A Single-Iteration Threshold Hamming Network

I. Meilijson, E. Ruppin, and M. Sipper

***Abstract*—We analyze in detail the performance of a Hamming network classifying inputs that are distorted versions of one of its m stored memory patterns, each being a binary vector of length n. It is shown that the activation function of the memory neurons in the original Hamming network may be replaced by a simple threshold function. By judiciously determining the threshold value, 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. For m growing exponentially in n, the resulting Threshold Hamming Network correctly classifies the input pattern in a single iteration, with probability approaching 1.**

I. 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 [7]–[9], it has received renewed attention in recent years [6], [1].

The framework we analyze is an HN storing $m+1$ memory patterns $\xi^1, \xi^2, \cdots, \xi^{m+1}$, 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 Fig. 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 $(j = 1, \cdots, n)$ be

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

Manuscript received January 27, 1992; revised December 6, 1993.

The authors are with the School of Mathematical Sciences, Raymond and Beverly Sackler, Faculty of Exact Sciences, Tel-Aviv University, 69978 Tel-Aviv, Israel.

IEEE Log Number 9400112.

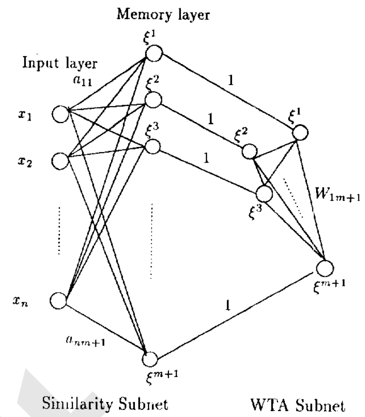

Fig. 1. A Hamming net.

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

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

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

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

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 [4], 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 HN's 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.

II. The Threshold Hamming Network

We first derive by elementary methods some sharp approximations to the binomial distribution. For a thorough review of binomial computation and approximation (including some of the material presented here), see [2].

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

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

and

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

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

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

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

Proof: Using Stirling's approximation $n! \approx (\frac{n}{e})^n\sqrt{2\pi n}$ in the binomial expression for $P(X = x_n)$, we have

$$\begin{aligned} P(X = x_n) &= \frac{n!}{x_n!(n-x_n)!}p^{x_n}(1-p)^{n-x_n} \\ &\approx \frac{(\frac{n}{e})^n\sqrt{2\pi n}p^{n\beta}(1-p)^{n(1-\beta)}}{(\frac{n\beta}{e})^{n\beta}(\frac{n(1-\beta)}{e})^{n(1-\beta)}2\pi n\sqrt{\beta(1-\beta)}} \\ &= \frac{1}{\sqrt{2\pi n\beta(1-\beta)}}\left(\left(\frac{p}{\beta}\right)^\beta\left(\frac{1-p}{1-\beta}\right)^{1-\beta}\right)^n \\ &= \frac{1}{\sqrt{2\pi n\beta(1-\beta)}} \\ &\quad \times \exp\left\{n\left[\beta\ln\left(\frac{p}{\beta}\right) + (1-\beta)\ln\left(\frac{1-p}{1-\beta}\right)\right]\right\}. \end{aligned} \tag{12}$$

Observing that

$$P(X = x+k) = P(X = x)\frac{n-x}{x+1}\frac{n-x-1}{x+2}\cdots \times \frac{n-x-k+1}{x+k}\left(\frac{p}{1-p}\right)^k \tag{13}$$

we see that

$$\begin{aligned} P(X = x_n + k) &\approx P(X = x_n)\left(\frac{p}{1-p}\right)^k \\ &\quad \times \frac{n-n\beta}{n\beta+1}\frac{n-n\beta-1}{n\beta+2}\cdots\frac{n-n\beta-k+1}{n\beta+k} \\ &\approx P(X = x_n)\left(\frac{(1-\beta)p}{\beta(1-p)}\right)^k \end{aligned} \tag{14}$$

and, by summing up the geometric series we obtain

$$\begin{aligned} P(X \geq x_n) &\approx P(X = x_n)\frac{1}{1-\frac{(1-\beta)p}{\beta(1-p)}} \\ &= \frac{1-p}{1-\frac{p}{\beta}}P(X = x_n) \\ &\approx \frac{1-p}{(1-\frac{p}{\beta})\sqrt{2\pi n\beta(1-\beta)}} \\ &\quad \times \exp\left\{-n\left[\beta\ln\frac{\beta}{p} + (1-\beta)\ln\frac{1-\beta}{1-p}\right]\right\}. \end{aligned} \tag{15}$$

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

$$m = \left(2 - \frac{1}{\beta}\right)\sqrt{2\pi n\beta(1-\beta)}\left(\ln\frac{1}{\gamma}\right)e^{nG(\beta)} \quad (16)$$

then

$$P(\max(X_1, X_2, \cdots, X_m) < x_n) \approx \gamma. \quad (17)$$

Proof:

$$\begin{aligned} P(\max X_i < x_n) &= (P(X_1 < x_n))^m \\ &= \left(1 - \frac{mP(X_1 \geq x_n)}{m}\right)^m \approx e^{-mP(X_1 \geq x_n)} \end{aligned} \quad (18)$$

converges to γ if and only if $mP(X_1 \geq x_n)$ converges to $\ln\frac{1}{\gamma}$, which holds by (11).

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

$$\beta = \alpha - \sqrt{\frac{\alpha(1-\alpha)}{n}}z_\eta - \frac{1}{2n} \quad (19)$$

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

$$\eta = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{z_\eta} e^{-x^2/2}dx. \quad (20)$$

Then, if Y and (X_i) are independent

$$\begin{aligned} &P(\max(X_1, X_2, \cdots, X_m) < Y) \\ &\quad \geq P(\max(X_1, X_2, \cdots, X_m) < x_n \leq Y) \end{aligned} \quad (21)$$

and the RHS of (21) converges to $\gamma\eta$ for m as in (16) and $n \to \infty$.

Proof: Observing that

$$\begin{aligned} &P(\max(X_1, X_2, \cdots, X_m) < x_n \leq Y) \\ &\quad = P(\max(X_1, X_2, \cdots, X_m) < x_n)P(Y \geq x_n) \end{aligned} \quad (22)$$

the proof follows using Lemma 2 on the first term of the RHS of (22), and the Central Limit Theorem on the second term.

Bearing these three lemmas, recall that the similarities $(Z_1, Z_2, \cdots, Z_m, Z_{m+1})$ are independent. If $\mathrm{Max}(Z_1, Z_2, \cdots, Z_m, Z_{m+1}) = Z_j$ 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 $\mathrm{Bin}(n, \frac{1}{2})$—distributed and Y is $\mathrm{Bin}(n, \alpha)$—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 (17) gives the probability γ that no wrong pattern declares itself the winner, while expression (20) 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 (19), let x_n be the integer closest to $n\beta$ and determine the memory capacity m using (16). 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 (21). 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 section we derive a more efficient choice of x_n (with uneven error probabilities), which yields a THN with optimal performance.

III. 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

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

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 $P(M < x) \approx \exp\left\{-m\frac{\exp\{-nG(\beta)\}}{\sqrt{2\pi n\beta(1-\beta)}(2-\frac{1}{\beta})}\right\}$. Since $G'(\beta) = \ln\frac{\beta}{(1-\beta)}$, then by Taylor expansion

$$\begin{aligned} P(M < x) &= P(M < x_0 + x - x_0) \\ &\approx \exp\left\{-m\frac{\exp\{-n[G(\beta + \frac{x-x_0}{n})]\}}{\sqrt{2\pi n\beta(1-\beta)}(2-\frac{1}{\beta})}\right\} \\ &\approx \exp\left\{-m\frac{\exp\{-nG(\beta) - (x-x_0)\ln\frac{\beta}{(1-\beta)}\}}{\sqrt{2\pi n\beta(1-\beta)}(2-\frac{1}{\beta})}\right\} \\ &= (P(M < x_0))^{\left(\frac{\beta}{1-\beta}\right)^{x_0-x}} = \gamma^{\left(\frac{\beta}{1-\beta}\right)^{x_0-x}} \end{aligned} \quad (24)$$

(in accordance with Gnedenko extreme-value distribution of type 1 [5]). Similarly,

$$\begin{aligned} P(Y < x) &= \exp\{\ln P(Y < x_0 + x - x_0)\} \\ &= \exp\left\{\ln P\left(Z < \frac{x_0 - n\alpha}{\sqrt{n\alpha(1-\alpha)}} + \frac{x - x_0}{\sqrt{n\alpha(1-\alpha)}}\right)\right\} \\ &\approx P(Y < x_0)\exp\left\{\frac{\phi(z)}{\Phi^*(z)}\frac{x - x_0}{\sqrt{n\alpha(1-\alpha)}}\right\} \\ &= (1-\eta)\exp\left\{r(z)\frac{x - x_0}{\sqrt{n\alpha(1-\alpha)}}\right\} \end{aligned} \tag{25}$$

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

$$P(M < x)P(Y \geq x) \approx \gamma^{\left(\frac{\beta}{1-\beta}\right)^{x_0 - x}}\left(1 - (1-\eta)\exp\left\{r(z)\frac{x - x_0}{\sqrt{n\alpha(1-\alpha)}}\right\}\right). \tag{26}$$

We differentiate expression (26) 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

$$\gamma = \exp\left\{-\frac{r(z)}{\sqrt{n\alpha(1-\alpha)}\ln\frac{\beta}{1-\beta}}\frac{1-\eta}{\eta}\right\}. \tag{27}$$

We now approximate

$$1 - \gamma \approx -\ln\gamma \approx \frac{r(z)}{\sqrt{n\alpha(1-\alpha)}\ln\frac{\beta}{1-\beta}}(1-\eta) \tag{28}$$

and thus the optimal proportion between the two error probabilities is

$$\delta = \frac{1-\gamma}{1-\eta} \approx \frac{r(z)}{\sqrt{n\alpha(1-\alpha)}\ln\frac{\beta}{1-\beta}}. \tag{29}$$

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

$$\gamma = 1 - \frac{\delta\epsilon}{1+\delta}, \qquad \eta = 1 - \frac{\epsilon}{(1+\delta)}. \tag{30}$$

We start with $\gamma = \eta = \sqrt{1-\epsilon}$, obtain β from (19) and δ from (23), recompute η and γ from (30) and iterate. The limiting values of β and γ in this iterative process give the maximal capacity m (by (16)) 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

$$\epsilon(m, n, \alpha) \approx \Gamma(1-\delta)\frac{1 - e^{-\delta\ln\frac{\beta}{1-\beta}}}{\delta\ln\frac{\beta}{1-\beta}}\frac{(\epsilon\delta)^\delta}{(1+\delta)^{1+\delta}}\epsilon \tag{31}$$

where

$$\Gamma(t) = \int_0^\infty x^{t-1}e^{-x}dx \tag{32}$$

is the Gamma function.

Proof:

$$\begin{aligned} P(Y \leq M) &= \sum_x P(Y \leq x)P(M = x) \\ &= \sum_x P(Y \leq x)[P(M < x+1) - P(M < x)] \\ &\approx \sum_x P(Y \leq x_0)e^{-\delta(x_0 - x)\ln\frac{\beta}{1-\beta}} \\ &\quad \times \left[(P(M < x_0))^{\left(\frac{\beta}{1-\beta}\right)^{x_0 - x - 1}} - (P(M < x_0))^{\left(\frac{\beta}{1-\beta}\right)^{x_0 - x}}\right]. \end{aligned} \tag{33}$$

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

$$\begin{aligned} P(Y \leq M) &\approx \sum_x P(Y \leq x_0)e^{-c(x_0 - x)} \\ &\quad \times\left[(P(M < x_0))^{b^{(x_0 - x - 1)}} - (P(M < x_0))^{b^{(x_0 - x)}}\right] \\ &\approx (1-\eta)\int_{-\infty}^{\infty}\left(\gamma^{b^{y-1}} - \gamma^{b^y}\right)e^{-cy}dy. \end{aligned} \tag{34}$$

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

$$\begin{aligned} &e^{-c}(1-\eta)\int_0^\infty (e^{-t} - e^{-bt})\left(\frac{t}{\ln\frac{1}{\gamma}}\right)^{\frac{-c}{\ln b}}\frac{dt}{t\ln b} \\ &\qquad = K_1\int_0^\infty (e^{-t} - e^{-bt})t^{-(1+K_2)}dt. \end{aligned} \tag{35}$$

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

$$(1-\eta)\frac{1 - e^{-c}}{c}\Gamma\left(1 - \frac{c}{\ln b}\right)\left(\ln\frac{1}{\gamma}\right)^{\frac{c}{\ln b}}. \tag{36}$$

Hence, we have

$$\begin{aligned} P(Y \leq M) &\approx (1-\eta)\frac{1 - e^{-\delta\ln\frac{\beta}{1-\beta}}}{\delta\ln\frac{\beta}{1-\beta}}\Gamma(1-\delta)\left(\ln\frac{1}{\gamma}\right)^\delta \\ &\approx \Gamma(1-\delta)\frac{1 - e^{-\delta\ln\frac{\beta}{1-\beta}}}{\delta\ln\frac{\beta}{1-\beta}}\frac{(\epsilon\delta)^\delta}{(1+\delta)^{1+\delta}}\epsilon \end{aligned} \tag{37}$$

TABLE I
PERCENTAGE OF ERROR. $n = 150, \alpha = 0.75$

m (Threshold)	100 (99)	200 (100)	400 (100)	800 (101)	1600 (102)	3200 (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 II
PERCENTAGE OF ERROR. $n = 225, \alpha = 0.75$

m (Threshold)	100 (147)	200 (147)	400 (148)	800 (149)	1600 (149)	3200 (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

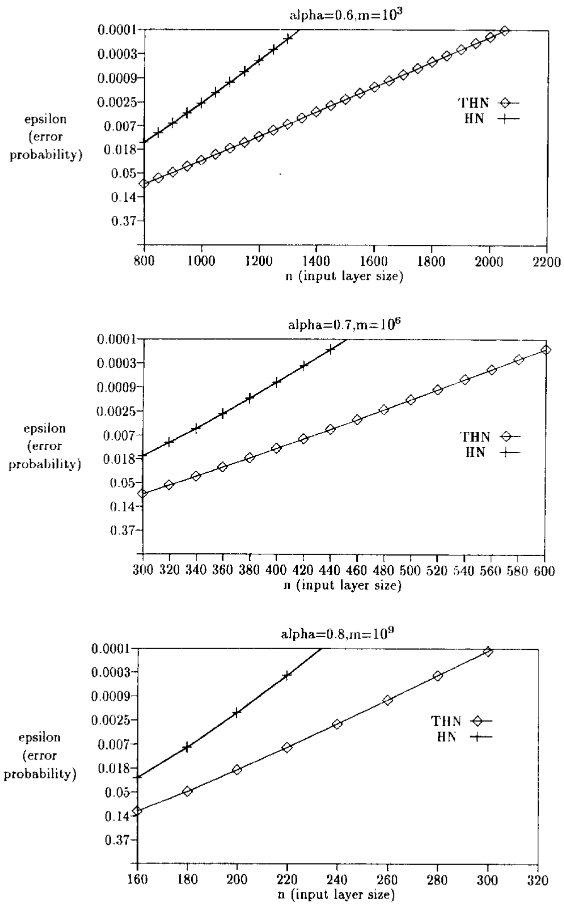

Fig. 2. Probability of error as a function of network size: three networks are depicted, displaying the performance at various values of α and m.

as claimed. Expression (31) 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 section, K is typically smaller.

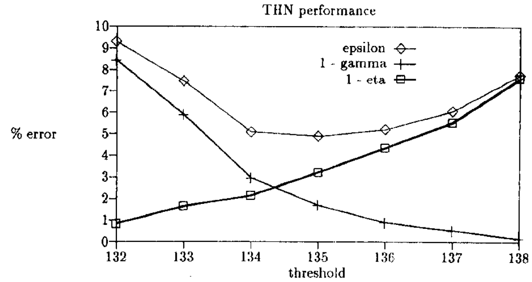

Fig. 3. Threshold sensitivity of the THN ($\alpha = 0.7$, $n = 210$, $m = 825$).

IV. NUMERICAL RESULTS

We examined the performance of the HN and the THN via simulations (of 10 000 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 I. 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 II, an increase of 50% in the input layer size n yields a THN which performs about as well as the previous HN.

Fig. 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.

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 Fig. 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.

V. CONCLUDING REMARKS

In this paper 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 [4], both the THN and the HN require $O(mn)$ connections. Hence, to perform as well as a given HN, the corresponding THN requires ≈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.

REFERENCES

[1] E. E. Baum, J. Moody, and F. Wilczek, "Internal representations for associative memory," *Bio. Cybernetics*, vol. 59, pp. 217–228, 1987.

[2] P. Diaconis and S. Zabell, "Closed form summation for classical distributions: Variations on a theme of de Moivre," *Statistical Sci.*, vol. 6, no. 3, pp. 284–302, 1991.

[3] E. Domany and H. Orland, "A maximum overlap neural network for pattern recognition," *Phys. Lett. A*, vol. 125, pp. 32–34, 1987.

[4] P. Floreen, "The convergence of Hamming memory networks," *IEEE Trans. Neural Net.*, vol. 2, pp. 449–457, 1991.

[5] M. R. Leadbetter, G. Lindgren, and H. Rootzen, *Extremes and Related Properties of Random Sequences and Processes*. Berlin, Germany: Springer-Verlag, 1983.

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

[7] K. Steinbuch, "Die lernmatrix," *Kybernetic*, vol. 1, pp. 36–45, 1961.

[8] K. Steinbuch and U. A. W. Piske, "Learning matrices and their applications," *IEEE Trans. Elect. Computers*, pp. 846–862, 1963.

[9] W.K. Taylor, "Cortico-thalamic organization and memory," *Proc. Royal Soc. London B*, vol. 159, pp. 466–478, 1964.