Fast Computation in Hamming and Hopfield Networks

Isaac Meilijson and Eytan Ruppin
School of Mathematical Sciences
Raymond and Beverly Sackler Faculty of Exact Sciences
Tel-Aviv University, 69978 Tel-Aviv, Israel
e-mail: isaco@math.tau.ac.il and ruppin@math.tau.ac.il

Moshe Sipper
Logic Systems Laboratory
Swiss Federal Institute of Technology
IN-Ecublens, CH-1015 Lausanne, Switzerland
e-mail: Moshe.Sipper@di.epfl.ch

January 9, 1997

1 General Introduction

This chapter reviews the work presented in [1, 2], concerned with the development of fast and efficient variants of the Hamming and Hopfield networks. In the first part, we analyze in detail the performance of a Hamming network, the most basic and fundamental neural network classification paradigm. We show that if the activation function of the memory neurons in the original Hamming network is replaced by a an appropriately chosen simple threshold function, the 'winner-take-all' subnet of the Hamming network (known to be the essential factor determining the time complexity of the network's computation) may be altogether discarded. Under some conditions, the resulting Threshold Hamming Network correctly classifies the input patterns in a *single iteration*, with probability approaching 1.

In the second part of this chapter, we present a methodological framework describing the two-iteration performance of Hopfield-like attractor neural networks with history-dependent, Bayesian dynamics. We show that the optimal signal (activation) function has a *slanted sigmoidal* shape, and provide an intuitive account of activation functions with a non-monotone shape. We show that even in situations where the input patterns are applied to only a small subset of the network neurons (and little information is hence conveyed to the network), optimal signaling allows for the fast convergence of the Hopfield network to the correct memory states in just two iterations.

2 Threshold Hamming Networks

2.1 Introduction

Neural networks are frequently employed as associative memories for pattern classification. The network typically classifies input patterns into one of several memory patterns it has stored, representing the various classes. A conventional measure used in the context of binary vectors is the Hamming distance, defined as the number of bits in which the pattern vectors differ. The Hamming network (HN) calculates the Hamming distance between the input pattern and each memory pattern, and selects the memory with the smallest Hamming distance, which is declared 'the winner'. This network is the most straightforward associative memory. Originally presented in [3, 4, 5], it has received renewed attention in recent years
[6, 7].

The framework we analyze is an HN storing $m+1$ memory patterns $\xi^1, \xi^2, \ldots, \xi^{m+1}$, each being an n-dimensional binary vector with entries ± 1. The $(m+1)n$ memory entries are independent with equally likely ± 1 values. The input pattern x is an n-dimensional vector of ± 1's, randomly generated as a distorted version of one of the memory patterns, (say ξ^{m+1}) such that $P(x_i = {\xi_i}^{m+1}) = \alpha$, $\alpha > 0.5$. α is the initial similarity between the input pattern and the *correct* memory pattern ξ^{m+1}.

A typical HN, sketched in figure 1, is composed of two subnets:

1. The *similarity* subnet, consisting of an n-neuron input layer and an m-neuron memory layer. Each memory layer neuron i is connected to all n input layer neurons.

2. The *winner-take-all* (WTA) subnet, consisting of a fully connected m-neuron topology.

A memory pattern ξ^i is stored in the network by letting the values of the connections between memory neuron i and the input-layer neurons j $(j = 1, \ldots, n)$ be

$$a_{ij} = {\xi_j}^i \tag{1}$$

The values of the weights W_{ij} in the WTA subnet are chosen so that for each
$i, j = 1, 2, \ldots, m+1$

$$W_{ii} = 1 \ , \quad -1/m < W_{ij} < 0 \quad \text{for } i \neq j \ . \tag{2}$$

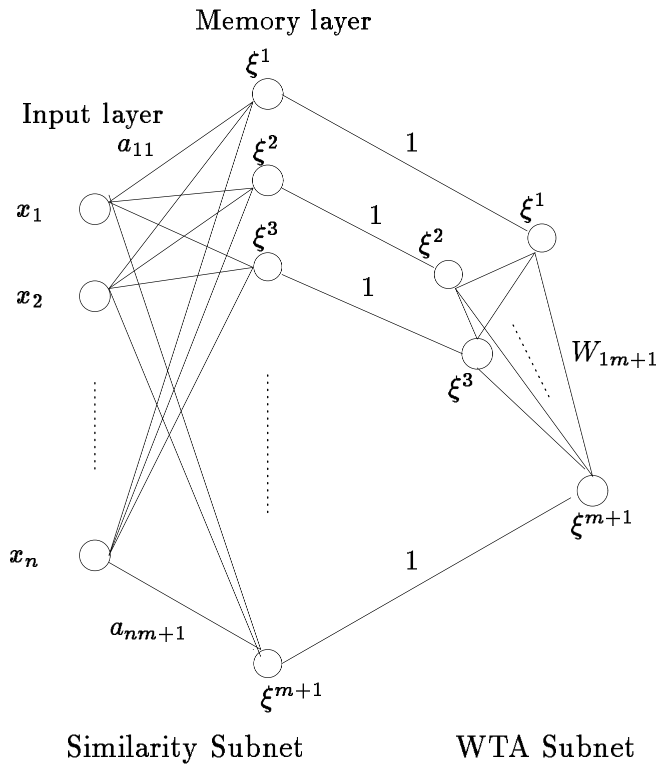

Figure 1: A Hamming net

After an input pattern x is presented on the input layer, the HN computation proceeds in two steps, each performed in a different subnet:

1. Each memory neuron i $(1 \leq i \leq m+1)$ in the similarity subnet computes its *similarity* Z_i with the input pattern

$$Z_i = \frac{1}{2}(\sum_{j=1}^{n} a_{ij} x_j + n) = \frac{1}{2}(\sum_{j=1}^{n} {\xi^i}_j x_j + n) \ . \tag{3}$$

2. Each memory-neuron i in the similarity subnet transfers its Z_i value to its duplicate in the WTA network (via a single 'identity' connection of magnitude 1). The WTA network then finds the pattern j with the maximal similarity: each neuron i in the WTA subnet sets its initial value $y_i(0) = Z_i/n$, and then computes $y_i(t)$ iteratively $(t = 1, 2, \ldots)$ by

$$y_i(t) = \Theta_0 \left(\sum_j W_{ij} y_j(t-1) \right) \tag{4}$$

where Θ_T is the threshold logic function

$$\Theta_T(u) = \begin{cases} u & \text{if } u \geq T \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

These iterations are repeated until the activity levels of the WTA neurons do not change any more, and the only memory neuron remaining active (i.e., with a positive y_i) is declared the winner. It is straightforward to see that given a winner memory neuron i, its corresponding memory pattern ξ^i can be retrieved on the input layer using the weights a_{ij}. The network's *performance* level is the probability that the winning memory will be the correct one, $m+1$.

Since the computation of the similarity subnet is performed in a single iteration, the time complexity of the network is primarily due to the time required for the convergence of the WTA subnet. In a recent paper [8], the worst-case convergence time of the standard WTA network described above was shown to be of the order of $\Theta(m \ln(mn))$ iterations. This time complexity can be very large, as simple entropy considerations show that the capacity of HNs is approximately given by

$$m \approx \sqrt{2\pi n \alpha(1-\alpha)} e^{nG(\alpha)} \tag{6}$$

where

$$G(\alpha) = \ln 2 + \alpha \ln \alpha + (1-\alpha) \ln(1-\alpha) \ . \tag{7}$$

As an example, if $\alpha = 0.7$ (70% correct entries) and $n = 400$, the memory capacity is $m \approx 10^7$, resulting in a large overall running time of the corresponding HN.

We present in this article a detailed analysis of the performance of a HN classifying distorted memory patterns. Based on our analysis, we show that it is possible to completely discard the WTA subnet by letting each memory neuron i in the similarity subnet operate the threshold logic function Θ_T on its calculated similarity Z_i. If the value of the threshold T is properly tuned, only the neuron standing for the 'correct' memory class will be activated. The resulting Threshold Hamming Network (THN) will perform correctly (with probability approaching 1) in a single iteration. Thereafter, we develop a close approximation to the error probabilities of the HN and the THN. We find the optimal threshold of the THN and compare its performance with that of the original HN.

2.2 The Threshold Hamming network

We first present some sharp approximations to the binomial distribution (proofs of these Lemmas are given in [1]).

Lemma 1.

Let $X \sim Bin(n,p)$. If x_n are integers such that $lim_{n\to\infty} \frac{x_n}{n} = \beta \in (p,1)$, then

$$P(X = x_n) \approx \frac{1}{\sqrt{2\pi n\beta(1-\beta)}} \exp\{-n[\beta \ln \frac{\beta}{p} + (1-\beta)\ln \frac{1-\beta}{1-p}]\} \tag{8}$$

and

$$P(X \geq x_n) \approx \frac{1-p}{(1-\frac{p}{\beta})\sqrt{2\pi n\beta(1-\beta)}} \exp\{-n[\beta \ln \frac{\beta}{p} + (1-\beta)\ln \frac{1-\beta}{1-p}]\} \tag{9}$$

in the sense that the ratio between LHS and RHS converges to 1 as $n \to \infty$. For the special case $p = \frac{1}{2}$, let $G(\beta) = \ln 2 + \beta \ln \beta + (1-\beta)\ln(1-\beta)$, then

$$P(X = x_n) \approx \frac{\exp\{-nG(\beta)\}}{\sqrt{2\pi n\beta(1-\beta)}} \tag{10}$$

$$P(X \geq x_n) \approx \frac{\exp\{-nG(\beta)\}}{(2-\frac{1}{\beta})\sqrt{2\pi n\beta(1-\beta)}} \tag{11}$$

The rationale for the next two lemmas will be intuitively clear interpreting X_i $(1 \leq i \leq m)$ as similarity between the initial pattern and (wrong) memory i, and Y as similarity with the correct memory $m+1$. If we use x_n as threshold, the decision will be correct if all X_i are below x_n and Y is above x_n. We will expand on this point later.

Lemma 2.

Let $X_i \sim Bin(n, \frac{1}{2})$ be independent, $\gamma \in (0,1)$, and let x_n be as in Lemma 1. If

$$m = (2 - \frac{1}{\beta})\sqrt{2\pi n\beta(1-\beta)} \left(\ln \frac{1}{\gamma}\right) e^{nG(\beta)}, \tag{12}$$

then

$$P(max(X_1, X_2, \cdots, X_m) < x_n) \approx \gamma \tag{13}$$

Lemma 3.

Let $Y \sim Bin(n, \alpha)$ with $\alpha > \frac{1}{2}$, let (X_i) and γ be as in Lemma 2, and let $\eta \in (0,1)$. Let x_n be the integer closest to $n\beta$, where

$$\beta = \alpha - \sqrt{\frac{\alpha(1-\alpha)}{n}} z_\eta - \frac{1}{2n} \tag{14}$$

and z_η is the η - quantile of the standard normal distribution, i.e.,

$$\eta = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z_\eta} e^{-x^2/2} dx \tag{15}$$

Then, if Y and (X_i) are independent

$$P\left(max(X_1, X_2, \cdots, X_m) < Y\right) \geq P(max(X_1, X_2, \cdots, X_m) < x_n \leq Y) \qquad (16)$$

and the RHS of (16) converges to $\gamma\eta$ for m as in (12) and $n \to \infty$.

Bearing these three lemmas, recall that the similarities $(Z_1, Z_2, \ldots, Z_m, Z_{m+1})$ are independent. If $Max(Z_1, Z_2, \ldots, Z_m, Z_{m+1}) = Z_j$ for a single memory neuron j, the conventional HN declares ξ^j the 'winning pattern'. Thus, the probability of error is the probability of a tie or of getting $j \neq m+1$. Let X_j be the similarity between the input vector and the $j'th$ memory pattern $(1 \leq j \leq m)$, and let Y be the similarity with the 'correct' memory pattern ξ^{m+1}. Clearly, X_j is $Bin(n, \frac{1}{2})$-distributed, and Y is $Bin(n, \alpha)$-distributed. We now propose a THN having a threshold value x_n: As in the HN, each memory neuron in the similarity subnet computes its similarity with the input pattern. But now, each memory neuron i whose similarity X_i is at least x_n declares itself 'the winner'. There is no WTA subnet. An error may arise if there is a multiplicity of memory neurons declaring themselves 'the winner', there is no winning pattern, or a wrong single winner. The threshold x_n is chosen so as to minimize the error probability.

To build a THN with probability of error not exceeding ϵ, observe that expression (13) gives the probability γ that no wrong pattern declares itself the winner, while expression (15) gives the probability η that the correct pattern $m+1$ declares itself the winner. The product of these two terms is the probability of correct decision (i.e., the performance level) of the THN, which should be at least $1-\epsilon$. Given n, ϵ and α, a THN may be constructed by simply choosing even error probabilities, i.e., $\gamma = \eta = \sqrt{1-\epsilon}$. Then, we determine β by (14), let x_n be the integer closest to $n\beta$, and determine the memory capacity m using (12). If m, ϵ and α are given, a THN may be constructed in a similar manner, since it is easy to determine n from m and ϵ by iterative procedures. Undoubtedly, the HN is superior to the THN, as explicitly shown by inequality (16). However, as we shall see, the performance loss using the THN can be recovered by a moderate increase in the network size n, while time complexity is drastically reduced by the abolition of the WTA subnet. In the next subsection we derive a more efficient choice of x_n (with uneven error probabilities), which yields a THN with optimal performance.

2.3 The Hamming Network and an Optimal Threshold Hamming Network

To find an optimal THN, we replace the ad-hoc choice of $\gamma = \eta = \sqrt{1-\epsilon}$ (among all pairs (γ, η) for which $\gamma\eta = 1 - \epsilon$) by the choice of the threshold x_n that maximizes the storage capacity $m = m(n, \epsilon, \alpha)$. We also compute the error probability $\epsilon(m, n, \alpha)$ of the HN for arbitrary m, n and α, and compare it with ϵ, the error probability of the THN.

Let ϕ (Φ) denote the standard normal density (cumulative distribution function), and let $r = \phi/(1-\Phi)$ denote the corresponding failure rate function. Then,

Lemma 4.

The optimal proportion δ between the two error probabilities satisfies

$$\delta = \frac{1-\gamma}{1-\eta} \approx \frac{r(z_\eta)}{\sqrt{n\alpha(1-\alpha)}\ln\frac{\beta}{1-\beta}} \,. \tag{17}$$

Proof:

Let $M = max(X_1, X_2, \cdots, X_m)$, and let Y denote the similarity with the 'correct' memory pattern, as before. We have seen that $P(M < x) \approx \exp\{-m\frac{\exp\{-nG(\beta)\}}{\sqrt{2\pi n\beta(1-\beta)(2-\frac{1}{\beta})}}\}$. Since $G'(\beta) = \ln\frac{\beta}{(1-\beta)}$, then by Taylor expansion

$$P(M < x) = P(M < x_0 + x - x_0) \approx \exp\{-m\frac{\exp\{-n[G(\beta + \frac{x-x_0}{n})]\}}{\sqrt{2\pi n\beta(1-\beta)}(2-\frac{1}{\beta})}\} \approx$$
$$\exp\{-m\frac{\exp\{-nG(\beta) - (x-x_0)\ln\frac{\beta}{(1-\beta)}\}}{\sqrt{2\pi n\beta(1-\beta)}(2-\frac{1}{\beta})}\} = (P(M < x_0))^{(\frac{\beta}{1-\beta})^{x_0 - x}} = \gamma^{(\frac{\beta}{1-\beta})^{x_0 - x}} \tag{18}$$

(in accordance with Gnedenko extreme-value distribution of type 1 [9]). Similarly,

$$P(Y < x) = \exp\{\ln P(Y < x_0 + x - x_0)\} = \exp\{\ln P\left(Z < \frac{x_0 - n\alpha}{\sqrt{n\alpha(1-\alpha)}} + \frac{x - x_0}{\sqrt{n\alpha(1-\alpha)}}\right)\}$$
$$\approx P(Y < x_0)\exp\{\frac{\phi(z)}{\Phi^*(z)}\frac{x - x_0}{\sqrt{n\alpha(1-\alpha)}}\} = (1-\eta)\exp\{r(z)\frac{x - x_0}{\sqrt{n\alpha(1-\alpha)}}\} \tag{19}$$

where $\Phi^* = 1 - \Phi$. The probability of correct recognition using a threshold x can now be expressed as

$$P(M < x)P(Y \geq x) \approx \gamma^{(\frac{\beta}{1-\beta})^{x_0 - x}}(1 - (1-\eta)\exp\{r(z)\frac{x - x_0}{\sqrt{n\alpha(1-\alpha)}}\}) \tag{20}$$

We differentiate expression (20) with respect to $x_0 - x$, and equate the derivative at $x = x_0$ to zero, to obtain the relation between γ and η that yields the optimal threshold, i.e., that which maximizes the probability of correct recognition. This yields

$$\gamma = \exp\{-\frac{r(z)}{\sqrt{n\alpha(1-\alpha)}\ln\frac{\beta}{1-\beta}}\frac{1-\eta}{\eta}\} \tag{21}$$

We now approximate

$$1-\gamma \approx -\ln\gamma \approx \frac{r(z)}{\sqrt{n\alpha(1-\alpha)}\ln\frac{\beta}{1-\beta}}(1-\eta) \tag{22}$$

and thus the optimal proportion between the two error probabilities is

$$\delta = \frac{1-\gamma}{1-\eta} \approx \frac{r(z)}{\sqrt{n\alpha(1-\alpha)}\ln\frac{\beta}{1-\beta}} \ . \tag{23}$$

Based on Lemma 4, if the desired probability of error is ϵ, we choose

$$\gamma = 1 - \frac{\delta\epsilon}{1+\delta}, \qquad \eta = 1 - \frac{\epsilon}{(1+\delta)} \ . \tag{24}$$

We start with $\gamma = \eta = \sqrt{1-\epsilon}$, obtain β from (14) and δ from (17), recompute η and γ from (24) and iterate. The limiting values of β and γ in this iterative process give the maximal capacity m (by 12) and threshold x_n (as the integer closest to $n\beta$).

We now compute the error probability $\epsilon(m,n,\alpha)$ of the original HN (with the WTA subnet) for arbitrary m, n and α, and compare it with ϵ.

Lemma 5.

For arbitrary n, α and ϵ, let m, β, γ, η and δ be as calculated above. Then, the probability of error $\epsilon(m,n,\alpha)$ of the HN satisfies

$$\epsilon(m,n,\alpha) \approx \Gamma(1-\delta)\frac{1-e^{-\delta\ln\frac{\beta}{1-\beta}}}{\delta\ln\frac{\beta}{1-\beta}}\frac{(\epsilon\delta)^{\delta}}{(1+\delta)^{1+\delta}}\epsilon \tag{25}$$

where

$$\Gamma(t) = \int_0^\infty x^{t-1}e^{-x}dx \tag{26}$$

is the Gamma function.

Proof:

$$P(Y \le M) = \sum_x P(Y \le x)P(M = x) = \sum_x P(Y \le x)[P(M < x+1) - P(M < x)] \approx$$
$$\sum_x P(Y \le x_0)e^{-\delta(x_0-x)\ln\frac{\beta}{1-\beta}}[(P(M < x_0))^{(\frac{\beta}{1-\beta})^{x_0-x-1}} - (P(M < x_0))^{(\frac{\beta}{1-\beta})^{x_0-x}}] \tag{27}$$

We now approximate this sum by the integral of the summand: let $b = \frac{\beta}{1-\beta}$ and $c = \delta \ln \frac{\beta}{1-\beta}$. We have seen that the probability of incorrect performance of the WTA subnet is equal to

$$P(Y \leq M) \approx \sum_x P(Y \leq x_0) e^{-c(x_0 - x)} [(P(M < x_0))^{b^{(x_0 - x - 1)}} - (P(M < x_0))^{b^{(x_0 - x)}}] \approx$$

$$(1-\eta) \int_{-\infty}^{\infty} (\gamma^{b^{y-1}} - \gamma^{b^y}) e^{-cy} dy \quad (28)$$

Now we transform variables $t = b^y \ln \frac{1}{\gamma}$ to get the integral in the form

$$e^{-c}(1-\eta) \int_0^{\infty} (e^{-t} - e^{-bt}) (\frac{t}{\ln \frac{1}{\gamma}})^{\frac{-c}{\ln b}} \frac{dt}{t \ln b} = K_1 \int_0^{\infty} (e^{-t} - e^{-bt}) t^{-(1+K_2)} dt \quad (29)$$

This is the convergent difference between two divergent Gamma function integrals. We perform integration by parts to obtain a representation as an integral with t^{-K_2} instead of $t^{-(1+K_2)}$ in the integrand. For $0 \leq K_2 < 1$, the corresponding integral converges. The final result is then

$$(1-\eta) \frac{1 - e^{-c}}{c} \Gamma(1 - \frac{c}{\ln b}) (\ln \frac{1}{\gamma})^{\frac{c}{\ln b}} \quad (30)$$

Hence, we have

$$P(Y \leq M) \approx (1-\eta) \frac{1 - e^{-\delta \ln \frac{\beta}{1-\beta}}}{\delta \ln \frac{\beta}{1-\beta}} \Gamma(1-\delta) (\ln \frac{1}{\gamma})^{\delta} \approx$$

$$\Gamma(1-\delta) \frac{1 - e^{-\delta \ln \frac{\beta}{1-\beta}}}{\delta \ln \frac{\beta}{1-\beta}} \frac{(\epsilon\delta)^{\delta}}{(1+\delta)^{1+\delta}} \epsilon \quad (31)$$

as claimed. Expression (25) is presented as $K(\epsilon, \delta, \beta) \cdot \epsilon$, where $K(\epsilon, \delta, \beta)$ is the factor (≤ 1) by which the probability of error ϵ of the THN should be multiplied in order to get the probability of error of the original HN with the WTA subnet. For small δ, K is close to 1. However, as will be seen in the next subsection, K is typically smaller.

2.4 Numerical results

We examined the performance of the HN and the THN via simulations (of 10000 runs each), and compared their error rates with those expected in accordance with our calculations. Due to its probabilistic characterization, the THN may perform reasonably only above some minimal size of n (depending on α and m). The results for such a 'minimal' network, indicating the percent of errors at various m values, are presented in table 1. As evident, the experimental results corroborate the accuracy of the THN and HN calculations already at this relatively small network storing a very small number of memories in relation to its capacity. The performance of the THN is considerably worse than that of the corresponding

HN. However, as shown in table 2, an increase of 50% in the input layer size n yields a THN which performs about as well as the previous HN.

m	100	200	400	800	1600	3200
(Threshold)	(99)	(100)	(100)	(101)	(102)	(102)
HN: predicted	0.031	0.05	0.1	0.15	0.25	0.41
experimental	0.02	0.04	0.15	0.10	0.19	0.47
THN: predicted	1.1	1.47	1.96	2.57	3.33	4.27
experimental	1.24	1.46	2.27	2.31	3.08	4.25

Table 1: Percentage of error. $n = 150$, $\alpha = 0.75$

m	100	200	400	800	1600	3200
(Threshold)	(147)	(147)	(148)	(149)	(149)	(150)
HN: predicted	0.0002	0.0003	0.0006	0.001	0.002	0.0036
experimental	0	0	0	0	0	0.01
THN: predicted	0.06	0.09	0.12	0.17	0.22	0.3
experimental	0.09	0.09	0.14	0.17	0.13	0.29

Table 2: Percentage of error. $n = 225$, $\alpha = 0.75$

Figure 2 presents the results of theoretical calculations of the HN and THN error probabilities, for various values of α and m as a function of n. Note the large difference in the memory capacity as α varies. For graphical convenience, we have plotted $\log \frac{1}{\epsilon}$ versus n. As seen above, a fair 'rule of thumb' is that a THN with $n' \approx 1.5n$ neurons in the input layer performs as well as a HN with n such neurons. To see this, simply pass a horizontal line through any error rate value ϵ, and observe the ratio between n and n' obtained at its intersection with the corresponding ϵ *vs.* n plots.

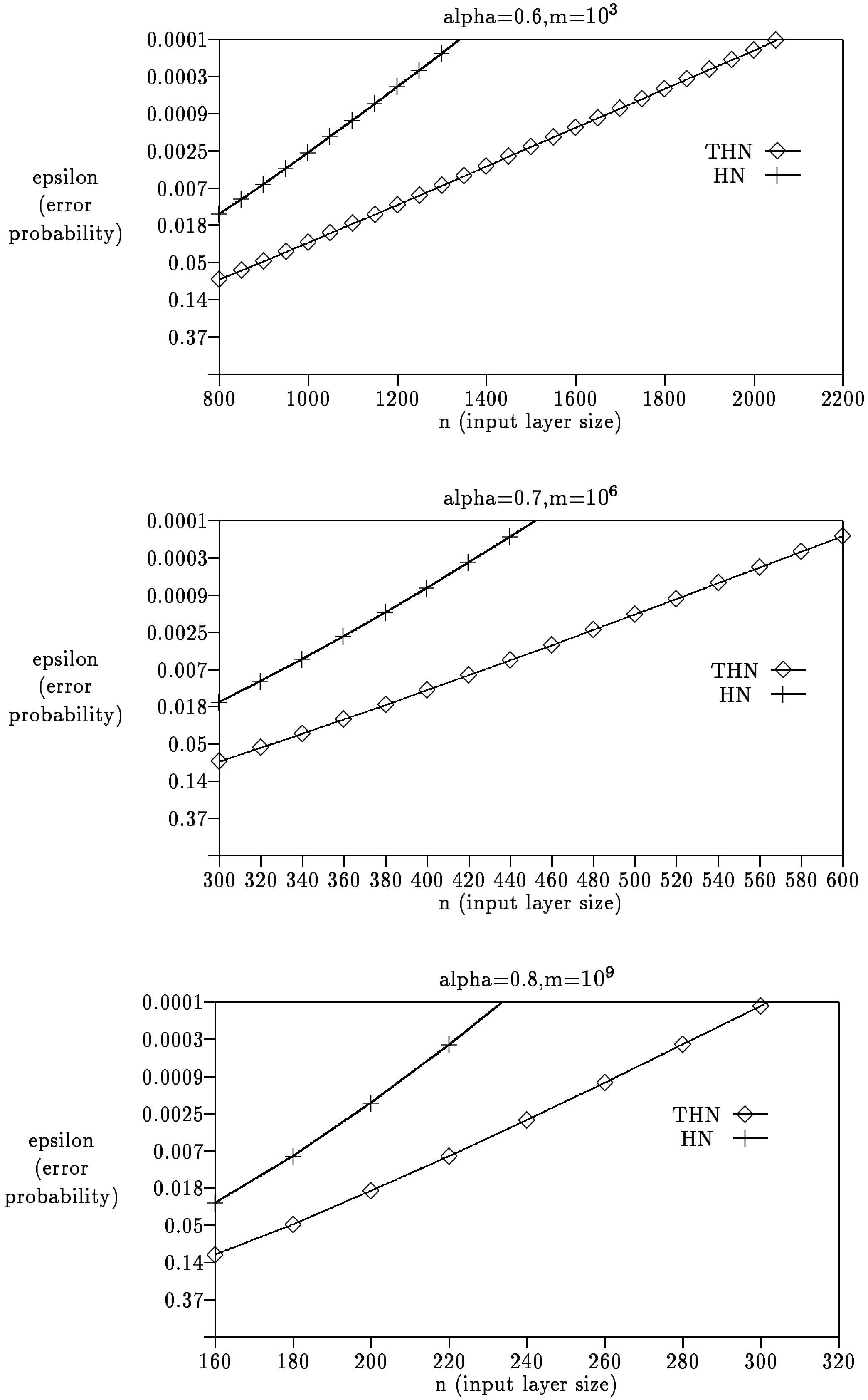

Figure 2: Probability of error as a function of network size: three networks are depicted, displaying the performance at various values of α and m.

To examine the sensitivity of the THN network to threshold variation, we have fixed $\alpha = 0.7$, $n = 210$, $m = 825$, and let the threshold vary between 132 and 138. As we can see in figure 3, the threshold value 135 is optimal, but the performance with threshold values of 134 and 136 is practically identical. The magnitude of the two error types varies considerably with the threshold value, but this variation has no effect on the overall performance near the optimum, and these two error probabilities might as well be taken equal to each other.

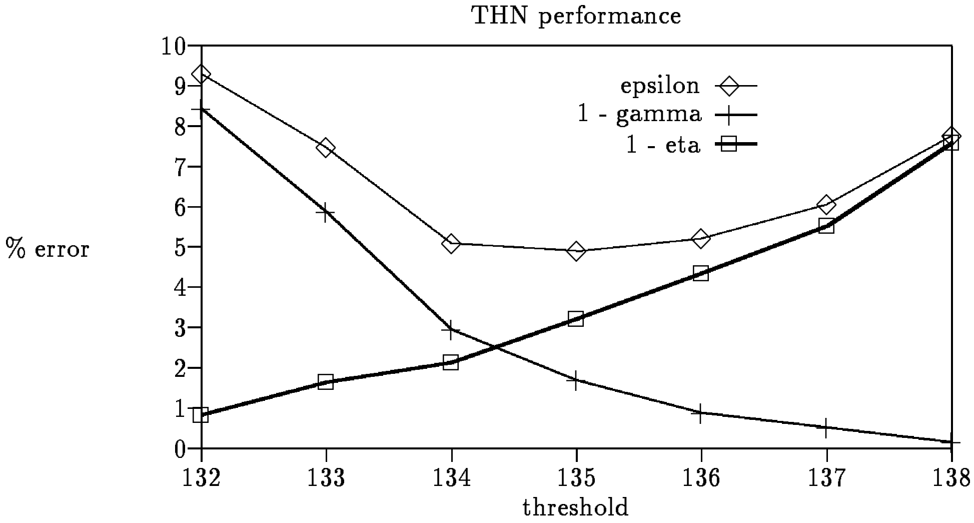

Figure 3: Threshold sensitivity of the THN ($\alpha = 0.7$, $n = 210$, $m = 825$).

2.5 Final Remarks:

In this section we analyzed in detail the performance of a HN and THN classifying inputs that are distorted versions of the stored memory patterns (in contrast to randomly selected patterns). Given an initial input similarity α, a desired storage capacity m and performance level $1 - \epsilon$, we described how to compute the minimal THN size n required to achieve this performance. As we have seen, the threshold x_n is determined as a function of the initial input similarity α. Obviously, however, the THN it defines will achieve even higher performance when presented with input patterns having initial similarity greater than α. It was shown that although the THN performs worse than its counterpart HN, an approximately 50% increase in the THN input layer size is sufficient to fully compensate for that. As the WTA network of the HN may be implemented with only $O(3m)$ connections [8], both the THN and the HN require $O(mn)$ connections. Hence, to perform as well as a given HN, the corresponding THN requires $\approx$ 50% more connections, but the $O(m \ln(mn))$ time complexity of the HN is drastically reduced to the $O(1)$ time complexity of the THN.

3 Two-Iteration Optimal Signaling in Hopfield Networks

3.1 Introduction

It is well known that a given cortical neuron can respond with a different firing pattern for the same synaptic input, depending on its firing history and on the effects of modulatory transmitters (see [10, 11] for a review). Working within the convenient framework of Hopfield-like attractor neural networks (ANN) [12, 13], but motivated by the history-dependent nature of neuronal firing, we continue our previous investigation of the two-iteration performance of feedback neural networks [14] (henceforth, M & R). We now extend our analysis to the study of continuous input/output signal functions which govern the firing rate of the neuron (such as the conventional sigmoidal function [15, 16]). The notion of a synchronous instantaneous 'iteration' is now viewed as an abstraction of the overall dynamics for some short length of time, during which the firing rate does not change significantly. We analyze the performance of the network after two such iterations, or intermediate times spans, a period sufficiently long for some significant neural information to be fed back within the network, but shorter than those the network may require for falling into an attractor. However, as demonstrated in subsection 3.6, the performance of history-dependent ANNs after two iterations is sufficiently high compared with that of memoryless (history-independent) models, that the analysis of two iterations becomes a viable end in its own right.

Examining this general family of signal functions, we now search for the computationally most efficient history-dependent neuronal signal (firing) function, and study its performance. We derive the optimal analog signal function, having the *slanted sigmoidal* form illustrated in figure 4a, and show that it significantly improves performance, both in relation to memoryless dynamics and versus the performance obtained with the previous dichotomous signaling. The optimal signal function is obtained by subtracting from the conventional sigmoid signal function some multiple of the current input field. As shown in figure 4a (or in figure 4b, plotting the discretized version of the optimal signal function) the neuron's signal may have a sign opposite to the one it believes in. [17, 18] and [19] have also observed that the capacity of ANNs is significantly improved by using nonmonotone analog signal functions. They studied the limit (after infinitely many iterations) under dynamics using a nonmonotone function of the current input field, similar in form to the slanted sigmoid. The Bayesian framework we work in provides, for the first time, a clear intuitive account of the non-monotone form and the seemingly bizarre sign reversal behavior. As we shall see, the slanted sigmoidal form of the optimal signal function is mainly a result of collective cooperation between neurons, whose 'common goal' is to maximize the network's performance. It is rather striking that the resulting slanted sigmoid endows the analytical model with some properties characteristic of cortical neurons' firing; this 'collectively optimal' function may be hard-wired into the cellular biophysical mechanisms determining each neuron's firing function.

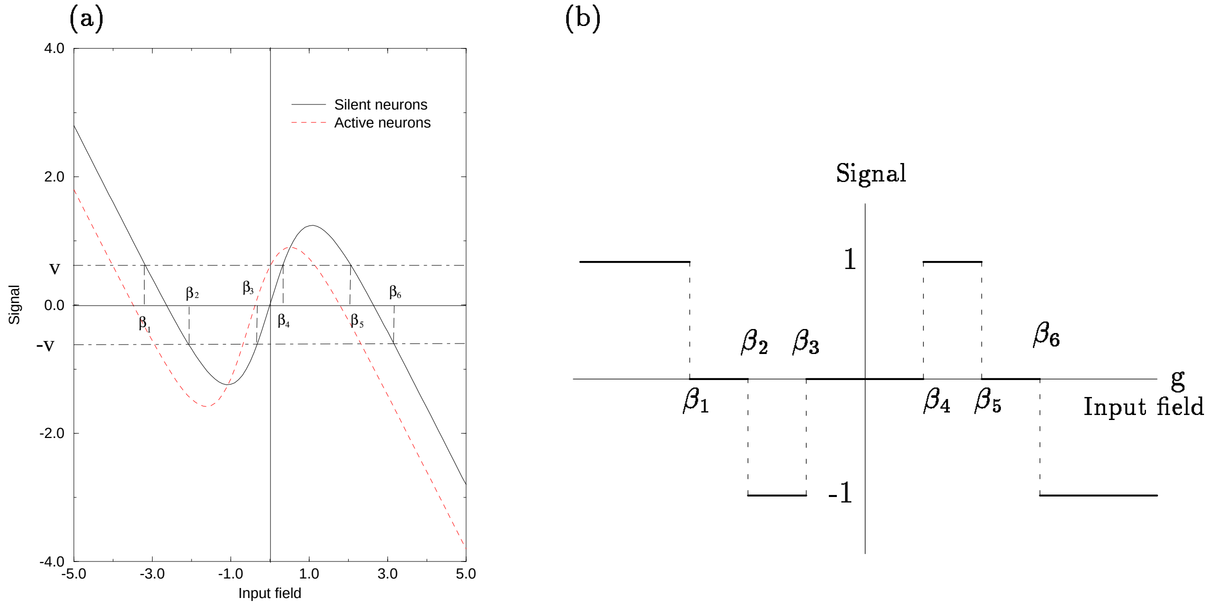

Figure 4: (a) A typical plot of the slanted sigmoid, Network parameters are $N = 5000$, $K = 3000$, $n_1 = 200$ and $m = 50$. (b) A sketch of its discretized version.

3.2 The model

Our framework is an ANN storing $m + 1$ memory patterns $\xi^1, \xi^2, \ldots, \xi^{m+1}$, each an N-dimensional vector. The network is composed of N neurons, each of which is randomly connected to K other neurons. The $(m+1)N$ memory entries are independent with equally likely ± 1 values. The initial pattern X, synchronously signaled by $L(\leq N)$ *initially active* neurons, is a vector of ± 1's, randomly generated from one of the memory patterns (say $\xi = \xi^{m+1}$) such that $P(X_i = \xi_i) = \frac{1+\epsilon}{2}$ for each of the L initially active neurons and $P(X_i = \xi_i) = \frac{1+\delta}{2}$ for each *initially quiescent* (non-active) neuron. Although $\epsilon, \delta \in [0, 1)$ are arbitrary, it is useful to think of ϵ as being 0.5 (corresponding to an initial similarity of 75%) and of δ as being zero - a quiescent neuron has no prior preference for any given sign.

Let $\alpha_1 = m/n_1$ denote the initial memory load, where $n_1 = LK/N$ is the average number of signals received by each neuron.

We follow a Bayesian approach under which the neuron's signaling and activation decisions are based on the a-posteriori probabilities assigned to its two possible true memory states, ± 1. We distinguish between input fields that model incoming spikes, and generalized fields that model history-dependent, adaptive post-synaptic potentials. Clearly, the *prior probability* that neuron i has memory state $+1$ is

$$\lambda_i{}^{(0)} = P(\xi_i = 1|X_i, I_i) = \begin{cases} \frac{1+\epsilon}{2} & \text{if } X_i = 1,\ I_i = 1 \\ \frac{1-\epsilon}{2} & \text{if } X_i = -1,\ I_i = 1 \\ \frac{1+\delta}{2} & \text{if } X_i = 1,\ I_i = 0 \\ \frac{1-\delta}{2} & \text{if } X_i = -1, I_i = 0 \end{cases} \tag{32}$$

$$= \frac{1 + (\epsilon I_i + \delta(1 - I_i))X_i}{2} = \frac{1}{1 + e^{-2g_i{}^{(0)}}}$$

where $I_i = 0, 1$ indicates whether neuron i has been active (i.e., transmitted a signal) in the first iteration, and the *generalized field* $g_i{}^{(0)}$ is given by

$$g_i{}^{(0)} = \begin{cases} g(\epsilon)X_i & \text{if } i \text{ is active} \\ g(\delta)X_i & \text{if } i \text{ is quiescent }. \end{cases} \tag{33}$$

where

$$g(t) = arctanh(t) = \frac{1}{2}\log\frac{1+t}{1-t} \quad ; \quad 0 \leq t < 1\ . \tag{34}$$

We also define the *prior belief* that neuron i has memory state $+1$

$$O_i{}^{(0)} = \lambda_i{}^{(0)} - (1 - \lambda_i{}^{(0)}) = 2\lambda_i{}^{(0)} - 1 = tanh(g_i{}^{(0)}) \tag{35}$$

whose possible values are $\pm\epsilon$ and $\pm\delta$ (The belief is simply a rescaling of the probability from the $[0, 1]$ interval to $[-1, +1]$).

The input field observed by neuron i as a result of the initial activity is

$$f_i{}^{(1)} = \frac{1}{n_1}\sum_{j=1}^{N} W_{ij} I_{ij} I_j X_j \tag{36}$$

where $I_{ij} = 0, 1$ indicates whether a connection exists from neuron j to neuron i and W_{ij} denotes its magnitude, given by the Hopfield prescription

$$W_{ij} = \sum_{\mu=1}^{m+1} \xi^\mu{}_i \xi^\mu{}_j \quad , \quad W_{ii} = 0. \tag{37}$$

As a result of observing the input field $f_i{}^{(1)}$, which is approximately normally distributed (given ξ_i, X_i and I_i) with mean and variance

$$E(f_i{}^{(1)}|\xi_i, X_i, I_i) = \epsilon\xi_i \tag{38}$$

$$Var(f_i{}^{(1)}|\xi_i, X_i, I_i) = \alpha_1 \ , \tag{39}$$

neuron i changes its opinion about $\{\xi_i = 1\}$ from $\lambda_i{}^{(0)}$ to the *posterior probability*

$$\lambda_i{}^{(1)} = P\left(\xi_i = 1|X_i, I_i, f_i{}^{(1)}\right) = \frac{1}{1 + e^{-2g_i{}^{(1)}}} \ , \tag{40}$$

with a corresponding *posterior belief* $O_i{}^{(1)} = tanh(g_i{}^{(1)})$, where $g_i{}^{(1)}$ is conveniently expressed as an additive generalized field (see Lemma $1(II)$ in M & R)

$$g_i{}^{(1)} = g_i{}^{(0)} + \frac{\epsilon}{\alpha_1} f_i{}^{(1)} \ . \tag{41}$$

We now get to the second iteration, in which, as in the first iteration, some of the neurons become active and signal to the network. Unlike the first iteration, in which initially active neurons had independent beliefs of equal strength and simply signaled their states in the initial pattern, the preamble to the second iteration finds neuron i in possession of a personal history $(X_i, I_i, f_i^{(1)})$, as a function of which the neuron has to determine the signal to transmit to the network. While the history-independent Hopfield dynamics choose sign$(f_i^{(1)})$ as this signal, we model the signal function as $h(g_i{}^{(1)}, X_i, I_i)$. This seems like four different functions of $g_i{}^{(1)}$. However, by symmetry, $h(g_i{}^{(1)}, +1, I_i)$ should be equal to $-h(-g_i{}^{(1)}, -1, I_i)$. Hence, we only have two functions of $g_i{}^{(1)}$ to define, $h_1(.)$ for the signals of the initially active neurons and $h_0(.)$ for the quiescent ones. For mathematical convenience we would like to insert into these functions random variables with unit variance. By (39) and (41), the conditional variance $Var(g_i{}^{(1)}|\xi_i, X_i, I_i)$ is $(\epsilon/\alpha_1)^2\alpha_1 = (\epsilon/\sqrt{\alpha_1})^2$. We thus define $\omega = \epsilon/\sqrt{\alpha_1}$ and let

$$h(g_i{}^{(1)}, X_i, I_i) = X_i h_{I_i}(X_i g_i{}^{(1)}/\omega) \ . \tag{42}$$

The field observed by neuron i following the second iteration (with n_2 updating neurons per neuron) is

$$f_i{}^{(2)} = \frac{1}{n_2}\sum_{j=1}^{N} W_{ij} I_{ij} h(g_j{}^{(1)}, X_j, I_j) \ , \tag{43}$$

on the basis of which neuron i computes its posterior probability

$$\lambda_i{}^{(2)} = P(\xi_i = 1|X_i, I_i, f_i{}^{(1)}, f_i{}^{(2)}) \tag{44}$$

and corresponding posterior belief $O_i{}^{(2)} = 2\lambda_i{}^{(2)} - 1$, which will be expressed in subsection 4.3 as $tanh(g_i{}^{(2)})$.

In this paper we stop at the above two information-exchange iterations and let each neuron express its final choice of sign as

$$X_i{}^{(2)} = sign(O_i{}^{(2)}) \ . \tag{45}$$

The performance of the network is measured by the final similarity

$$S_f = P(X_i{}^{(2)}) = \frac{1 + \frac{1}{N}\sum_{j=1}^{N} X_j{}^{(2)}\xi_j}{2} \tag{46}$$

(where the last equality holds asymptotically).

Our first task is to present (as simple as possible) an expression for the performance under arbitrary architecture and activity parameters, for general signal functions h_0 and h_1. Then, using this expression, our main goal is to find the best choice of signal functions which maximize the performance attained. We find these functions when there are either no restrictions on their range set or they are restricted to the values $\{-1, 0, 1\}$, and calculate the performance achieved in Gaussian, random and multi-layer patterns of connectivity. The optimal choice will be shown to be the *slanted sigmoid*

$$h(g_i{}^{(1)}, X_i, I_i) = O_i{}^{(1)} - cf_i{}^{(1)} \tag{47}$$

for some c in $(0, 1)$. We present explicitly all formulas, providing their derivation in [2].

3.3 Rationale for nonmonotone Bayesian signaling

3.3.1 Non-monotonicity

The common Hopfield convention is to have neuron i signal $sign(f_i{}^{(1)})$. Another possibility, studied in M & R, is to signal the preferred sign only if this preference is strong enough, otherwise remain silent. However, an even better performance was seen to be achieved by counterintuitive signals which are not monotone in $g_i{}^{(1)}$ [17, 19, 14]. In fact, precisely those neurons that are *most* convinced of their signs should signal the sign *opposite* to the one they so strongly believe in! We would like to offer now an intuitive explanation for this seeming pathology, and proceed later to the mathematics leading to it.

In the initial pattern, the different entries X_i and X_j are conditionally independent given ξ_i and ξ_j. This is not the case for the input fields $f_i{}^{(1)}$ and $f_j{}^{(1)}$, whose correlation is proportional to the synaptic weight W_{ij} (M & R). For concreteness, let $\epsilon = 0.5$ and $\alpha_1 = 0.25$ and suppose that neuron i has observed an input field $f_i{}^{(1)} = 3$. Neuron i now knows that either its true memory state is $\xi_i = +1$ in which case the 'noise' in the input field is $3 - \epsilon = 2.5$ (i.e., five standard deviations above the mean) or its true memory state is $\xi_i = -1$ and the noise is $3 + \epsilon = 3.5$ (or seven standard deviations above the mean). In a Gaussian distribution, deviations of five or seven standard deviations are very unusual, but seven is so much more unusual than five, that neuron i is practically convinced that its true state is $+1$. However, neuron i knows that its input field ${f_i}^{(1)}$ is grossly inflicted with noise and since the input field ${f_j}^{(1)}$ of neuron j is correlated with its own, neuron i would want to warn neuron j that its input field has unusual noise too and should not be believed on face value. Neuron i, a good student of Regression Analysis, wants to tell neuron j, without knowing the weight W_{ij}, to subtract from its field a multiple of $W_{ij}{f_i}^{(1)}$. This is accomplished, to the simultaneous benefit of all neurons j, by signaling a multiple of $-{f_i}^{(1)}$. We see that neuron i, out of 'purely altruistic traits', has a conflict between the positive act of signaling its assessed true sign and the negative act of signaling the opposite as a means of correcting the fields of its peers. It is not surprising that this inhibitory behavior is the dominant one only when field values are strong enough.

3.3.2 The Potential of Bayesian Updating

Neuron i starts with a prior probability ${\lambda_i}^{(0)} = P(\xi_i = +1)$ and after observing input fields ${f_i}^{(1)}, {f_i}^{(2)}, \ldots, {f_i}^{(t)}$ computes the posterior probability

$$ {\lambda_i}^{(t)} = P\left(\xi_i = +1 | {f_i}^{(1)}, {f_i}^{(2)}, \ldots, {f_i}^{(t)}\right) \tag{48} $$

It now signals

$$ {h_i}^{(t)} = h^{(t)}\left({\lambda_i}^{(0)}, {f_i}^{(1)}, {f_i}^{(2)}, \ldots, {f_i}^{(t)}\right) \tag{49} $$

and computes the new input field

$$ {f_i}^{(t+1)} = \sum_j W_{ij} I_{ij} {h_j}^{(t)} \ . \tag{50} $$

This description proceeds inductively.

The stochastic process ${\lambda_i}^{(0)}, {\lambda_i}^{(1)}, {\lambda_i}^{(2)}, \ldots$ is of the form

$$ X_t = E(Z | Y_1, Y_2, \ldots, Y_t) $$

where $Z = I_{\{\xi_i = +1\}}$ is a (bounded) random variable and the Y-process adds in every stage some more information to the data available earlier. Such a process is termed a *Martingale* in Probability theory. The following facts are well known, the first being actually the usual definition

1. For all t,

$$E(X_{t+1}|Y_1, Y_2, \ldots, Y_t) = X_t \;\; a.s.$$

   (where a.s. means 'almost surely' or 'except for an event with probability zero'.)

2. In particular, $E(X_t)$ is the same for all t.

3. If the finite interval $[a, b]$ is such that $P(a \leq X_t \leq b) = 1$ for all t and Ψ is a convex function on $[a, b]$, then for all t,

$$E(\Psi(X_{t+1})|Y_1, Y_2, \ldots, Y_t) \geq \Psi(X_t) \;\; a.s.$$

4. In particular, for all t,

$$E(\Psi(X_t)) \leq E(\Psi(X_{t+1}))$$

5. (A special case of Doob's Martingale Convergence Theorem)
   For every bounded Martingale (X_t) there is a random variable X such that

$$X_t \to X \;\; as \;\; t \to \infty \;, \; a.s.$$

   and in fact the Martingale is the sequence of 'opinions' about X: For all t,

$$X_t = E(X|Y_1, Y_2, \ldots, Y_t) \;\; a.s.$$

6. In particular, $E(X) = E(X_t)$ and $E(\Psi(X)) \geq E(\Psi(X_t))$ for all t, for any convex function Ψ defined on $[a, b]$.

A neuron with posterior probability ${\lambda_i}^{(t)}$ as in (48) decides momentarily that its true state is $+1$ if ${\lambda_i}^{(t)} > 1/2$ and -1 if ${\lambda_i}^{(t)} < 1/2$. The strength of belief, or confidence in the preferred state, is given by the *convex* function $\Psi(x) = Max(x, 1-x)$ applied to the $[0, 1]$-bounded Martingale $({\lambda_i}^{(t)})$. For large N, the current *similarity* of the network, or proportion of neurons whose preferred state is the correct one, is mathematically characterized as $E\left(\Psi({\lambda_i}^{(t)})\right)$. By the above, Bayesian updatings are always such that every neuron has a well defined final decision about its state (we may call this a 'fixed point') and the network's similarity increases with every iteration, being at the 'fixed point' even higher. This holds true for arbitrary signal functions h, and not only for those that are in some sense optimal. By the above, whatever similarity we achieve after two Bayesian iterations is a lower bound for what can be achieved by more iterations, unlike memoryless Hopfield dynamics which are known to do reasonably well at the beginning even below capacity, in which case they converge eventually to random fixed points [20].

3.4 Performance

3.4.1 Architecture parameters

This subsection introduces and illustrates certain parameters whose relevance will become apparent in subsection 3.4.3. There are N neurons in the network and K incoming synapses projecting on every neuron. If there is a synapse from neuron i to neuron j, the probability is r_2 that there is a synapse from neuron j to neuron i. If there are synapses from i to j and from j to k, the probability is r_3 that there is a synapse from i to k. If there are synapses form i to each of j and k, and from j to l, the probability is r_4 that there is a synapse from k to l.

We saw in M & R that Bayesian neurons are adaptive enough to make r_2 irrelevant for performance, but that r_3 and r_4, which we took simply to be K/N assuming *fully random connectivity*, are of relevance. It is clear that if each neuron is connected to its K closest neighbors, then r_2 is 1 and r_3 and r_4 are large. For *fully connected* networks all three are equal to 1.

For *Gaussian connectivity*, if neurons i and j are at a distance x from each other, then the probability that there is a synapse from j to i is

$$P(synapse) = pe^{-\frac{x^2}{2s^2}} \tag{51}$$

where $p \in (0, 1]$ and $s^2 > 0$ are parameters. Since the sum of n independent and identically distributed Gaussian random vectors is Gaussian with variance n times as large as that of the summands, we get that in d-dimensional space

$$r_k = \int \left(pe^{-\frac{1}{2s^2}\sum_{i=1}^{d} {x_i}^2}\right) \frac{e^{-\frac{1}{2s^2(k-1)}\sum_{i=1}^{d} {x_i}^2}}{(2\pi(k-1)s^2)^{d/2}} dx_1 dx_2 \ldots dx_d \tag{52}$$

$$= \frac{p}{k^{d/2}} \int \frac{e^{-\frac{1}{2s^2((k-1)/k)}\sum_{i=1}^{d} {x_i}^2}}{(2\pi s^2((k-1)/k))^{d/2}} dx_1 dx_2 \ldots dx_d = \frac{p}{k^{d/2}} \ .$$

Thus, in 3-dimensional space, $r_2 = p/(2\sqrt{2})$, $r_3 = p/(3\sqrt{3})$, $r_4 = p/8$, depending on the parameter p but not on s.

For *multilayered networks* in which there is full connectivity between consecutive layers but no other connections, r_2 and r_4 are equal to 1 and r_3 is 0 (unless there are three layers cyclically connected, in which case $r_3 = 1$ as well).

3.4.2 One-iteration performance

Clearly, if neuron i had to choose for itself a sign on the basis of one iteration, this sign would have been

$$X_i{}^{(1)} = sign(O_i{}^{(1)}) \ . \tag{53}$$

Hence, letting $\omega = \epsilon/\sqrt{\alpha_1}$, if $P(X_i = \xi_i) = (1+t)/2$ (where t is either ϵ or δ), then after one iteration (similar to [21]),

$$P(X_i{}^{(1)} = \xi_i) = P(\lambda_i{}^{(1)} > 0.5|\xi_i = 1) = P\left(g(t)X_i + \frac{\epsilon}{\alpha_1} f_i{}^{(1)} > 0|\xi_i = 1\right) \tag{54}$$

$$= \frac{1+t}{2} P\left(g(t) + \frac{\epsilon}{\alpha_1}(\epsilon + \sqrt{\alpha_1}Z) > 0\right) + \frac{1-t}{2} P\left(-g(t) + \frac{\epsilon}{\alpha_1}(\epsilon + \sqrt{\alpha_1}Z) > 0\right)$$

$$= \frac{1+t}{2}\Phi\left(\omega + \frac{g(t)}{\omega}\right) + \frac{1-t}{2}\Phi\left(\omega - \frac{g(t)}{\omega}\right)$$

where Z is a standard normal random variable and Φ is its distribution function. Letting

$$Q^*(x,t) = \frac{1+t}{2}\Phi(x + \frac{g(t)}{x}) + \frac{1-t}{2}\Phi(x - \frac{g(t)}{x}) \ ; \ 0 \le t < 1, x > 0 \ , \tag{55}$$

we see that (54) is expressible as $Q^*(\omega, t)$. Since the proportion of initially active neurons is n_1/K, the similarity after one iteration is

$$S_1 = \frac{n_1}{K} Q^*(\omega, \epsilon) + \left(1 - \frac{n_1}{K}\right) Q^*(\omega, \delta) \ . \tag{56}$$

As for the relation between the current similarity S_1 and the initial similarity, observe that $Q^*(x,t)$ is strictly increasing in x and converges to $\frac{1+t}{2}$ as $x \downarrow 0$. Hence, S_1 strictly exceeds the initial similarity $\frac{n_1}{K}\frac{1+\epsilon}{2} + (1 - \frac{n_1}{K})\frac{1+\delta}{2}$. Furthermore, S_1 is a strictly increasing function of n_1 $(= m/\alpha_1)$.

3.4.3 The second iteration

In order to analyze the effect of a second iteration, it is necessary to identify the (asymptotic) conditional distribution of the new input field $f_i{}^{(2)}$, defined by (43), given $(\xi_i, X_i, I_i, f_i{}^{(1)})$. Under a working paradigm that, given ξ_i, X_i and I_i, the input fields $(f_i{}^{(1)}, f_i{}^{(2)})$ are jointly normally distributed, the conditional distribution of $f_i{}^{(2)}$ given $(\xi_i, X_i, I_i, f_i{}^{(1)})$ should be normal with mean depending linearly on $f_i{}^{(1)}$ and variance independent of $f_i{}^{(1)}$. More explicitly, if (U, V) are jointly normally distributed with correlation coefficient $\rho = Cov(U, V)/(\sigma_U \sigma_V)$, then

$$E(V|U) = E(V) + \rho(\sigma_V/\sigma_U)(U - E(U)) \tag{57}$$

and

$$Var(V|U) = Var(V)(1-\rho^2)\ . \tag{58}$$

Thus, the only parameters needed to define dynamics and evaluate performance are $E({f_i}^{(2)}|\xi_i, X_i, I_i)$, $Cov({f_i}^{(1)}, {f_i}^{(2)}|\xi_i, X_i, I_i)$ and $Var({f_i}^{(2)}|\xi_i, X_i, I_i)$. In terms of these, the conditional distribution of ${f_i}^{(2)}$ given $(\xi_i, X_i, I_i, {f_i}^{(1)})$ is normal with

$$E({f_i}^{(2)}|\xi_i, X_i, I_i, {f_i}^{(1)}) = \tag{59}$$

$$= E({f_i}^{(2)}|\xi_i, X_i, I_i) + \frac{Cov({f_i}^{(1)}, {f_i}^{(2)}|\xi_i, X_i, I_i)}{Var({f_i}^{(1)}|\xi_i, X_i, I_i)}\left({f_i}^{(1)} - E({f_i}^{(1)}|\xi_i, X_i, I_i)\right)$$

and

$$Var({f_i}^{(2)}|\xi_i, X_i, I_i, {f_i}^{(1)}) = Var({f_i}^{(2)}|\xi_i, X_i, I_i) - \frac{Cov^2({f_i}^{(1)}, {f_i}^{(2)}|\xi_i, X_i, I_i)}{Var({f_i}^{(1)}|\xi_i, X_i, I_i)}\ . \tag{60}$$

Assuming a model of joint normality, as in M & R, we rigorously identify limiting expressions for the three parameters of the model. Although we do not have as yet sufficient formal evidence pointing to the correctness of the joint normality assumption, the simulation results presented in subsection 3.6 fully support the adequacy of this common model.

In M & R we proved that $E({f_i}^{(2)}|\xi_i, X_i, I_i)$ is a linear combination of ξ_i and X_iI_i, which we denote by

$$E({f_i}^{(2)}|\xi_i, X_i, I_i) = \epsilon^*\xi_i + bX_iI_i\ . \tag{61}$$

We also proved that $Cov({f_i}^{(1)}, {f_i}^{(2)}|\xi_i, X_i, I_i)$ and $Var({f_i}^{(2)}|\xi_i, X_i, I_i)$ are independent of (ξ_i, X_i, I_i). These parameters determine the *regression coefficient*

$$a = \frac{Cov({f_i}^{(1)}, {f_i}^{(2)}|\xi_i, X_i, I_i)}{Var({f_i}^{(1)}|\xi_i, X_i, I_i)} \tag{62}$$

and the *residual variance*

$$\tau^2 = Var({f_i}^{(2)}|\xi_i, X_i, I_i, {f_i}^{(1)})\ . \tag{63}$$

These facts remain true in the current more general framework. We present in [2] formulas for a, b, ϵ^* and τ^2, whose derivation is cumbersome. The posterior probability that neuron i has memory state $+1$ is (see (40) and Lemma $1(II)$ in M & R)

$$\lambda_i^{(2)} = P(\xi_i = 1|X_i, I_i, {f_i}^{(1)}, {f_i}^{(2)}) = \tag{64}$$

$$= \frac{1}{1 + \exp\{-2\left[{g_i}^{(1)} + \frac{\epsilon^* - a\epsilon}{\tau^2}\left({f_i}^{(2)} - a{f_i}^{(1)} - bX_iI_i\right)\right]\}}$$

from which we obtain the final belief ${O_i}^{(2)} = 2{\lambda_i}^{(2)} - 1 = tanh({g_i}^{(2)})$, where ${g_i}^{(2)}$ should be defined as

$$ {g_i}^{(2)} = \left(\frac{\epsilon}{\alpha_1} - \frac{(\epsilon^* - a\epsilon)a}{\tau^2}\right) {f_i}^{(1)} + \left(\frac{\epsilon^* - a\epsilon}{\tau^2}\right) {f_i}^{(2)} + \begin{cases} g(\delta)X_i & \text{if } I_i = 0 \\ \left(g(\epsilon) - \frac{b(\epsilon^* - a\epsilon)}{\tau^2}\right) X_i & \text{otherwise} \end{cases} \tag{65} $$

to yield the final decision ${X_i}^{(2)} = sign({g_i}^{(2)})$. Since $({f_i}^{(1)}, {f_i}^{(2)})$ are jointly normally distributed given (ξ_i, X_i, I_i), any linear combination of the two, such as the one in expression (65), is normally distributed. After identifying its mean and variance, a standard computation reveals that the final similarity $S_2 = P({X_i}^{(2)} = \xi_i)$ - our global measure of performance - is given by a formula similar to expression (56) for S_1, with heavier activity n^* than n_1:

$$ S_2 = \frac{n_1}{K} Q^* \left(\frac{\epsilon}{\sqrt{\alpha^*}}, \epsilon\right) + \left(1 - \frac{n_1}{K}\right) Q^* \left(\frac{\epsilon}{\sqrt{\alpha^*}}, \delta\right) \tag{66} $$

where

$$ \alpha^* = \frac{m}{n^*} = \frac{m}{n_1 + m\left(\frac{\epsilon^*/\epsilon - a}{\tau}\right)^2} \; . \tag{67} $$

In agreement with the ever-improving nature of Bayesian updatings, S_2 exceeds S_1 just as S_1 exceeds the initial similarity. Furthermore, S_2 is an increasing function of $|\frac{\epsilon^*/\epsilon - a}{\tau}|$.

3.5 Optimal signaling and performance

By optimizing over the factor $|\frac{\epsilon^*/\epsilon - a}{\tau}|$ determining performance, we show in [2] that the optimal signal functions are

$$ h_1(y) = R^*(y, \epsilon) - 1 \;,\; h_0(y) = R^*(y, \delta) \tag{68} $$

where R^* is

$$ R^*(y, t) = \frac{1}{\epsilon}(1 + r_3\omega^2)\left[tanh(\omega y) - c(\omega y - g(t))\right] \tag{69} $$

and c is a constant in $(0, 1)$.

The nonmonotone form of these functions, illustrated in figure 4, is clear. Neurons that have already signaled $+1$ in the first iteration have a lesser tendency to send positive signals than quiescent neurons. The signaling of quiescent neurons which receive no prior information ($\delta = 0$) has a symmetric form.

The signal function of the initially active neurons may be shifted without affecting performance: if instead of taking $h_1(y)$ to be $R^*(y, \epsilon) - 1$ we take it to be $R^*(y, \epsilon) - 1 + \Delta$

for some arbitrary Δ, we will get the same performance because the effect of such Δ on the second iteration input field ${f_i}^{(2)}$ would be (see (43)) the addition of

$$\frac{1}{n_2}\sum_{j=1}^{N} W_{ij}I_{ij}\Delta X_j I_j = \Delta\frac{n_1}{n_2}{f_i}^{(1)} \tag{70}$$

which history-based Bayesian updating rules can fully adapt to. As shown in [2], Δ appears nowhere in $(\epsilon^*/\epsilon - a)$ nor in τ but it affects a. Hence, Δ may be given several roles:

- Setting the ratio of the coefficients of ${f_i}^{(1)}$ and ${f_i}^{(2)}$ in (65) to a desired value, mimicking the passive decay of the membrane potential.

- Making the final decision ${X_i}^{(2)}$ (see (65)) free of ${f_i}^{(1)}$, by letting the coefficient of the latter vanish. A judicious choice of the value of the reflexivity parameter r_2 (which, just as Δ, doesn't affect performance) can make the final decision ${X_i}^{(2)}$ free of whether the neuron was initially quiescent or active. For the natural choice $\delta = 0$ this will make the final decision free of the initial state as well and become simply the usual history-independent Hopfield rule ${X_i}^{(2)} = sign({f_i}^{(2)})$, except that ${f_i}^{(2)}$ is the result of carefully tuned slanted sigmoidal signaling.

- We may take $\Delta = 1$ in which case both functions h_0 and h_1 are given simply by $R^*(y,t)$, where $t = \epsilon$ or δ depending on whether the neuron is initially active or quiescent. Let us express this signal explicitly in terms of history. By Table 1 and expression (42), the signal emitted by neuron i (whether it is active or quiescent) is

$$h\left({g_i}^{(1)}, X_i, I_i\right) = X_i h_{I_i}\left(X_i {g_i}^{(1)}/\omega\right) = \tag{71}$$
$$\frac{1+r_3\omega^2}{\epsilon} X_i \left[tanh(X_i {g_i}^{(1)}) - c(X_i {g_i}^{(1)} - g(t))\right] =$$
$$\frac{1+r_3\omega^2}{\epsilon}\left[tanh({g_i}^{(1)}) - c\left({g_i}^{(1)} - X_i g(t)\right)\right] = \frac{1+r_3\omega^2}{\epsilon}\left[tanh({g_i}^{(1)}) - c{f_i}^{(1)}\right] \ .$$

We see that the signal is essentially equal to the sigmoid (see expression (41)) $tanh({g_i}^{(1)}) = 2{\lambda_i}^{(1)} - 1$, modified by a correction term depending only on the current input field, in full agreement with the intuitive explanations of subsection 2. This correction is never too strong; note that c is always less than 1. In a fully-connected network c is simply

$$c = \frac{1}{1+\omega^2} \ ,$$

i.e., in the limit of low memory load ($\omega \to \infty$), the best signal is simply a sigmoidal function of the generalized input field.

To obtain a discretized version of the slanted sigmoid, we let the signal be $sign(h(y))$ as long as $|h(y)|$ is big enough - where h is the slanted sigmoid. The resulting signal, as a function of the generalized field, is (see figure 4a and 4b)

$$h_j(y) = \begin{cases} +1 & y < {\beta_1}^{(j)} \text{ or } {\beta_4}^{(j)} < y < {\beta_5}^{(j)} \\ -1 & y > {\beta_6}^{(j)} \text{ or } {\beta_2}^{(j)} < y < {\beta_3}^{(j)} \\ 0 & \text{otherwise} \end{cases} \tag{72}$$

where $-\infty < {\beta_1}^{(0)} < {\beta_2}^{(0)} \leq {\beta_3}^{(0)} < {\beta_4}^{(0)} \leq {\beta_5}^{(0)} < {\beta_6}^{(0)} < \infty$ and $-\infty < {\beta_1}^{(1)} < {\beta_2}^{(1)} \leq {\beta_3}^{(1)} < {\beta_4}^{(1)} \leq {\beta_5}^{(1)} < {\beta_6}^{(1)} < \infty$ define, respectively, the firing pattern of the neurons that were silent or active in the first iteration. To find the best such discretized version of the optimal signal, we search numerically for the activity level v which maximizes performance. Every activity level v, used as a threshold on $|h(y)|$, defines the (at most) twelve parameters ${\beta_i}^{(j)}$ (which are identified numerically via the Newton-Raphson method) as illustrated in figure 4b.

3.6 Results

Using the formulation presented in the previous subsection, we investigate numerically the two-iteration performance achieved in several network architectures with optimal analog and discretized signaling.

Figure 5 displays the performance achieved in the network, when the input signal is applied only to the small fraction (4%) of neurons which are active in the first iteration (expressing possible limited resources of input information). While low activity is enforced in the first iteration, the number of neurons allowed to become active in the second iteration is not restricted, and best performance is typically achieved when about 70% of the neurons in the network are active (both with optimal signaling and with the previous, heuristic signaling). We see that (for $K > 1000$) near perfect final similarity is achieved even when the 96% initially quiescent neurons get no initial clue as to their true memory state, if no restrictions are placed on the second iteration activity level. The performance loss due to discretization is not considerable.

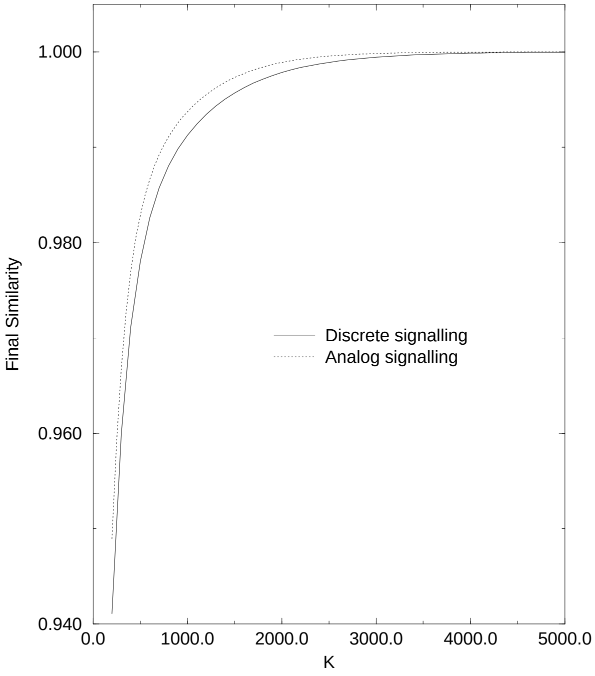

Figure 5: Two-iteration performance in a low-activity network as a function of connectivity K. Network parameters are $N = 5000$, $m = 50$, $n_1 = 200$, $\epsilon = 0.5$ and $\delta = 0$.

Figure 6 illustrates the performance when connectivity and the number of signals received by each neuron are held fixed, but the network size is increased. A region of decreased performance is evident at mid-connectivity ($K \approx N/2$) values, due to the increased residual variance. Hence, for neurons capable of forming K connections on the average, the network should either be fully connected or have a size N much larger than K. Since (unavoidable eventually) synaptic deletion would sharply worsen the performance of fully connected networks, cortical ANNs should indeed be sparsely connected. As evident, performance approaches an upper limit (the performance achieved with $r_3 = 0$ and $r_4 = 0$) as the network size is increased, and any further increase in the network size is unrewarding. The final similarity achieved in the fully connected network (with $N = K = 200$) should be noted. In this case, the memory load (0.2) is significantly above the critical capacity of the Hopfield network [22], but optimal history-dependent dynamics still manage to achieve a rather high two-iterations similarity (0.975) from initial similarity 0.75. This is in agreement with the findings of [18, 17], who show that nonmonotone dynamics increase capacity.

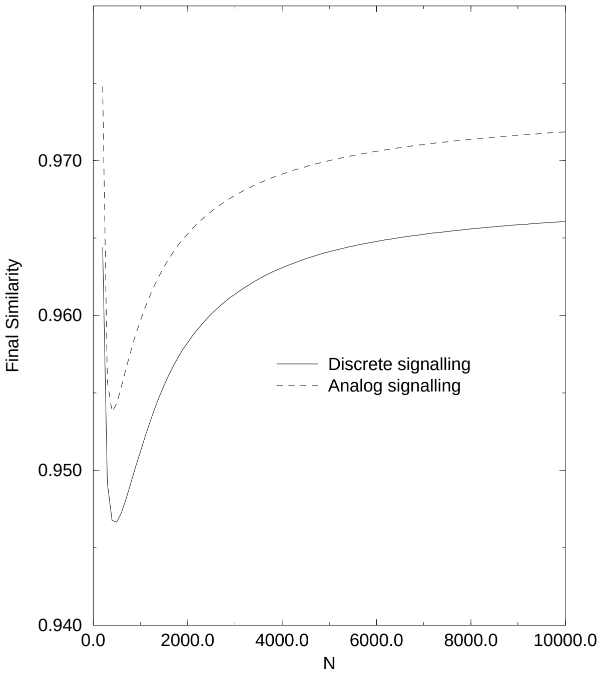

Figure 6: Two-iteration performance in a full-activity network as a function of network size N. Network parameters are $n_1 = K = 200$, $m = 40$ and $\epsilon = 0.5$.

Our theoretical predictions have been extensively examined by network simulations, and already in relatively small-scale networks close correspondence is achieved. For example, simulating a fully-connected network storing 100 memories with 500 neurons, the performance achieved with discretized dynamics under initial full activity (averaged over 100 trials, with $\epsilon = 0.5$ and $\delta = 0$) was 0.969 versus the 0.964 predicted theoretically. When m, n_1 and K were reduced by half (i.e., $N = 500$, $K = 250$,$m = 50$ and $n_1 = 250$) the predicted performance was 0.947 and that achieved in simulation was 0.946. When m, n_1 and K were further reduced by half (into $K = 125$, $m = 25$ and $n_1 = 125$) the predicted performance was 0.949 and that actually achieved was 0.953. In a larger network, with $N = 1500$, $K = 500$, $m = 50$, $n_1 = 250$, $\epsilon = 0.5$ and $\delta = 0$, the predicted performance is 0.977 and that obtained numerically was 0.973.

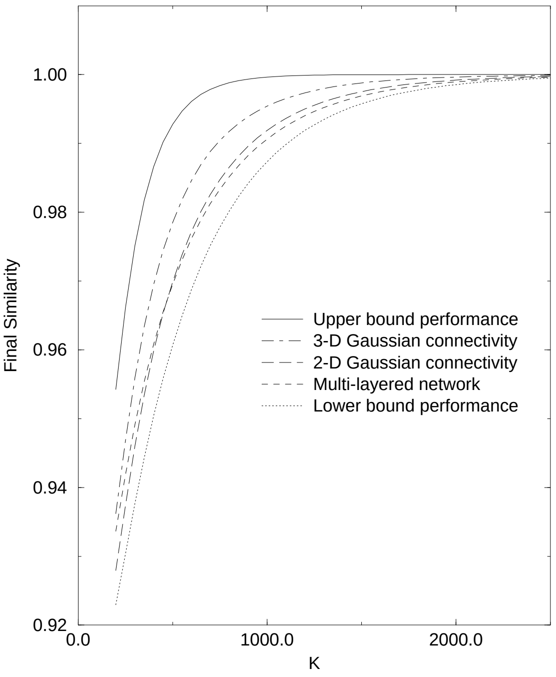

Figure 7: Two-iteration performance achieved with various network architectures, as a function of the network connectivity K. Network parameters are $N = 5000$, $n_1 = 200$, $m = 50$, $\epsilon = 0.5$ and $\delta = 0$.

Figure 7 illustrates the performance achieved with various network architectures, all sharing the same network parameters N, K, m and input similarity parameters n_1, ϵ, δ, but differing in the spatial organization of the neurons' synapses. Five different configurations are examined, characterized by different values of the architecture parameters r_3 and r_4, as described in subsection 3.4.1. The upper bound on the final similarity that can be achieved in ANNs in two iterations is demonstrated by letting $r_3 = 0$ and $r_4 = 0$. A lower bound (i.e., the worst possible architecture) on the performance gained with optimal signaling has been calculated by letting $r_4 = 1$ and searching for r_3 values that yielded the worst performance (such values began around 0.6 and increased to ≈ 0.8 as K was increased). The performance of the Multi-layered architecture was calculated by letting $r_4 = 1$ and $r_3 = 0$. Finally, the worst performance achievable with 2-D and 3-D Gaussian connectivity (corresponding to $p = 1$ in (51)) has been demonstrated by letting $r_3 = 1/3$,$r_4 = 1/4$ and $r_3 = 1/(3\sqrt{3})$,$r_4 = 1/8$ respectively. As evident, even in low-activity sparse-connectivity conditions, the decrease in performance with Gaussian connectivity (in relation, say, to the upper bound) does not seem considerable. Hence, history-dependent ANNs can work well in a cortical-like architecture. It is interesting but not surprising to see that 3-D Gaussian-connectivity architecture is superior to the 2-D one along the whole connectivity range. Random connectivity, with $r_3 = r_4 = K/N$, is not displayed but is slightly above the performance achieved with 3-D Gaussian connectivity.

3.7 Discussion

We have shown that Bayesian history-dependent dynamics make performance increase with every iteration, and that two iterations already achieve high similarity. The Bayesian framework gives rise to the slanted-sigmoid as the optimal signal function, displaying the nonmonotone shape proposed by [18]. The two-iteration performance has been analyzed in terms of general connectivity architectures, initial similarity and activity level.

The optimal signal function has some interesting biological perspectives. The possibly asymmetric form of the function, where neurons that have been silent in the previous iteration have an increased tendency to fire in the next iteration versus previously active neurons, is reminiscent of the bi-threshold phenomenon observed in biological neurons (see [23] for a review), where the threshold of neurons held at a hyperpolarized potential for a prolonged period of time is significantly lowered. As we have shown in subsection 3.5, the precise value of the parameter Δ leads to different biological interpretations of the slanted sigmoid signal function. The most obvious one is letting Δ set the ratio of the coefficients of ${f_i}^{(1)}$ and ${f_i}^{(2)}$ so as to mimic the decay of the membrane voltage. Perhaps more important, the finding that history-dependent neurons can maintain optimal performance in face of a broad range of Δ values points out that neuromodulators may change the form of the signal function without changing the performance of the network. Obviously, the history-free variant of the optimal final decision is not resilient to such modulatory changes.

The performance of ANN models can be heavily affected by dynamics, as exhibited by the sharp improvements obtained by fine tuning the neuron's signal function. When there is a sizable evolutionary advantage to fine tuning, theoretical optimization becomes an important research tool: the solutions it provides and the qualitative features it deems critical may have their parallels in reality. In addition to the computational efficiency of nonmonotone signaling, the numerical investigations presented in the previous subsection point out to a few more features with possible biological relevance:

- In an efficient associative network, input patterns should be applied with high fidelity on a small subset of neurons, rather than spreading a given level of initial similarity as a low fidelity stimulus applied to a large subset of neurons.

- If neurons have some restriction on the number of connections they may form, such that each neuron forms some K connections on the average, then efficient ANNs, converging to high final similarity within few iterations, should be sparsely connected.

- With a properly tuned signal function, cortical-like Gaussian-connectivity ANNs perform nearly as well as randomly-connected ones.

4 Concluding Remarks

This chapter has presented efficient dynamics for fast memory retrieval in both Hamming and Hopfield networks. However, as shown in this chapter, the linear (in network size) capacity of the Hopfield network is no match to the exponential capacity of the Hamming network, even with efficient dynamics. Yet, it is tempting to believe that the more biologically-plausible distributed encoding manifested in the Hopfield network may have its own computational advantages. In our minds, a promising future challenge might be the development of Hamming-Hopfield 'hybrid' networks which may allow one to enjoy the merits of both paradigms. A possible step towards this goal may involve the incorporation of the activation dynamics presented in this chapter, in a unified manner.

The feasibility of designing a hybrid Hamming-Hopfield network stems from the straightforward observation that the single-layer Hopfield network dynamics can be mapped in a one-to-one manner onto a bi-layered Hamming network architecture. This is easy to see by noting that each Hopfield iteration calculating the input field f_i of neuron i may be represented as

$$f_i = \sum_j W_{ij} X_j = \sum_j \sum_\mu \xi^\mu{}_i \xi^\mu{}_j X_j = \sum_\mu \xi^\mu{}_i \sum_j \xi^\mu{}_j X_j = \sum_\mu \xi^\mu{}_i O v_\mu \tag{73}$$

where, in the terminology of the HN, $Ov_\mu = (Z_\mu - n)/2$. Hence, each iteration in the original 1-layered Hopfield network may be carried out by performing two sub-iterations in the bi-layered Hamming architecture: In the first, the input pattern is applied to the input layer, and the resulting overlaps Ov_μ are calculated on the memory layer. Thereafter, in the second sub-iteration, these overlaps are used following equation (73) to calculate the new input fields of the next Hopfield iteration for the neurons of the input layer. This hybrid network architecture hence raises the possibility of finding efficient signaling functions which may enhance its performance, and lead to highly efficient memory systems.

As evident, there is much to gain in terms of space and time complexity by using efficient dynamics in both feedforward and feedback networks. One may wonder if such efficient signaling functions would have biological counterparts in the brain.

References

[1] I. Meilijson, E. Ruppin, and M. Sipper. A single iteration threshold hamming network. *IEE Trans. of NN*, 6 (1):261–266, 1995.

[2] I. Meilijson and E. Ruppin. Optimal signalling in attractor neural networks. *Network*, 5 (2):277–298, 1994.

[3] K. Steinbuch. Die lernmatrix. *Kybernetic*, 1:36–45, 1961.

[4] K. Steinbuch and U.A.W. Piske. Learning matrices and their applications. *IEEE Transactions on Electronic Computers*, pages 846–862, 1963.

[5] W.K. Taylor. Cortico-thalamic organization and memory. *Proc. of the Royal Society of London B*, 159:466–478, 1964.

[6] R.P. Lippmann, B. Gold, and M.L. Malpass. A comparison of Hamming and Hopfield neural nets for pattern classification. Technical Report TR-769, MIT Lincoln Laboratory, 1987.

[7] E.E. Baum, J. Moody, and F. Wilczek. Internal representations for associative memory. *Biological Cybernetics*, 59:217–228, 1987.

[8] P. Floreen. The convergence of hamming memory networks. *IEEE Trans. on Neural Networks*, 2(4):449–457, 1991.

[9] M.R. Leadbetter, G. Lindgren, and H. Rootzen. *Extremes and related properties of random sequences and processes.* Springer-Verlag, Berlin-Heidelberg-NewYork, 1983.

[10] B.W. Connors and M.J. Gutnick. Intrinsic firing patterns of diverse neocortical neurons. *TINS*, 13(3):99–104, 1990.

[11] Peter C. Schwindt. Ionic currents governing input-output relations of betz cells. In T. McKenna, J. Davis, and S.F. Zornetzer, editors, *Single neuron computation*, pages 235–258. Academic Press, 1992.

[12] J.J. Hopfield. Neural networks and physical systems with emergent collective abilities. *Proc. Nat. Acad. Sci. USA*, 79:2554, 1982.

[13] J.J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Nat. Acad. Sci. USA*, 81:3088, 1984.

[14] I. Meilijson and E. Ruppin. History-dependent attractor neural networks. *Network*, 4:195–221, 1993.

[15] H.R. Wilson and J.D. Cowan. Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical Journal*, 12:1–24, 1972.

[16] J.C. Pearson, L.H. Finkel, and G.M.Edelman. Plasticity in the organization of adult cerebral cortical maps: A computer simulation based on neuronal group selection. *Journal of Neuroscience*, 7(12):4209–4223, 1987.

[17] S. Yoshizawa, M. Morita, and S.-I. Amari. Capacity of associative memory using a nonmonotonic neuron model. *Neural Networks*, 6:167–176, 1993.

[18] M. Morita. Associative memory with nonmonotone dynamics. *Neural Networks*, 6:115–126, 1993.

[19] P. De Felice, C. Marangi, G. Nardulli, G. Pasquariello, and L. Tedesco. Dynamics of neural networks with non-monotone activation function. *Network*, 4:1–9, 1993.

[20] S.I. Amari and K. Maginu. Statistical neurodynamics of associative memory. *Neural Networks*, 1:67–73, 1988.

[21] H. Englisch, A. Engel, A. Schutte, and M. Stcherbina. Improved retrieval in nets of formal neurons with thresholds and non-linear synapses. *Studia Biophysica*, 137:37–54, 1990.

[22] D. J. Amit, H. Gutfreund, and H. Sompolinsky. Storing infinite numbers of patterns in a spin-glass model of neural networks. *Phys. Rev. Lett.*, 55:1530–1533, 1985.

[23] David C. Tam. Signal processing in multi-threshold neurons. In T. McKenna, J. Davis, and S.F. Zornetzer, editors, *Single neuron computation*, pages 481–501. Academic Press, 1992.