# Effects of Noise in Training Patterns on the Memory Capacity of the Fully Connected Binary Hopfield Neural Network: Mean-Field Theory and Simulations

Lipo Wang

***Abstract*— We show that the memory capacity of the fully connected binary Hopfield network is significantly reduced by a small amount of noise in training patterns. Our analytical results obtained with the mean field method are supported by extensive computer simulations.**

***Index Terms*— Associative memory, Hebbian learning, Hopfield, neural network, mean-field theory, memory capacity, training noise.**

## I. Introduction

SINCE the publication of Hopfield's classic paper in 1982 [9], there have been numerous studies on the so-called Hopfield neural network and its variants (e.g., [1], [4], [7], [10]–[11], [15], [16], and [18]–[29]). For example, a Hopfield network can be used as content-addressable memory (CAM). After a set of memory patterns are learned by the network, a presentation of a noisy input causes the network to recall a memorized pattern in a successful retrieval. Time-dependent sequences of spatial patterns (spatio-temporal sequences) [11], [18], [19] can also be stored and retrieved with a Hopfield network. The patterns used to train a Hopfield network are generally assumed to be ideal patterns that are free of noise. It is known that the memory capacity, i.e., the maximum number of spatial patterns that can be stored in a fully connected binary Hopfield network (stationary or temporal), is about $0.14N$, $N$ being the number of neurons in the network. In this paper we study the following important question [21]: When one needs to use a Hopfield network as CAM or to store spatio-temporal sequences, and when the training patterns, or training sequences of spatial patterns, are contaminated with noise, how will the memory capacity of the fully connected binary Hopfield network be affected?

We will proceed as follows. In Section II, we present a theoretical analysis using the *mean field* method [7], [15], after generalizing the Hebbian learning rule [8], [3], [9] used in the original Hopfield network to allow for noise in training patterns. The equations that determine the memory capacity of the Hopfield network is derived analytically and the memory capacity is shown to decrease as the noise in the training patterns increases. In Section III, we present results of extensive computer simulations to support our theoretic studies. Closing remarks are presented in Section IV.

Manuscript received March 11, 1997; revised October 12, 1997 and March 29, 1998. This work was supported by the Australian Research Council and Deakin University.

The author is with the School of Computing and Mathematics, Deakin University, Clayton, Victoria 3168, Australia.

Publisher Item Identifier S 1045-9227(98)04452-X.

## II. Mean Field Theory

The binary Hopfield network consists of $N$ McCulloch–Pitts neurons [14] that have two states: firing and quiescent, or, $S_i = \pm 1$, where $i = 1, \cdots, N$. 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 exceeds a threshold, or remains quiescent otherwise [7]. Quantitatively, neuron $i$ receives the following inputs from other neurons:

$$h_i^o = \sum_{j \neq i} T_{ij} S_j. \tag{1}$$

In addition, we consider the noise in neuronal signals due to probabilistic release of synaptic vesicles and neurotransmitters that accounts for the spontaneous firing of a neuron [2]. Similar noise also exists in electronic implementations of neural networks. In the presence of this signal transmission noise, the dynamics of the neuron becomes stochastic and we assume that neuron $i$ updates according the following probability function [13]:

$$\text{Prob}(S_i = \pm 1) = f(\pm h_i^o) = \frac{1}{1 + e^{\mp 2\beta h_i^o}} \tag{2}$$

where $\beta$ is inversely proportional to the standard deviation of the signal transmission noise [13], [17], [29]. Hopfield [9] studied a fully connected network in which neurons are updated sequentially and synaptic connections are chosen to be [3]

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

where $\vec{S}^\mu \equiv \{S_1^\mu, S_2^\mu, \cdots, S_N^\mu\}$ is the $\mu$th stored binary pattern ($S_i^\mu = \pm 1$), and $p$ is the number of stored patterns. Patterns $\{\vec{S}^\mu\}$ are assumed to be randomly generated so that they are quasiorthogonal.

When training patterns are corrupted with noise, how should the learning rule given by (3) be generalized? We discuss the following two possible strategies. Suppose that instead of *clean* training patterns $\vec{S}^\mu$, there exist sets of training patterns

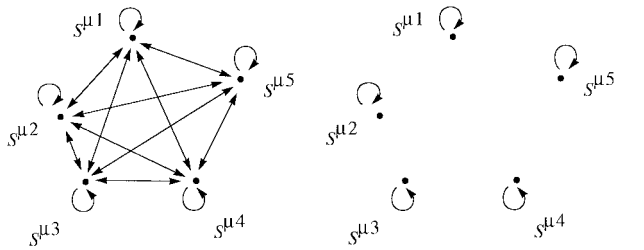


Fig. 1. Schematic representations for two possible generalizations of the learning rule in the presence of training noise. (a) Associative mappings among the noisy training patterns: equivalent to obtaining *approximate clean training patterns* by carrying out averages over the noisy training patterns. (b) Autoassociative mappings between the noisy training patterns themselves: in closer analogy with the Hebbian learning rule.

$\vec{S}^{\mu 1}, \vec{S}^{\mu 2}, \cdots$, that *may* deviate from *clean* training patterns $\vec{S}^{\mu}$, as shown in Fig. 1. The *clean* pattern $\vec{S}^{\mu}$ itself is not explicitly shown, since any number of the noisy patterns may be the same as the *clean* pattern $\vec{S}^{\mu}$, i.e., the noise may take zero values.

To derive the first possible learning strategy for noisy training patterns, we notice that (3) may be understood as a set of mappings of patterns $\{\vec{S}^{\mu}, \mu = 1, 2, \cdots, p\}$ onto themselves, and hence the term *autoassociative* memories. This interpretation has been generalized to store *heteroassociative* memories with mappings between different patterns, for example, the bidirectional associative memory (BAM) [12]. When noisy training patterns $\vec{S}^{\mu 1}, \vec{S}^{\mu 2}, \cdots$ are available, it is reasonable to store them by establishing mappings among them, that is [see Fig. 1(a)]

$$T_{ij} \propto \sum_{\mu} \sum_{k} \sum_{k'} (S_i^{\mu k} S_j^{\mu k'}). \tag{4}$$

This is in fact the same as

$$T_{ij} \propto \sum_{\mu} \left( \sum_{k} S_i^{\mu k} \right) \left( \sum_{k'} S_j^{\mu k'} \right). \tag{5}$$

Hence this strategy is equivalent to replacing the *clean* patterns $\vec{S}^{\mu}$ in (3) by the *average* of the noisy patterns, that is, in this strategy one first obtains a set of *approximate clean patterns* by averaging over (a form of preprocessing) the noisy patterns and then use these *approximate clean patterns* in learning with the original rule (3).

Let us now discuss another possible learning strategy in the presence of noisy training patterns. The algorithm shown in (3) is in the spirit of the Hebbian rule [8], which in essence states that the increment of a synaptic weight during presentations of training patterns is proportional to the *simultaneous activities* of the two neurons involved, i.e.,

$$\Delta T_{ij} \propto S_i S_j. \tag{6}$$

Equation (3) can be obtained from (6) if all (clean) training patterns are presented to the network consequtively during learning. When the noisy training patterns are presented to the network, one training pattern at a given instance in time, learning may occur as follows, again *in the spirit of the above Hebbian rule*:

$$T_{ij} \propto \sum_{\mu} \sum_{k} S_i^{\mu k} S_j^{\mu k}. \tag{7}$$

Since the second learning strategy discussed above is more directly related to the Hebbian learning, we choose to use it in this paper (a comparison between the above two learning strategies may be the subject of future studies)

$$T_{ij} = \frac{1}{qN} \sum_{\mu=1}^{p} \sum_{k=1}^{q} S_i^{\mu k} S_j^{\mu k} \tag{8}$$

where $\{\vec{S}^{\mu k} | \mu = 1, 2, \cdots, p; k = 1, 2, \cdots, q\}$ are the noisy training patterns and $q$ noisy patterns are used to store each standard memory. Equation (8) reduces to the original prescription (3) if $\vec{S}^{\mu k} = \vec{S}^{\mu}$ for all $k$ and $\mu$.

For our analysis in this paper, we choose the following form of training noise:

$$S_j^{\mu k} = S_j^{\mu} + \delta_j^{\mu k} \tag{9}$$

where $\delta_j^{\mu k}$ is the difference between the training pattern and the standard pattern, and may take values $0, \pm 2$, since both $S_i^{\mu}$ and $S_i^{\mu k}$ can be $\pm 1$ only. We assume that these differences are independent random numbers with a zero average and a standard deviation $\delta$, i.e.,

$$\langle \delta_j^{\mu k} \rangle = \frac{1}{p} \sum_{\mu=1}^{p} \delta_j^{\mu k} = \frac{1}{N} \sum_{j=1}^{N} \delta_j^{\mu k} = 0 \tag{10}$$

and similarly

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

For instance, $\delta^2 = 0.5$ indicates that $\delta^2/4 = 12.5\%$ of the bits in $\{\vec{S}^{\mu k}\}$ are randomly chosen and flipped, since each bit flipped gives $(\delta_i^{\mu k})^2 = (\pm 2)^2 = 4$. We have used "$\langle \cdot \rangle$" to indicate a statistical average, which may be carried out over the stored patterns and the neurons (10), as well as the signal transmission noise (12) below.

According to (2), when averaged over the signal transmission noise [7]

$$\begin{aligned} \langle S_i \rangle &= \mathrm{Prob}(S_i = +1)(+1) + \mathrm{Prob}(S_i = -1)(-1) \\ &= \tanh(\beta h_i^o). \end{aligned} \tag{12}$$

Averages over the stored patterns and the neurons will be carried out later.

In the mean field theory [7], we replace $h_i^o$ in (1) by its average over signal transmission noise and combine (1) with (12)

$$\langle S_i \rangle = \tanh \left( \beta \sum_{j \neq i} T_{ij} \langle S_j \rangle \right). \tag{13}$$

To solve (13), let us consider the overlap between the average state of the network and a memory pattern $\vec{S}^{\nu}$

$$m^{\nu} = \frac{1}{N} \langle \vec{S} \rangle \cdot \vec{S}^{\nu} = \frac{1}{N} \sum_{i} S_i^{\nu} \langle S_i \rangle. \tag{14}$$

Substituting (13), (8), and (9) into (14), we obtain

$$\begin{aligned} m^\nu &= \frac{1}{N}\sum_i S_i^\nu \tanh\left[\beta\,\frac{1}{qN}\right. \\ &\qquad \left.\cdot \sum_{\mu,k,j\neq i} S_i^{\mu k}\,(S_j^\mu+\delta_j^{\mu k})\langle S_j\rangle\right] \\ &= \frac{1}{N}\sum_i S_i^\nu \tanh\left[\beta\left(\frac{1}{q}\sum_{\mu k} S_i^{\mu,k} m^\mu\right.\right. \\ &\qquad \left.\left.+\,\frac{1}{qN}\sum_{\mu,k,j\neq i} S_i^{\mu k}\delta_j^{\mu k}\langle S_j\rangle\right)\right] \\ &= \frac{1}{N}\sum_i S_i^\nu \tanh\left\{\beta\left[\sum_\mu (S_i^\mu+\delta_i^\mu)m^\mu+\eta_i\right]\right\} \end{aligned} \tag{15}$$

where

$$\delta_i^\mu \equiv \frac{1}{q}\sum_k \delta_i^{\mu k} \tag{16}$$

and

$$\eta_i \equiv \frac{1}{qN}\sum_{\mu,k,j\neq i} S_i^{\mu k}\,\delta_j^{\mu k}\langle S_j\rangle. \tag{17}$$

Suppose the network is initially close to pattern $\vec{S}^1$. We consider the retrieval of this pattern $\vec{S}^1$ and evaluate

$$m \equiv m^1. \tag{18}$$

Let us first rewrite (15) slightly, using the fact that $S_i^1 = \{-1,+1\}$ and $\tanh(x)$ is an odd function

$$\begin{aligned} m^\nu &= \frac{1}{N}\sum_i S_i^\nu\, S_i^1 S_i^1 \\ &\qquad \cdot\tanh\left\{\beta\left[\sum_\mu (S_i^\mu+\delta_i^\mu)m^\mu+\eta_i\right]\right\} \\ &= \frac{1}{N}\sum_i S_i^\nu\, S_i^1 \\ &\qquad \cdot\tanh\left\{\beta\left[\sum_\mu (S_i^\mu+\delta_i^\mu)S_i^1 m^\mu+S_i^1\eta_i\right]\right\}. \end{aligned} \tag{19}$$

Then

$$m = \frac{1}{N}\sum_i \tanh[\beta(m+\Delta_i)] \tag{20}$$

where

$$\Delta_i \equiv \delta_i^1 S_i^1 m + \sum_{\mu\neq 1}(S_i^\mu+\delta_i^\mu)S_i^1 m^\mu + S_i^1\eta_i. \tag{21}$$

We now investigate the property of $\Delta_i$. We assume that [7] $\{m^\mu, \mu\neq 1\}, \{S_i^\mu\}$, and $\{\delta_i^{\mu\nu}\}$ are all *independent* random variables with mean zero, that is,

$$\langle m^\mu\rangle = 0, \qquad \text{for all } \mu\neq 1 \tag{22}$$

and

$$\langle S_i^\mu\rangle = 0, \qquad \text{for all } \mu, i \tag{23}$$

in addition to (10). Hence when averaged over the stored patterns and the neurons, we have

$$\begin{aligned} \langle\Delta_i\rangle &= \langle\delta_i^1 S_i^1 m\rangle + \sum_{\mu\neq 1}\langle(S_i^\mu+\delta_i^\mu)S_i^1 m^\mu\rangle + \langle S_i^1\eta_i\rangle \\ &= \langle\delta_i^1\rangle\langle S_i^1 m\rangle + \sum_{\mu\neq 1}\langle(S_i^\mu+\delta_i^\mu)S_i^1\rangle\langle m^\mu\rangle \\ &\quad + \langle S_i^1\rangle\langle\eta_i\rangle \\ &= 0. \end{aligned} \tag{24}$$

Hence the first term $m$ in the right-hand side of (20) may be regarded as the *signal* term, which drives the system toward the memory state $\vec{S}^1$, whereas the second term $\Delta_i$ is a *noise* that interferes with the converging process.

Let us now evaluate the standard deviation of this noise term $\langle\Delta_i^2\rangle$: all cross-terms between the terms on the right-hand side of (21) vanish after average, because of the independence among the random variables and their zero-averages, and thus only the squared terms survive, i.e.,

$$\begin{aligned} v^2 &\equiv \langle\Delta_i^2\rangle \\ &= \langle(\delta_i^1)^2 m^2\rangle + \sum_{\mu\neq 1}\langle(m^\mu)^2\rangle \\ &\qquad + \sum_{\mu\neq 1}\langle(\delta_i^\mu m^\mu)^2\rangle + \langle(\eta_i)^2\rangle. \end{aligned} \tag{25}$$

Combining (11) and (16), we obtain

$$\langle(\delta_i^\mu)^2\rangle = \frac{1}{q^2}\sum_k\langle(\delta_i^{\mu k})^2\rangle = \delta^2/q \equiv \delta_q^2. \tag{26}$$

Similary, according to (17)

$$\begin{aligned} \langle(\eta_i)^2\rangle &= \frac{1}{(qN)^2}\sum_{\mu,k,j\neq i}\langle(S_i^{\mu k})^2\rangle\langle(\delta_j^{\mu k}\langle S_j\rangle)^2\rangle \\ &\approx \frac{\delta^2 p(N-1)}{qN^2} \approx \delta_q^2\alpha \end{aligned} \tag{27}$$

where we have assumed that

$$\langle(\delta_j^{\mu k}\langle S_j\rangle)^2\rangle \approx \delta^2 \tag{28}$$

and both $p$ and $N$ approach $+\infty$; however

$$\alpha \equiv \frac{p}{N} \tag{29}$$

remains a finite constant. Since random variables $\{m^\mu, \mu\neq 1\}$ should all have the same variance [7], we have, following (25):

$$\begin{aligned} v^2 &= \delta_q^2 m^2 + r\alpha + \delta_q^2 r\alpha + \delta_q^2\alpha \\ &= r\alpha + \delta_q^2(m^2+\alpha+r\alpha) \end{aligned} \tag{30}$$

where

$$\begin{aligned} r &\equiv \frac{1}{\alpha}\sum_{\mu\neq 1}\langle(m^\mu)^2\rangle = \frac{1}{\alpha}\,p\langle(m^\mu)^2\rangle \\ &= N\langle(m^\mu)^2\rangle, \qquad \mu\neq 1. \end{aligned} \tag{31}$$

We have also assumed that among all overlaps $m^\mu$ $(\mu = 1,2,\cdots,N)$, only the overlap with the attracting pattern $m\equiv m^1$ is on the order of one, and all other overlaps $m^\mu$ $(\mu\neq 1)$

are infinitesimal. In fact, as we will shown below, $m^\mu$ $(\mu \neq 1)$ are on the order of $1/\sqrt{N}$ and $r$ given in (31) is finite.

Again due to the independence among the random variables $\{m^\mu, \mu \neq 1\}, \{S_i^\mu\}$, and $\{\delta_i^{\mu\nu}\}$ and their zero-averages, we have

$$\langle \Delta_i \Delta_j \rangle = 0, \qquad \text{for } i \neq j. \tag{32}$$

We have thus established that $\{\Delta_i\}$ in (20) is a set of *independent* random variables with a zero-average and a standard deviation $v$ given by (30). According to *the central limit theorem* [6], we can replace $N^{-1}\ \Sigma_i$ in (20) with an average over a *Gaussian* noise with a zero-average and a standard deviation $v$ given by (30)

$$m = \int \frac{dz}{\sqrt{2\pi}}\, e^{-z^2/2} \tanh[\beta(m + vz)]. \tag{33}$$

The form of (33) is identical to the corresponding equation derived for the case without training noise by Amit *et al.* [4] and the effect of training noise is included in the standard deviation given by (30). That is, by letting $\delta_q^2 = 0$, (33) and (30) reduce to those of Amit *et al.* [4].

Since $v$ in (33) depends on $r$, we need to evaluate $r$ self-consistently by starting from (19) for $\nu \neq 1$

$$m^\nu = \frac{1}{N} \sum_i S_i^\nu S_i^1 \cdot \tanh\{\beta[m + (S_i^\nu + \delta_i^\nu) S_i^1 m^\nu + \zeta_i]\} \tag{34}$$

where

$$\zeta_i \equiv \delta_i^1 S_i^1 m + \sum_{\mu \neq 1, \nu} (S_i^\mu + \delta_i^\mu) S_i^1 m^\mu + S_i^1 \eta_i. \tag{35}$$

Following exactly the same analysis for $\Delta_i$ defined in (21), we can show that

$$\langle \zeta_i \rangle = \langle \Delta_i \rangle = 0 \tag{36}$$

and

$$\langle (\zeta_i)^2 \rangle = \langle (\Delta_i)^2 \rangle = v^2 = r\alpha + \delta_q^2 (m^2 + \alpha + r\alpha). \tag{37}$$

Another way to show (36) and (37) is by observing that

$$\zeta_i = \Delta_i - (S_i^\nu + \delta_i^\nu) S_i^1 m^\nu \tag{38}$$

and $m^\nu$ with $\nu \neq 1$ is negligible by itself.

Expanding the second term on the right-hand side of (34), which is proportional to the small quantity $m^\nu$, we obtain

$$\begin{aligned} m^\nu &= \frac{1}{N} \sum_i S_i^\nu S_i^1 \tanh[\beta(m + \zeta_i)] \\ &\quad + \frac{\beta}{N} \sum_i \{1 - \tanh^2[\beta(m + \zeta_i)]\} \\ &\quad \cdot (1 + \delta_i^\nu S_i^\nu) m^\nu \\ &= \frac{1}{N} \sum_i S_i^\nu S_i^1 \tanh[\beta(m + \zeta_i)] \\ &\quad + [\beta(1 - g) + x] m^\nu \end{aligned} \tag{39}$$

where

$$g \equiv \int \frac{dz}{\sqrt{2\pi}}\, e^{-z^2/2} \tanh^2 [\beta(m + vz)] \tag{40}$$

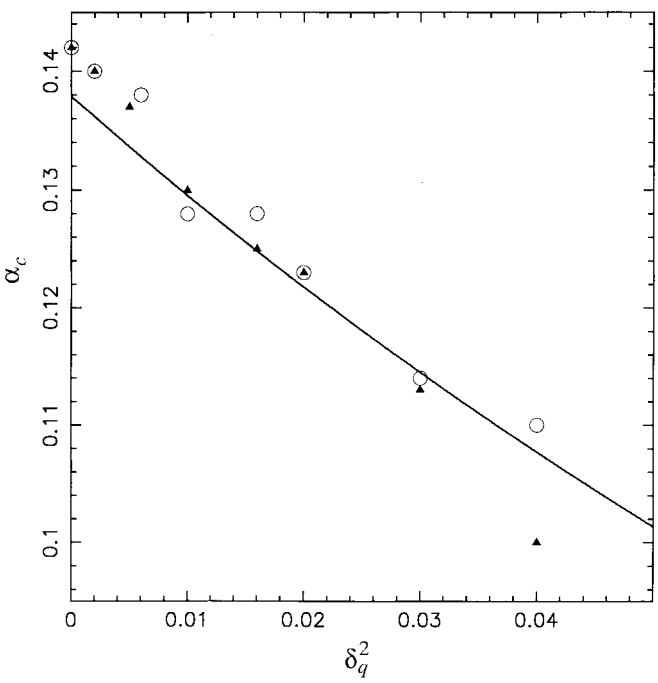


Fig. 2. The ratio $\alpha_c$ between the memory capacity $p_c$ and the number of neurons $N$, i.e., $\alpha_c = p_c/N$, for the fully connected binary Hopfield neural network, as a function of $\delta_q^2 \equiv \delta^2/q$, where $\delta$ is the standard deviation of training noise as defined in (9) and $q$ is the number of noisy training patterns used to stored each memory pattern. Solid line: the mean field theory (solutions of (48), (49), (51), and (30). Circles: simulations with $q = 5$. Triangles: simulations with $q = 10$.

and

$$x \equiv \frac{\beta}{N} \sum_i \{1 - \tanh^2[\beta(m + \zeta_i)]\} \delta_i^\nu S_i^\nu. \tag{41}$$

In (40) we have again used the central limit theorem [6] to replace the sum by an integration over a Gaussian distribution.

We now show that the $x$ in (39) is much smaller compared to the preceding term in (39) and can therefore be neglected. According to (41)

$$\begin{aligned} x^2 = \left(\frac{\beta}{N}\right)^2 \sum_{ij} & \delta_i^\nu S_i^\nu \delta_j^\nu S_j^\nu \{1 - \tanh^2[\beta(m + \zeta_i)]\} \\ & \times \{1 - \tanh^2[\beta(m + \zeta_j)]\}. \end{aligned} \tag{42}$$

Now we calculate the statistical average of $x^2$. Let us carry out the average over $\vec{S}^\nu$ first and all terms with $i \neq j$ vanish after this average. Hence

$$\begin{aligned} \langle x^2 \rangle &= \frac{1}{N} \beta^2 \delta_q^2 \left\langle \frac{1}{N} \sum_i \{1 - \tanh^2[\beta(m + \zeta_i)]\}^2 \right\rangle \\ &= \frac{1}{N} \beta^2 \delta_q^2 \int \frac{dz}{\sqrt{2\pi}}\, e^{-z^2/2} \{1 - \tanh^2[\beta(m + vz)]\}^2 \\ &\to 0, \quad \text{as } N \to +\infty. \end{aligned} \tag{43}$$

Solving (39) (with $x$ neglected) for $m^\nu$, we obtain

$$m^\nu = \frac{1}{1 - \beta(1 - g)} \frac{1}{N} \sum_i S_i^\nu S_i^1 \tanh[\beta(m + \zeta_i)] \tag{44}$$

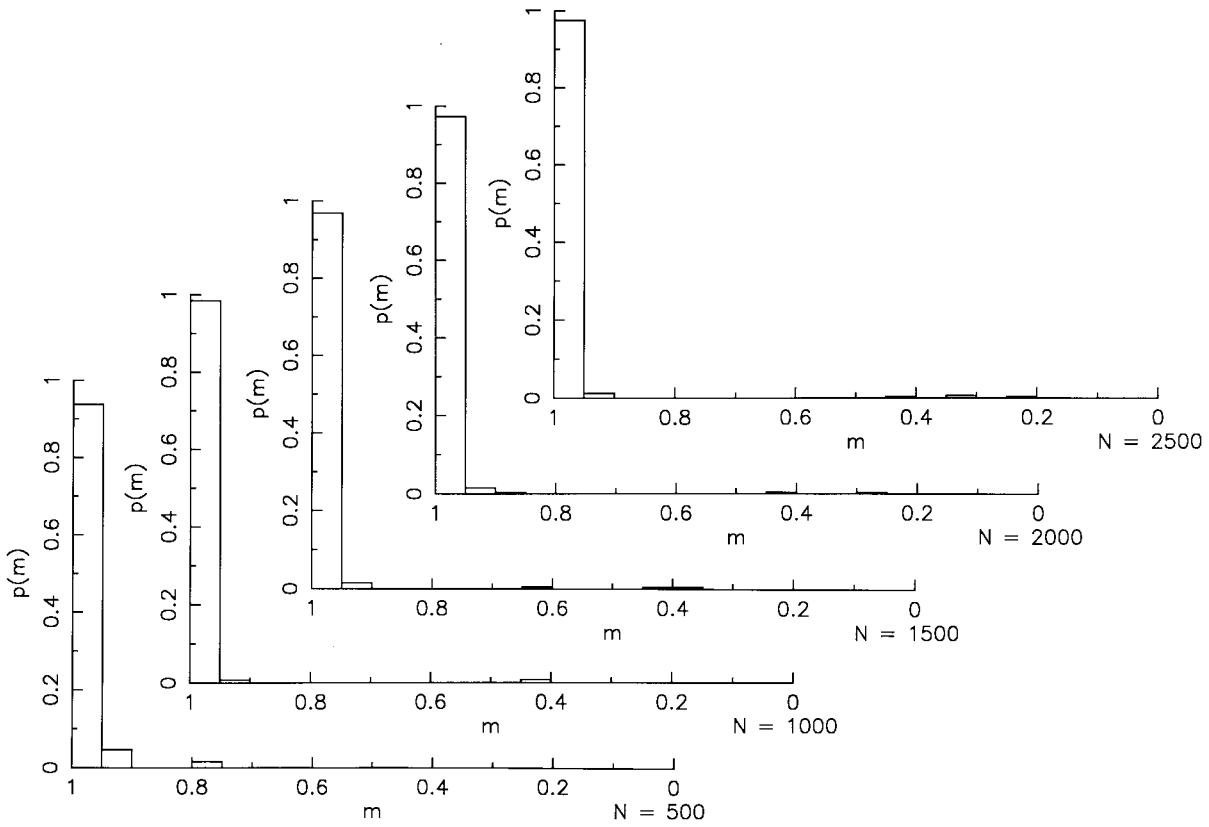


Fig. 3. Histograms of the overlaps of the retrieved state with the initial state of the network (one of the stored patterns), for $\alpha = 0.13$ and various numbers of neurons in the network $N$.

or

$$(m^\nu)^2 = \frac{1}{[1-\beta(1-g)]^2}\,\frac{1}{N^2} \cdot \sum_{ij} S_i^\nu S_i^1 S_j^\nu S_j^1 \tanh[\beta(m+\zeta_i)] \cdot \times \tanh[\beta(m+\zeta_j)]. \quad (45)$$

Now we average (45) over the stored patterns. The average over $\vec{S}^\nu$ again eliminates all terms with $i \neq j$. The average over the remaining patterns gives a factor of $g$ as in (40). Hence (45) becomes, in combination with (31)

$$r = \frac{g}{[1-\beta(1-g)]^2}. \quad (46)$$

In the absence of signal transmission noise [$\beta = +\infty$ in (2)], we take the limit $\beta \to +\infty$ in (33), (46), and (40). First, we notice, according to (40)

$$\begin{aligned}\beta(1-g) &= \beta \int \frac{dz}{\sqrt{2\pi}}\, e^{-z^2/2}(1-\tanh^2[\beta(m+vz)]) \\ &\approx \frac{\beta}{\sqrt{2\pi}}\, e^{-z^2/2}|_{m+vz=0} \int dz \cdot (1-\tanh^2[\beta(m+vz)]) \\ &= \frac{1}{\sqrt{2\pi}v}\, e^{-m^2/2v^2} \int dz\, \frac{\partial}{\partial z} \tanh[\beta(m+vz)].\end{aligned} \quad (47)$$

To obtain the second line of (47), we have observed that as $\beta \to +\infty$, the integrand in the first line of (47) vanishes for all $z$ except the close vicinity of $m+vz=0$ or $z=-m/v$. The smooth part of the integrand, i.e., the exponential factor, is approximately a constant in this small region of $z$ and can therefore be moved outside of the integration. Hence (40) reduces to

$$C \equiv \beta(1-g) = \sqrt{\frac{2}{\pi}}\,\frac{1}{v}\, e^{-m^2/2v^2}. \quad (48)$$

Since (48) shows $g \to 1$ as $\beta \to +\infty$, (46) becomes

$$r = \frac{1}{(1-C)^2}. \quad (49)$$

In addition, as $\beta \to +\infty$

$$\begin{aligned}\int \frac{dz}{\sqrt{2\pi}}\, e^{-z^2/2} \tanh[\beta(m+vz)] &\longrightarrow \int \frac{dz}{\sqrt{2\pi}}\, e^{-z^2/2}\mathrm{sgn}(m+vz) \\ &= 2\int_{-m/v}^{\infty} \frac{dz}{\sqrt{2\pi}}\, e^{-z^2/2} - 1.\end{aligned} \quad (50)$$

Hence (33) is simply

$$m = \mathrm{erf}\left(\frac{m}{\sqrt{2}v}\right) \quad (51)$$

where

$$\mathrm{erf}(x) \equiv \frac{2}{\sqrt{\pi}} \int_0^x dz\, e^{-z^2} \quad (52)$$

is the standard error function.

The memory capacity of the network trained with noisy patterns can be obtained by solving (48), (49), (51), and (30) collectively. Before we proceed to find the solutions, we note

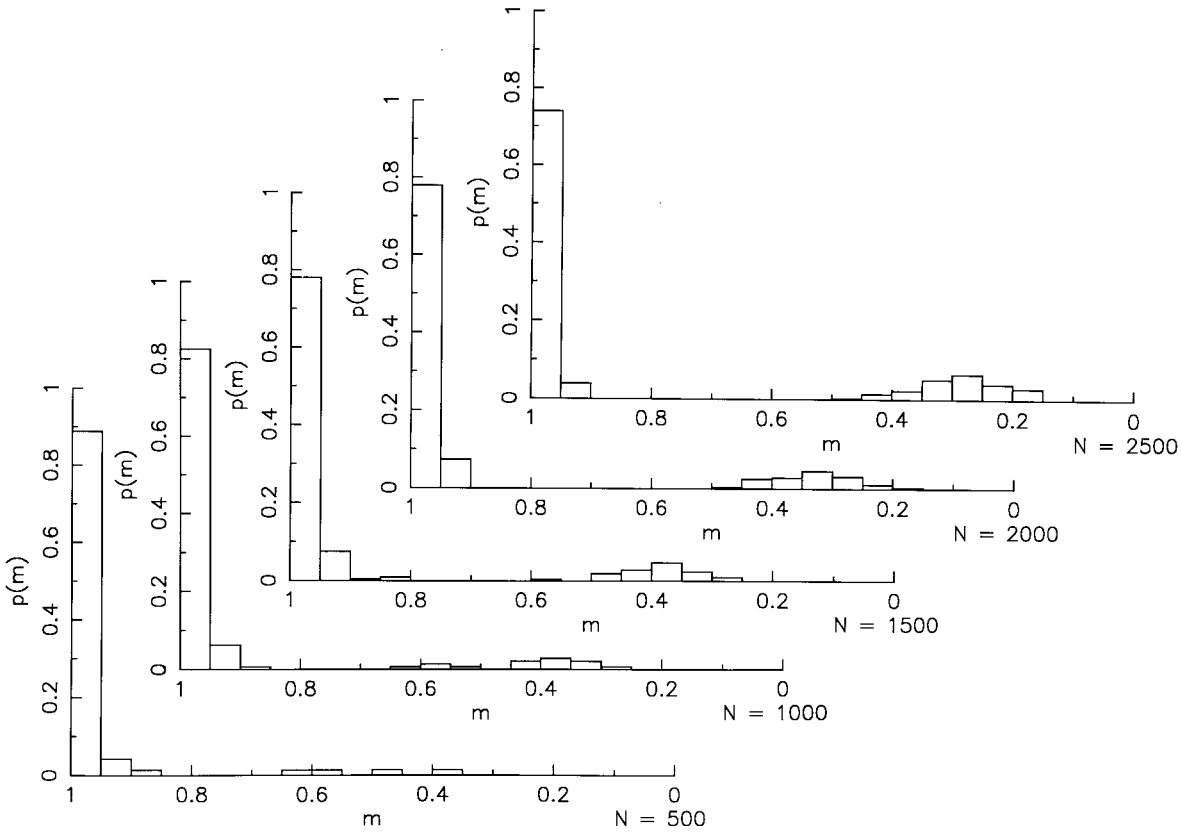


Fig. 4. Same as Fig. 3 for $\alpha = 0.143$.

that for the special case of zero noise in training patterns, i.e., $\delta = 0$, these equations reduce to the equations derived by Amit *et al.* with a replica method [4]. The procedure of solving (48), (49), (51), and (30) is presented in the Appendix. The memory capacity of the network, according to the mean field theory, is presented in Fig. 2, which shows that the capacity decreases monotonously as the noise in the training patterns increases. In the absence of training noise, i.e., $\delta_q^2 = 0$, we have $\alpha_c = 0.138$, which is the same as the result in [4]. Here

$$\alpha_c \equiv \frac{p_c}{N} \tag{53}$$

and $p_c$ is the maximum (critical) number of stored patterns at which the autoassociative memory breaks down for a network of $N$ neurons (we have assumed in this paper that $p_c \to +\infty$ and $N \to +\infty$).

## III. Computer Simulations

Fully connected binary Hopfield networks trained with noisy patterns are implemented according to (1), (8), and (9) as follows. For each $N$ (the number of neurons in the network) and $\alpha, p = \alpha N$ "clean" patterns $\{\vec{S}^\mu | \mu = 1, 2, \cdots p\}$ are formed by choosing each bit randomly from $+1$ and $-1$. For a given set of $\delta^2$ and $q, q$ noisy training patterns are formed for each "clean" pattern by flipping each bit in the "clean" pattern with a probability $\delta^2/4$. The synaptic weights are then calculated with (8). The stability of each stored pattern is then checked: the initial state of the network is set to be each of the patterns $\vec{S}^\mu$ $(\mu = 1, 2, \cdots, p)$ and the neurons are updated sequentially (asynchronously) until the network stabilizes. The updating rule is deterministic, i.e., $\beta = +\infty$ in (2)

$$\begin{aligned} S_i(t + \Delta t) &= -1, \qquad \text{if } h_i(t) \le 0 \\ &= +1, \qquad \text{if } h_i(t) > 0. \end{aligned} \tag{54}$$

The distribution of the overlaps between the final network states and the initial memory states are recorded. For each set of $N, \alpha, q$, and $\delta^2$, the above processes, including network creation and memory stability checking, are repeated four times and average results are obtained.

Let us first consider the case without training noise ($\delta_q^2 \equiv \delta^2/q = 0$). We choose $q = 1$, since the results are independent of $q$ in the absence of training noise, according to (8). Fig. 3 shows the histograms for $\alpha = 0.13$ with various values of $N$ [4]. As $N$ increases, these histograms do not change qualitatively: there are sharp peaks near $m = 1$ and very small peaks scattered around lower $m$ values. As $\alpha$ is increased with an increment of 0.001 in each round of simulation, the histograms do not change appreciably until $\alpha = 0.143$. Fig. 4 shows the histograms for $\alpha = 0.143$. As $N$ increases, the sharp peaks near $m = 1$ gradually shrink and the wider peaks at lower $m$ values start to form and grow. It can thus be extrapolated that the network never stabilizes at $m = 1$ for $N = +\infty$ and $\alpha = 0.143 \equiv \alpha_s$. Thus the autoassociative memories of a fully connected Hopfield network breaks down at $\alpha_s = 0.143$. We let the memory capacity $\alpha_c = \alpha_s - 0.001 = 0.142$ (with a confidence range of $\pm 0.001$), in the absence of training noise ($\delta_q^2 = 0$). We note that this simulation result with $\alpha_c = 0.142$ is slightly higher than the theoretical result with $\alpha_c = 0.138$.

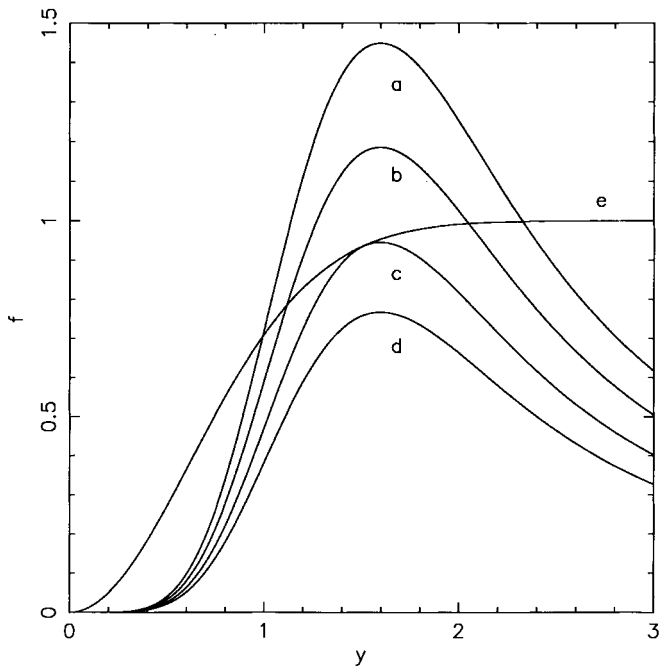


Fig. 5. $f_1(y)$ and $f_2(y)$ in (A.5) in the absence of training noise ($\delta_q^2 = 0$) and for various choices of $\alpha$. (a)–(d): $f_1(y)$. (e): $f_2(y)$. (a): $\alpha = 0.09$. (b): $\alpha = 0.11$. (c): $\alpha = 0.138$. (d): $\alpha = 0.17$.

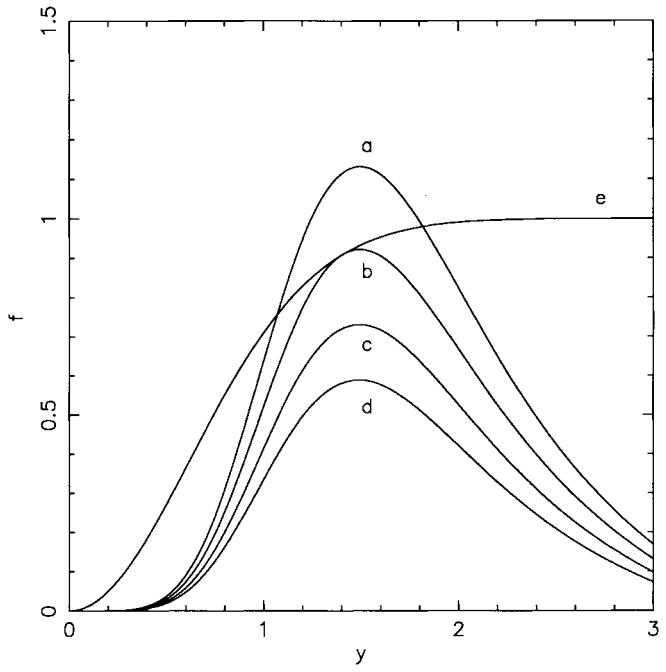


Fig. 6. Same as Fig. 5 for nonzero training noise $\delta_q^2 = 0.0365$.

In the presence of training noise, i.e., $\delta_q^2 \neq 0$, we have run simulations with both $q = 5$ and $q = 10$. The overall behaviors are similar to those when $\delta_q^2 = 0$, i.e., the $m = 1$ peaks start to shrink as $N$ increases if $\alpha$ values are increased to be sufficiently large. The method of determining $\alpha_c$ at a given $\delta_q^2 \neq 0$ is thus as follows. Starting from low $\alpha$ values and gradually increasing $\alpha$, we look for the $\alpha_s$ values at which the sharp peaks near $m = 1$ start to shrink as $N$ increases, and we let the memory capacities $\alpha_c = \alpha_s - 0.001$ (with a confidence range of $\pm 0.001$).

The memory capacities extracted from these extensive simulations are presented in Fig. 2. The quantitative agreement between the theory and the simulations is reasonably good. The small deviations between the mean field and simulation results may be attributed to the mean field theory itself, i.e., the fundamental assumption made in (13) is not rigorous.

As indicated at the beginning of this section, for each set of $N$, $\alpha, q$, and $\delta^2$, only four simulations are run. This is due to the enormous computational time required to simulate large networks operating near or above memory capacity. For example, for $N = 2500, \alpha = 0.143, \delta^2 = 0$, and $q = 1$ (no noise in training patterns), it takes *seven days* to run *one simulation* in a SUN SPARC 20, despite all synaptic weights and training patterns were stored in RAM rather than written to disks, so as to maximize the computational speed.

In the absence of training noise, i.e., $\delta_q^2 = 0$, Amit *et al.* [4] assumed that

$$P = A \exp\left[B(\alpha - \alpha_c)N\right] \tag{55}$$

where $P$ is the area under the peak near $m = 1$, and $A$ and $B$ are constants. By fitting (55), Amit *et al.* [4] obtained $\alpha_c = 0.145 \pm 0.01$. We did not use this method, since the error range of 0.01 obtained with this method is too large. A possible cause of this large error range may be the validity of (55) itself, whose theoretic origin has not been demonstrated. In fact, (55) predicts an exponential growth of the area under the $m = 1$ peak for $\alpha < \alpha_c$ as $N$ increases, but we did not observe such exponential growth in our simulations.

## IV. Summary and Discussions

In summary, the Hebbian learning rule used in Hopfield's original work is generalized to allow for the existence of noise in training patterns and the memory capacity of the fully connected binary Hopfield network is discussed analytically. Both theoretical and simulation results show that the memory capacity of the fully connected binary Hopfield network decreases as the amount of training noise increases. To achieve an even better quantitative agreement between theoretic and simulation results, a more rigorous theoretic approach is needed. The replica method used in [4] is much more complicated mathematically and yet yields the same result as the mean field method [7], [15] used in the present paper, at least in the absence of training noise. The inclusion of the so-called replica symmetry breaking may be helpful [4]; however, this is out of the scope of the present paper.

## Appendix
## Solving (48), (49), (51), and (30)

To solve these equations collectively, we need to cast them into a form with only one variable. Let

$$y = \frac{m}{\sqrt{2}v}. \tag{A.1}$$

Then (51) leads to

$$m = \operatorname{erf}(y) \tag{A.2}$$

and

$$v = \frac{1}{\sqrt{2}y} \operatorname{erf}(y). \tag{A.3}$$

Substituting (49) in (48), we have

$$r = \frac{1}{\left(1 - \sqrt{\frac{2}{\pi}}\,\frac{1}{v}\,e^{-y^2}\right)^2}. \quad \text{(A.4)}$$

We have now rewritten $m, v$, and $r$ in terms of $y$. Substituting (A.2)–(A.4) in (30), we obtain, after some manipulations

$$\frac{1}{2\alpha(1+\delta_q^2)}\left\{\left[\frac{\operatorname{erf}(y)}{y}\right]^2 - 2\delta_q^2[\operatorname{erf}^2(y)+\alpha]\right\} \cdot\left[\operatorname{erf}(y) - \frac{2}{\sqrt{\pi}}\,ye^{-y^2}\right]^2 = \operatorname{erf}^2(y). \quad \text{(A.5)}$$

We denote the left-hand side and the right-hand side of (A.5) by $f_1(y)$ and $f_2(y)$, respectively. We plot these two functions for various choices of $\alpha$ and $\delta_q^2$ in Figs. 5 and 6.

Let us first consider $\delta_q^2 = 0$. If $\alpha < 0.138$, there are two positive intersecting points between $f_1(y)$ and $f_2(y)$, which correspond to positive solutions for $y$ and $m$, the larger solution measuring the retrieval quality (see [4] for discussions on the meaning of the smaller solution). If $\alpha > 0.138$, there are no positive intersecting points between $f_1(y)$ and $f_2(y)$, which represents the breakdown of the autoassociative memories. At the critical memory capacity $\alpha_c(\delta_q^2 = 0) = 0.138$, there is one positive intersecting (tangent) point between $f_1(y)$ and $f_2(y)$.

As $\delta_q^2$ increases from zero, $f_2(y)$ does not change; however, $f_1(y)$ curves for various choices of $\alpha$ move downwards. For example, when $\delta_q^2 = 0.0365$, the $f_1(y)$ curve with $\alpha = 0.11$ becomes tangent to $f_2(y)$. Thus $\alpha_c(\delta_q^2 = 0.0365) = 0.11 < \alpha_c(\delta_q^2 = 0) = 0.138$. In general, the memory capacity $\alpha_c(\delta_q^2)$ is a decreasing function of the noise in training patterns.

## ACKNOWLEDGMENT

The author thanks the reviewers for many helpful comments and suggestions.

## REFERENCES

[1] S. Abe, "Global convergence and suppression of spurious states of the Hopfield neural networks," *IEEE Trans. Circuits Syst.—II*, vol. 40, pp. 246–257, Apr. 1993.
[2] M. Abelles, *Local Cortical Circuits.* New York: Springer-Verlag, 1982, p. 21.
[3] S.-I. Amari, "Learning patters and pattern sequences by self-organizing nets of threshold elements," *IEEE Trans. Comput.*, vol. 21, p. 1197, 1972.
[4] D. J. Amit, H. Gutfreund, and H. Sompolinsky, "Statistical mechanics of neural networks near saturation," *Ann. Phys.*, vol. 173, pp. 30–67, 1987.
[5] B. Derrida, E. Gardner, and A. Zippelius, "An exactly solvable asymmetric neural-network model," *Europhys. Lett.*, vol. 4, no. 2, pp. 167–173, July 1987.
[6] J. E. Freund, *Mathematical Statistics*, 5th ed. Englewood Cliffs, NJ: Prentice-Hall, 1992, p. 295.
[7] T. Geszti, *Physical Models of Neural Networks*. Singapore: World Scientific, 1990.
[8] D. O. Hebb, *The Organization of Behavior*. New York: Wiley, 1949, p. 62.
[9] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in *Proc. Nat. Academy Sci. USA*, vol. 79, 1982, pp. 2554–2558.
[10] ——, "Neurons with graded response have collective computational properties like those of two-state neurons," in *Proc. Nat. Academy Sci. USA*, vol. 81, May 1984, pp. 3088–3092.
[11] D. Kleinfeld, "Sequential state generation by model neural networks," in *Proc. Nat. Academy Sci. USA*, vol. 83, 1986, pp. 9469–9473.
[12] B. Kosko, "Bidirectional associative memories," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, pp. 49–60, Jan./Feb. 1988.
[13] W. A. Little, "The existence of persistent states in the brain," *Math. Biosci.*, vol. 19, pp. 101–120, 1974.
[14] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Math. Biophys.*, vol. 5, pp. 115–133, 1943.
[15] P. Peretto, "On learning rules and memory storage abilities of asymmetrical neural networks," *J. Phys.*, France, vol. 49, pp. 711–726, 1988.
[16] D. Schonfeld, "On the hysteresis and robustness of Hopfield neural networks," *IEEE Trans. Circuits Syst.—II*, vol. 40, pp. 745–748, Nov. 1993.
[17] G. L. Shaw and R. Vasudevan, "Persistent states of neural networks and the random nature of synaptic transmission," *Math. Biosci.*, vol. 21, pp. 207–217, 1974.
[18] H. Sompolinsky and I. Kanter, "Temporal association in asymmetric neural networks," *Phys. Rev. Lett.*, vol. 57, pp. 2861–2864, 1986.
[19] L. Wang, "Processing spatio-temporal sequences with any static associative neural network," *IEEE Trans. Circuit Syst.—II*, vol. 5, May 1998.
[20] ——, "On the dynamics of discrete-time, continuous-state Hopfield neural networks," *IEEE Trans. Circuits Syst.—II: Analog and Digital Signal Processing*, vol. 5, May 1998.
[21] L. Wang, "Noise injection into inputs in sparsely connected Hopfield and winner-take-all neural networks," *IEEE Trans. Syst., Man, Cybern.*, vol. 27, pp. 868–870, October 1997.
[22] L. Wang, "Discrete-time convergence theory and updating rules for neural networks with energy functions," *IEEE Trans. Neural Networks*, vol. 8, pp. 445–447, Mar. 1997.
[23] ——, "Oscillatory and chaotic dynamics in neural networks under varying operating conditions," *IEEE Trans. Neural Networks*, vol. 7, pp. 1382–1388, Nov. 1996.
[24] ——, "Suppressing chaos with hysteresis in a higher order neural network," *IEEE Trans. Circuits Syst.—II*, vol. 43, pp. 845–846, Dec. 1996.
[25] L. Wang, E. E. Pichler, and J. Ross, "Oscillations and chaos in neural networks: an exactly solvable model," in *Proc. Nat. Academy Sci. USA*, vol. 87, Dec. 1990, pp. 9467–9471.
[26] L. Wang and J. Ross, "Synchronous neural networks of nonlinear threshold elements with hysteresis," in *Proc. Nat. Academy Sci. USA*, vol. 87, Feb. 1990, pp. 988–992.
[27] ——, "Interactions of neural networks: models for distraction and concentration," in *Proc. Nat. Academy Sci. USA*, vol. 87, Sept. 1990, pp. 7110–7114.
[28] ——, "Chaos, multiplicity, crisis, and synchronicity in higher order neural networks," *Phys. Rev. A*, vol. 44, no. 4, pp. R2259–2262, Aug. 1991.
[29] ——, "Physical modeling of neural networks," in *Methods in Neuroscience, Computers and Computations in the Neurosciences*, vol. 10, P. M. Conn, Ed. San Diego, CA: Academic, 1992, pp. 549–567.

**Lipo Wang** received the B.S. degree in laser optics from the National University of Defense Technology, Changsha, China, in 1983, and the Ph.D. degree in physics from Louisiana State University, Baton Rouge, in 1988 (with a CUSPEA assistantship).

In 1989 he worked at Stanford University, California, as a Postdoctoral Fellow. In 1990 he was a Lecturer in the University College of the University of New South Wales, Australia. From 1991 to 1994 he was on the research staff of the National Institutes of Health, Bethesda, MD. Since 1995 he has been with the computing faculty at the School of Computing and Mathematics, Deakin University, Australia, where he is a tenured computing lecturer. His research interests include theory and applications of neural networks, temporal behavior, pattern recognition, control, data mining, fuzzy networks, and biological networks. He is author or coauthor of 40 journal publications and 30 conference presentations, holds a U.S. patent on neural networks, and is coauthor/editor of *Artificial Neural Networks: Oscillations, Chaos, and Sequence Processing* (Los Alamitos, CA: IEEE Computer Soc. Press, 1993).

Dr. Wang is an Associate Editor for Knowledge and Information Systems, an international journal, and has served on the program committees of several conferences.