavily on computer simulations.

Quantitatively, the ith neuron obeys the following response rule:

$$S_i(t+\Delta t) = \mathrm{sign}\,[h_i(t)], \tag{1}$$

where $\mathrm{sign}(x) = +1$, for >0 and $\mathrm{sign}(x) = -1$, for $x \leq 0$; and the total input h_i that the ith neuron receives at time t is given by:

$$h_i(t) = \sum_{j=1}^{N} T_{ij} S_j(t) + \eta_i, \tag{2}$$

η_i models the noise in neuronal signals due to the probabilistic release of synaptic vesicles and neurotransmitters that accounts for the spontaneous firing of a neuron.[14–17,25] It is assumed that η_i has an average zero and a standard deviation σ_0.

Hopfield[5] studied a fully connected network in which neurons are updated sequentially, and synaptic connections are chosen to be:

$$T_{ij}^H = N^{-1} \sum_{\mu=1}^{p} S_i^\mu S_j^\mu, \tag{3}$$

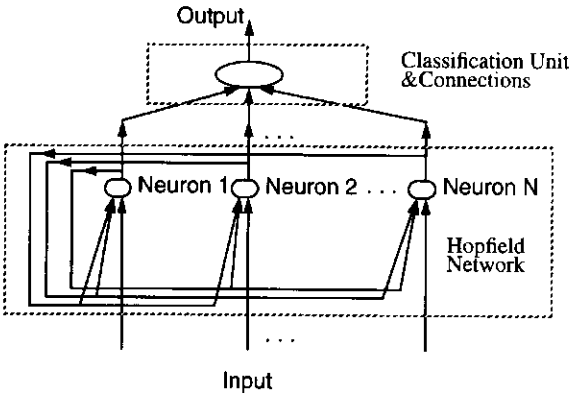

Fig. 1. Architectural structure for the generalized Hopfield neural network (HNN). The lower half represents the original Hopfield network. The upper half represents the newly-added linear classification unit and its connections.

according to the Hebbian rule,[3] $\mathbf{S}^\mu \equiv \{S_1^\mu, S_2^\mu, \ldots, S_N^\mu\}$ is the μth stored binary pattern, and p is the number of stored patterns. Patterns $\{\mathbf{S}^\mu\}$ are randomly generated so that they are (quasi-)orthogonal. The architecture of a Hopfield network is shown in the lower half of Fig. 1.

An energy function can be defined as a function of neuronal states, and is shown to decrease monotonically as neurons are updated when $\delta_0=0$. Since the energy is finite, and the stored patterns are shown to be energy minima, neurons in the HNN update until an energy minimum (a memory) is reached. This unique property makes the HNN suitable for tasks such as pattern completion[18] and combinatorial optimization,[26] e.g. for solving the travelling salesman problem (TSP). In the TSP, a network is designed such that the energy function of the system is equal to the total cost of travel and the network is able to find a low-cost travel route after a certain relaxation time. In the case of pattern completion, p random patterns can be stored in the HNN by using the Hebbian learning rule given in equation (3). When the network is presented with a noisy input, the initial state $\mathbf{S}(t=0)$ is set to be the same as this input pattern. If this input pattern is most similar to one of the stored patterns, neurons in the network will update recurrently according to the neuronal response rule given by equations (1) and (2), until the network stabilizes itself at this stored pattern and the input pattern is considered to have been *completed*.

The equilibrium properties of the HNN have been studied[16,17] using powerful tools in statistical physics, and the system is shown to make a transition to a disordered (no-memory) state if the number of stored patterns p and the standard deviation of the noise in signal transmission (δ_0) exceed certain critical values. For instance, when $\sigma_0=0$, the network has the highest memory capacity, i.e. the maximum number of random patterns that can be stored by the network is $P=0.138N$, where N is the total number of neurons in the network. When $\sigma_0\neq 0$, the memory capacity P of the HNN is determined[5] by solving the following coupled equations at a critical P/N where nontrivial solutions ($m>0$) disappear:

$$m=\int \frac{\mathrm{d}z}{\sqrt{2\pi}}\,\mathrm{e}^{-z^2/2}\tanh\left(\frac{\sqrt{\alpha r}z+m}{T}\right), \qquad (4)$$

$$g=\int \frac{\mathrm{d}z}{\sqrt{2\pi}}\,\mathrm{e}^{-z^2/2}\tanh^2\left(\frac{\sqrt{\alpha r}z+m}{T}\right), \qquad (5)$$

$$r = g/(1 - T^{-1}+gT^{-1})^2. \qquad (6)$$

Here $\alpha\equiv P/N$; $m=\mathbf{S}\cdot\mathbf{S}^\mu/N$ is the overlap between the state of the network and a memory pattern. The "temperature" T is related to the standard deviation σ_0 of $\boldsymbol{\eta}$ simply by:[14]

$$\sigma_0=\sqrt{2/\pi}\,T=0.798T. \qquad (7)$$

Note that the temperature T in equations (4)–(7) is actually twice the temperature introduced by Little.[14,15] These results will be used in the subsequent analysis.

2.2. A generalized HNN that performs pattern classification

As stated above, the HNN performs pattern *completion* and the HNN by itself cannot serve as a classifier: it outputs one of the stored patterns when presented with an input, and one extra step is needed to classify this output. One possible method to do so could be as follows. After the HNN reaches a memory state, all N neurons are connected to a *linear* output device through N output synapses, which form a vector $\mathbf{A}\equiv\{A_1, A_2, \ldots, A_N\}$ (see the upper half of Fig. 1). $\mathbf{A}$ is chosen in such a way that the output of the added linear device is μ' if the HNN reaches the memory state $\mathbf{S}^{\mu'}$, i.e. $\mathbf{A}\cdot\mathbf{S}^{\mu'}=\mu'$ for all μ'. The following design fulfils this requirement:

$$\mathbf{A}=N^{-1}\sum_{\mu=1}^{p}\mu\mathbf{S}^\mu, \qquad (8)$$

if $\{\mathbf{S}^\mu\}$ are orthogonal. The HNN has thus been generalized to perform pattern-classification tasks by the addition of an output device and N output synapses to the original HNN.

2.3. A generalized Hebbian learning rule for noisy training patterns

If training patterns deviate from the standard memories $\{\mathbf{S}^\mu\}$ that need to be stored, the Hebbian rule given by equation (3) is generalized as follows:

$$T_{ij}=(qN)^{-1}\sum_{\mu=1}^{p}\sum_{\nu=1}^{q} S_i^{\mu\nu}S_j^{\mu\nu}, \qquad (9)$$

where $\{\mathbf{S}^{\mu\nu}|\mu=1,2,\ldots,p;\nu=1,2,\ldots,q\}$ are the training patterns. Equation (9) reduces to the original Hopfield prescription equation (3) if $\mathbf{S}^{\mu\nu}=\mathbf{S}^\mu$ for all ν

and μ. For the analysis undertaken here, the following form of noise in the training patterns is chosen:

$$S_j^{\mu\nu} = S_j^\mu + \delta_j^{\mu\nu}, \tag{10}$$

where it is assumed that the differences between the training patterns and the standard patterns, $\{\delta_j^{\mu\nu}\}$, which may take values 0, ± 2, are independent random numbers with a zero average and a standard deviation δ, i.e.

$$\langle \delta_j^{\mu\nu} \rangle = \frac{1}{p}\sum_{\mu=1}^{p} \delta_j^{\mu\nu} = \frac{1}{N}\sum_{j=1}^{N} \delta_j^{\mu\nu} = 0, \tag{11a}$$

and similarly,

$$\langle (\delta_j^{\mu\nu} - \langle \delta_j^{\mu\nu} \rangle)^2 \rangle = \langle (\delta_j^{\mu\nu})^2 \rangle = \delta^2. \tag{11b}$$

For instance, $\delta^2 = 1$ indicate that $\delta^2/4 = 25\%$ of the bits in $\mathbf{S}^{\mu\nu}$ are randomly chosen and flipped, since each bit flipped gives $(\delta_i^{\mu\nu})^2 = 4$. Thereafter it is assumed that $\delta^2 < 0.5$ so that despite the noise in the training patterns, $\mathbf{S}^{\mu\nu'}$ is more similar to $\mathbf{S}^{\mu\nu}$ than $\mathbf{S}^{\mu'\nu''}$ is to $\mathbf{S}^{\mu\nu}$ for any set of (ν, ν', ν'') if $\mu \neq \mu'$. Thus the noisy training pattens form broadened bands around the standard patterns $\{\mathbf{S}^\mu\}$, with band gaps on the order of $0.25N$ bits.

2.4. Training the generalized HNN with noisy patterns

Substituting equations (9) and (10) into equation (2) yields

$$h_i(t) = \sum_{j=1}^{N} T_{ij}^{II} S_j(t) + (qN)^{-1} \left\{ \sum_{j=1}^{N} \sum_{\mu=1}^{p} \sum_{\nu=1}^{q} S_i^\mu \delta_j^{\mu\nu} S_j(t) + \sum_{j=1}^{N} \sum_{\mu=1}^{p} \sum_{\nu=1}^{q} \delta_i^{\mu\nu} S_j^\mu S_j(t) + \sum_{j=1}^{N} \sum_{\mu=1}^{p} \sum_{\nu=1}^{q} \delta_i^{\mu\nu} \delta_j^{\mu\nu} S_j(t) \right\} + \eta_i. \tag{12}$$

In equation (12) the terms in the curly brackets on the right-hand side represent the effects of noise in the training patterns. According to equation (11), it is known that they have zero averages and a total standard deviation:

$$\sigma_H^2 = \frac{p}{Nq}(2\delta^2 + \delta^4). \tag{13}$$

The memory capacity P of the HNN trained with noise patterns, which is the number of patterns the network is able to store and classify, is then determined by equations (4)–(7), with σ_0 in equation (7) replaced by σ_H given in equation (13). Figure 2 shows P plotted against δ^2. The numerical solution of these equations indicates that in the absence of noise in training patterns, i.e. $\delta^2 = 0$, the memory capacity P reaches a maximum, which is $0.138N$, and decreases monotonically as δ^2 increases. For instance, if $\delta^2 = 0.45$ and $q = 1$, the memory capacity is $0.09N$, a 35% reduction.

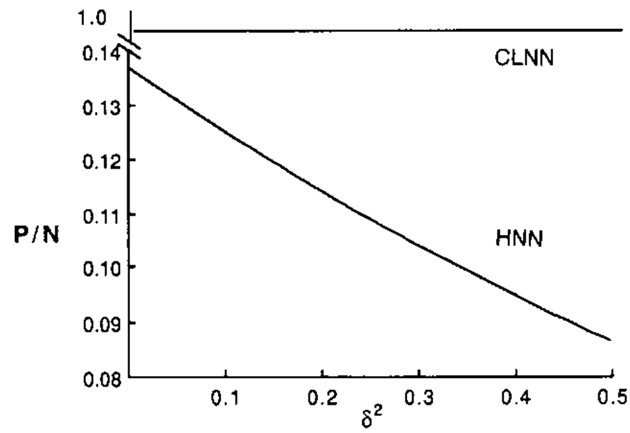

Fig. 2. Comparison of performance (memory capacity P) between the simple competitive learning neural network and the generalized Hopfield neural network, in terms of noise in training patterns δ^2, for $q = 1$.

From equation (3), the Hopfield prescription of Hebbian learning rule, it can be observed that each synapse is responsible for many memories. From the nature of Hebbian coincidence learning,[3] this property is not restricted to the Hopfield prescription, but is universal for all mathematical implementations:[4] Hebbian learning without lateral inhibition leads to *synaptically distributed* memory.

3. TRAINING WITH NOISY PATTERNS IN A COMPETITIVE LEARNING NEURAL NETWORK

In order to isolate the effects of lateral inhibition, it is necessary to discuss a neural-network model with a very simple architecture. To carry out a fair performance comparison with the generalized HNN, it is essential to make sure that the CLNN being discussed has exactly the same degree of structural and computational complexity compared to the generalized HNN. Furthermore, to avoid relying on numerical simulations, which usually have limited generality, the CLNN should be analytically tractable. For these reasons a simple CLNN will now be developed, rather than using an existing CLNN.

The new CLNN has exactly the same set of components as the generalized HNN classifier: a layer of N McCulloch–Pitts neurons and one linear output device (see Fig. 3). Each neuron has N input synapses that can be modified during learning. The input synapses of the linear output device are connected to the outputs of the neurons.

The crucial difference between the present CLNN and the generalized HNN classifier discussed in the previous section is that there exists maximum lateral inhibition among the N neurons. That is, for any given system input, the lateral inhibition is so strong that only the most strongly responding neuron has an output, and other neurons remain quiescent. This "winner-

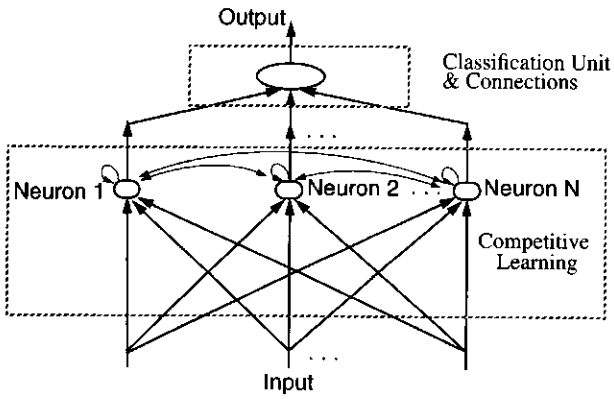

Fig. 3. Architectural structure for the simple competitive learning neural network. The lower half represents a "winner-take-all" competitive learning network. The upper half represents a linear classification unit and its connections.

take-all" rule has been used in the ART[12] and the Rumelhart–Zipser model.[13]

Before proceeding, observe that the total signal that a neuron receives, i.e. the first term in equation (2), is in fact a dot-product between the input pattern **S** and the vector formed by the synapses of the ith neuron, $\mathbf{T}_i$. The better matched the input pattern is with the synapses, the larger the signal. Therefore, from a computational point of view, an artificial neuron described by equation (2) is essentially a template-matcher, its excitation reflecting a good match between the input and its synapses. The neuron whose synapses are most similar to the input will be the most excited neuron, and this neuron will win the competition by shunting all the other neurons. Thus *coincidence* learning only occurs at the best matched neuron.

The learning algorithm used here is such that the set of synapses for each neuron represents a cluster of training patterns that are sufficiently similar to each other. This resembles the sequential leader clustering algorithm described by Hartigan.[27,28] The detailed algorithm for the present CLNN is stated as follows.

The first input is assigned to the synapses of a neuron as the exemplar for the first cluster. The subsequent input patterns are compared to the synapses of the neurons for the existing cluster exemplar(s). If an input is sufficiently similar to the best matched exemplar, i.e. from a vigilance test,[12] the synapses of the most excited neuron are updated, that is, the cluster exemplar is modified. Otherwise the input is assigned to the synapses of another neuron and a new cluster is created.

The synapse modification rule that will be used for the present CLNN was first proposed by von der Malsburg,[9] and was later used in the Rumelhart–Zipser model.[13] At the τ_ith updating of the synapses of the ith neuron, suppose the input pattern is $\mathbf{I}_k$, the synapses are allowed to give up some portion, i.e. $1/\tau_i$, of its weights and the weights are then distributed among the synapses in proportion to the training input pattern. Following Grossberg,[29] the sum of the synapses will not be normalized to 1, since the synaptic normalization is not necessary to achieve the desired learning properties. Explicitly, the updated synapses are:

$$T_{ij}(\tau_i) = \left(1 - \frac{1}{\tau_i}\right) T_{ij}(\tau_i - 1) + \frac{1}{\tau_i} I_{jk} = \frac{1}{\tau_i} \sum_{\tau'=1}^{\tau_i} I_{jk}(\tau'), \qquad (14)$$

which implies that the synapse vector is an overall average of the contributing training patterns, and all contributing training patterns, independent of the temporal order in which the training patterns are presented, contribute *equally* to learning.

Using equation (14) and the noisy training patterns described in the previous section, i.e. equation (10), gives an analytical expression of the synapses after training (with a vigilance threshold for cluster creation equivalent to a mismatch of $0.25N$ bits):

$$T_{\mu j} = S_j^{\mu} + q^{-1} \sum_{\nu=1}^{q} \delta_j^{\mu\nu}, \qquad (15)$$

where $\mu = 1, 2, \ldots, p$. Hence p neurons are used in the learning process, i.e. p clusters are created and modified during training.

It is required that, as in the generalized HNN classifier discussed in the previous section, the output of the linear device is μ' if the input pattern is classified as $\mathbf{S}^{\mu'}$. Since only one neuron carries an output for any given input pattern, this requirement can be fulfilled by letting:

$$A_{\mu'} = \mu', \qquad (16)$$

where $\mu' = 1, 2, \ldots, p$, and $A_{\mu'}$ is the synapse that connects the output of the μ'th neuron to the linear output device of the present CLNN (see Fig. 3).

From equation (15) it can be observed that after training with noisy patterns, the synapses of the participating neuron consist of the standard memories $\{\mathbf{S}^{\mu}\}$ and noise terms with a standard deviation:

$$\delta_D^2 = \delta^2 / q. \qquad (17)$$

Since $q \geqslant 1$ and it has been assumed that $\delta^2 < 0.5$, $\delta_D^2 < 0.5$. Hence the cluster examplars stored in the synapses differ from each other by at least 25% of the N bits, and the present CLNN can classify $P = N$ patterns after training with noisy patterns. Therefore, the memory capacity of the CLNN is larger than the generalized HNN classifier trained by patterns with or without noise. Furthermore, the noise in the training patterns does not decrease the memory capacity of the CLNN, whereas any amount of noise in the training patterns will decrease the memory capacity of the HNN, as shown in Fig. 2.

Equation (15) shows that in the CLNN, each neuron, as well as each synapse within a neuron, is responsible for only one memory. From the analysis presented in this section, it can be seen that the winner-take-all competitive learning facilitated by maximum lateral inhibition leads to *synaptically localized* memory, in contrast with Hebbian learning without lateral inhibition, which leads to *synaptically distributed* memory, as pointed out in the previous section. The discussions presented in this paper suggest that the difference in the abilities to tolerate noise in training patterns is a result of the difference in memory distribution in neural networks.

4. SUMMARY AND DISCUSSIONS

This paper presents analytical studies of a performance comparison between a competitive learning neural network (CLNN) and the Hopfield neural network (HNN), which uses Hebbian learning without lateral inhibition. Specifically, it studies how their abilities as pattern classifiers are influenced by noise in training patterns. The HNN is first generalized to perform pattern classification in addition to its known capability for pattern completion. The Hopfield formulation of the Hebbian learning rule is generalized to allow for the existence of noise in training patterns and the memory capacity of the HNN is calculated analytically. It is shown that the performance of the generalized HNN as a classifier decreases as the noise in the training patterns increases. A parallel study is then carried out for a simple CLNN with the same components used by the generalized HNN and features used in existing CLNNs. In contrast, this simple CLNN is shown to be robust with respect to noise in training patterns, i.e. the CLNN is able to store and classify more patterns than the HNN in the presence or absence of training noise. This analysis suggests that the reason for this difference in performance between the two types of networks is that in the CLNN each synapse is devoted to only one memory, whereas in the HNN each synapse is responsible for many memories. It is concluded that competitive learning, which leads to localized memory, is superior to Hebbian learning without lateral inhibition, which leads to distributed memory, at tolerating noise in training patterns.

This paper represents a first effort to discuss the general properties of unsupervised learning rules in a unified paradigm. Other general properties, such as the computational advantages and disadvantages of various forms of lateral inhibition will be investigated in the present framework, and work will continue towards obtaining an optimal learning algorithm.

REFERENCES

1. Widrow B. and Hoff M. E., Adaptive switching circuits. *1960 IRE WESCON Convention Record*, pp. 96–104 (1960).
2. Rumelhart D. E. and McClelland J. L., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, I and II. MIT Press, Cambridge, MA (1986).
3. Hebb D. O., *The Organization of Behavior*, p. 62. Wiley, New York (1949).
4. Brown T. H., Kairiss E. W. and Keenan C. L., Hebbian synapses: biophysical mechanisms and algorithms. *Ann. Rev. Neurosci.* **13**, 475–511 (1990).
5. Hopfield J. J., Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl Acad. Sci. USA* **79**, 2554–2558 (1982).
6. For a review on Hopfield-type neural networks, see, e.g. Sompolinsky H., Statistical mechanics of neural works. *Phys. Today* **41**, 70–80 (1988).
7. The earliest origin of the competitive learning rule has not been established. See discussions in Hecht-Nielsen, R. *Neurocomputing*, p. 63. Addison-Wesley, Reading, MA (1990).
8. Grossberg S., *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control.* Reidel Press, Boston (1982).
9. von der Malsburg C., Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik* **14**, 85–100 (1973).
10. Kohonen T., *Self-Organization and Associative Memory.* Springer, Berlin, (1984).
11. Fukushima T., Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **36**, 193–202 (1980).
12. Carpenter G. A. and Grossberg S., Neural dynamics of category learning and recognition: attention, memory consolidation, and amnesia. In *Brain Structure, Learning, and Memory* (Edited by Davis J., Newburgh R. and Wegman E.), AAAS Symposium Series, pp. 223–290 (1986).
13. Rumelhart D. E. and Zipser D., Feature discovery by competitive learning. *Cog. Sci.* **9**, 75–112 (1985).
14. Little W. A., The existence of persistent states in the brain. *Math. Biosci.* **19**, 101–120 (1974).
15. Shaw G. L. and Vasudevan R., Persistent states of neural networks and the random nature of synaptic transmission. *Math. Biosci.* **21**, 207–217 (1974).
16. Amit D. J., Gutfreund H. and Sompolinsky H., Spin-glass models of neural networks. *Phys. Rev. A* **32**, 1007–1018 (1985).
17. Amit D. J., Gutfreund H. and Sompolinsky H., Statistical mechanics of neural networks near saturation. *Ann. Phys.* **173**, 30–67 (1987).
18. Kinzel W., Learning and pattern recognition in spin glass models. *Z. Phys.* **B60**, 205–213 (1985).
19. Wang L. and Alkon D. L. (Eds), *Artificial Neural Networks: Oscillations, Chaos, and Sequence Processing.* IEEE Computer Press, Los Alamitos, CA (1993).
20. Wang L. and Ross J., Synchronous neural networks of non-linear threshold elements with hysteresis. *Proc. Natl Acad. Sci. USA* **87**, 988–992 (1990); Interactions of neural networks: models for distraction and concentration. *Ibid.* **87**, 7110–7114 (1990).
21. Wang L., Pichler E. E. and Ross J., Oscillations and chaos in neural networks: an exactly solvable model. *Proc. Natl Acad. Sci. USA* **87**, 9467–9471 (1990).
22. Wang L. and Ross J., Variable threshold as a model for selective attention, (de)sensitization, and anesthesia in associative neural networks. *Biol. Cybern.* **64**, 231–241 (1991).
23. Wang L. and Ross J., Physical modeling of neural networks. In *Methods in Neurosciences*, Vol. 10, *Computers and Computations in the Neurosciences* (Edited by Conn P. M.), pp. 549–567. Academic Press, San Diego, CA (1992).
24. McCulloch W. S. and Pitts W., A logical calculus of the ideas immanent in nervous activity. *Math. Biophys.* **5**, 115–133 (1943).
25. Abelles M., *Local Cortical Circuits*, p. 21. Springer, New York (1982).
26. Kirkpatrick S., Gelatt C. D. Jr and Vecchi M. P., Optimization by simulated annealing. *Science* **220**, 671–680 (1983).
27. Hartigan J. A., *Clustering Algorithms.* Wiley, New York (1975).
28. Lippmann R. P., An introduction to computing with neural nets. *IEEE ASSP Magazine*, April, pp. 4–22 (1987).
29. Grossberg S., Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors. *Biol. Cybern.* **23**, 121–134 (1976).

AUTHOR'S BIOGRAPHY

Lipo Wang received his Ph.D. in physics from Louisiana State University in 1988 through the CUSPEA program which selected the top-100 physics graduates in China annually. Since then, he has been interested in the theory and application of neural networks. In 1989 he worked at Stanford University as a postdoctoral fellow. In 1990 he was a lecturer in electrical engineering at the University College of the University of New South Wales. From 1991 to 1994 he worked at the National Institutes of Health as a Fogarty Associate. He is now a lecturer at the School of Computing and Mathematics, Deakin University (Rusden Campus). His specific research interests in neural networks include temporal behavior, speech processing, pattern recognition, and biological networks. He is the author or co-author of over 30 journal publications, co-inventor of a U.S. patent on neural networks, and co-editor of *Artificial Neural Networks: Oscillations, Chaos, and Sequence Processing* (IEEE Computer Press).