ON RECOGNITION ABILITY OF RANDOMIZED HOPFIELD NETWORKS

B.V.Kryzhanovsky, V.N.Koshelev, A.L.Mikaelian
Institute of Optical Neural Technologies
Russian Academy of Sciences
44/2 Vavilov Street, 117333 Moscow

A.Fonarev
Department of Engineering Science and Physics, The College of Staten Island
CUNY, SI, New York 10314

Abstract
The recognition ability of the Hopfield network is studied. The paper gives a formally strict derivation of the maximum probability for the random vector recognition error using the conventional technique of large deviation probability. Exact exchange relationships connecting the recognition error probability to the neural network dimension, the observation channel noise, and the number of stored patterns are found. The relationships are true for both large and small numbers of vectors being stored. The experimental data agree well with the results of simulation.

1. Introduction
Researchers from different fields of science have made efforts to investigate the function of the brain for many years. A lot of models have been developed to get insight into the work of neural networks. One type of such models employs the idea of associative memory [1-27]. Associative memory consists of a set of interconnected memory elements. In contrast to conventional computers with sequential access memory, the associative memory elements are accessed in parallel – as a sample vector [14].

Modeling of neural networks from the viewpoint of possible applications in the computer architecture has drawn much interest recently. The most promising model is the formal neural network of the Hopfield associative memory type (HAM). This kind of network consists of strongly interconnected threshold element called neurons [1, 2, 5-17, 24]. These models are often referred to as binary associative-memory neural networks. In the classical HAM model [2], each particular neuron receives the output signals from all other neurons of the network.

The paper considers some analytical aspects of the Hopfield model. Specifically, we are interested in such parameters of the HAM model as the memory capacity and pattern recognition efficiency. We consider the case when the neuron output is either +1 or −1. The consideration relies on the concepts of the information theory. To compute the neural network parameters, we use the Chernov-Chebyshev method of large deviations [25].

Conducted time and again, studies like that [26-29] are dedicated to refining exchange relationships \( P=P(N; p; M) \) between number of neurons \( N \), observation channel noise \( p \), number of patterns \( M \), and recognition error probability \( P \). For the most part, the exchange relationships are determined by lengthy considerations without resort to proper argumentation (see [12], [26-28] for example). Some papers (e.g. [29]) employ the theory of normal approximation and the theorem of
permutational random variables to define asymptotic properties of the neural network memory. However, either approach allows us to determine only the upper limit of the network memory, i.e. the limit where the neural network loses all its recognition abilities. Generally, the storage capacity of HAM can be evaluated as $M \sim N/\log N$. But again, we should point out that this well-known estimation is true only for neural networks of large dimension ($N \to \infty$) and for $M \gg \sqrt{N}$. On the other hand, practical use calls for the exchange relationships that are true for neural networks whose memory holds a relatively small number of patterns $M \sim O(1)$. We give three examples where such relationships are needed:

a) As the training of the net starts with storing a single pattern ($M=1$), the case of $M \sim O(1)$ helps to understand how the recognition efficiency changes during statistically adaptive learning.

b) When researchers [10, 17] attempt to realize the HAM model by optical means, the limited dynamic range of real optical media does not allow them to record a large number of patterns ($M \gg 1$): the noise growing with $M$ prohibits reading [15, 16]. The possible way out is hierarchical multiple-level neural network structures where a particular micro net can recognize just a few pattern features $m \sim O(1)$, while the macro net consisted of such micro nets recognizes the whole pattern (the micro net distinguishes characters and the macro net makes up the word). Preliminary calculations [30] show that a simple set of single-level micro nets can store and effectively recognize a much greater number of words ($M \sim e^{\eta N}$) than a conventional HAM network with the same number of neurons and the same predefined recognition error.

c) It has been found that basic elements accounting for high-level functioning of the cerebral cortex are micro nets – cortical columns of closely coupled neurons with collective properties [31-35]. This fact permits researchers to model the cortical network as a group of processing logic elements which are arrays of cortical columns [36]. For example, the network [37, 38] is modeled as a distributed grid of logically coupled cards. Paper [39] considers the dynamics of a network consisting of parametrically coupled cortical micro nets. This kind of multiple-level network has been shown to demonstrate properties similar to those of the brain: the excitation of the network can take the properties of statistic or dynamic attractor depending on conditions and input signals. It should be noted that the above-cited researches use macro nets that are built without any formal restrictions on the configuration and dimension of constituent micro nets. Such an approach allows the macro nets to have arbitrary characteristics. With dependence of trajectories in the state space of the micro net on its configuration and dimension and the small enough size and storage capacity of cortical columns ($M \sim O(1)$), we do consider the introduction of certain formal restrictions necessary.

The aim of the paper is to obtain exchange relationships $P=P(N; p; N)$ that are true for any $M$ within the range $1 \leq M \leq N$. The model we will use in our considerations is the classic feedback-free Hopfield model. The findings can be applied to other modifications of formal neural networks.

2. Randomized Hopfield model

We will consider a regular HAM model offered in [1, 2]. According to this model, the neural network is a collection of neurons $\{X_i\}$ ($1 \leq i \leq N$, where $N$ is the total number of neurons) which can take either +1 or −1. Each neuron uses a weight (or interconnection) matrix $T$ to interact with other
neurons. The interconnection matrix is symmetric about the diagonal of zero elements. The neuron is a simple adder which adds up signals from other neurons in accordance with their weights and compares the result to a predefined threshold value. The neuron takes on the value +1 if the sum is greater than the threshold; otherwise it is –1. The neuron states are changed depending on the time meter: the \( i \)-th neuron changes its state at time \( t+1 \) according to the formula:

\[
X_i(t+1) = \text{sgn} \ S_i
\]  

(1)

where \( S_i \) is the cumulative signal arriving at the input of the \( i \)-th neuron at time \( t \)

\[
S_i = \sum_{j \neq i} T_{ij} X_j(t)
\]  

(2)

The contraction function \( \text{sgn}(S) \) is chosen to have the simplest form and zero threshold:

\[
\text{sgn} \ S = \begin{cases} 
  +1, & S \geq 0 \\
  -1, & S < 0 
\end{cases}
\]  

(3)

Let us define a regular Hopfield network over a set on \( N \)-dimensional vectors \( \{ x_m \} \), \( m \in 1, ..., M \), where \( M \) is the storage capacity of the network:

\[
x_m = (x_{m1}, x_{m2}, \ldots, x_{mN})
\]  

(4)

We will consider only a randomized memory architecture, i.e. a network built around vectors \( x_m \) whose components takes either +1 or –1 with probability \( \frac{1}{2} \):

\[
x_{mi} = \begin{cases} 
  -1, & \text{Pr} \ \frac{1}{2} \\
  +1, & \text{Pr} \ \frac{1}{2} 
\end{cases}, \quad m \in 1, M \ , \ i \in 1, N
\]  

(5)

Interconnection matrix \( T \) should be initially organized so that vectors (4) are recorded in the neural network memory. Vector \( x_m \) is considered successfully memorized if it can be restored (recognized) by feeding the network with an appropriate sample vector. Let \( x_k \in \{ x_m \}, \ m \in 1, M \). Then a pattern is restored in the following way. A sample vector \( x'_k \) is fed to the network input. At time \( t=0 \) the network goes to the excited state \( X' = x'_k \) with the state of the \( i \)-th neuron being \( X'_i = x'_{ki} \) \( (i \in 1, N) \). After a few cycles of transformations (1) the network passes in the steady state \( X \), which is similar to vector \( x_k \) in the Hemming space. If vector \( X \) is exactly like \( x_k \), we can say that the recognition is performed successfully.

Different methods are used to design associative memory. One method is to form an interconnection matrix of orthogonal vectors [14, 18, 24]. Here we use a more general method of building the network that uses a set of non-orthogonal randomized vectors whose components meet condition (5). The weight matrix is chosen to be of the form:

\[
T_{ij} = \sum_{m=1}^{M} x_{mi} x_{mj} (1 - \delta_{ij})
\]  

(6)
where

\[ \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (7) \]

In further consideration we rely on the assumption that the initial state of the network always relaxes towards the stable vector. This assumption was proved by Hopfield [2] who showed that during transformations (1), “energy” \( E \) of the network

\[ E = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} T_{ij} x_i x_j \]

behaves as a monotonely decreasing function, which reaches one of its minimums (the stable states) in a particular number of time steps \( t \). Vectors \( X_m \) corresponding to these minimums are usually referred to as images that the neural network stores in its memory. Indeed, from (1) and (6) it follows that the initial feeding of the network with one of these vectors \( x_m \) puts the network in the state \( X_m = x_m \) that does not change during transformations (1). Moreover, any stable vector \( X_{m0} \in \{ X_m \} \) has an area of attraction, i.e. vectors close to \( X_{m0} \) (points in the Hemming space) that collapse to \( X_{m0} \) in the course of the network functioning (1). In other words, if a particular vector \( X' \) from the attraction area of \( X_{m0} \) arrives at the network input, the transformations (1) will result in the system relaxing to state \( X_{m0} \), and we can say that the neural network recognizes image \( X_{m0} \). With the recognition process so defined, any vector \( X' \) from the attraction area of \( X_{m0} \) can be called the distorted image of vector \( X_{m0} \), i.e. regarded as noisy vector \( X_{m0} \). A set \( \{ x_m \} \) can be considered as successfully stored by the neural network only when the form of the weight matrix is defined so that all points \( x_m \) prove to be the minimums of energy \( E \) (i.e. \( x_m \in \{ X_m \} \)). This requirement is not always met if the interconnection matrix \( T \) is of the form (6). Sometimes, a particular vector from the stored set \( \{ x_m \} \) is not a stable point and its arrival to the network input results in a wrong output vector, i.e. the error of recognition. The evaluation of this error is considered in the following paragraphs.

3. Method of large deviations

Let us analyze the recognition process. For definiteness we will consider a so-called synchronized data processing method which suggests that all components of the vector to be recognized change simultaneously, i.e. the states of all neurons change at each stage of transformations (1). Consider the case when the network is fed with vector \( x_m \) belonging to the set of vectors (4) around which, according to (6), the weight matrix \( T \) is built. Let us follow the way the \( i \)-th component \( x_{mi} \) of this vector is recognized. Erroneous recognition of a particular component evidences that vector \( x_m \) is not a stable point of the network and, therefore, can not be recognized correctly. As seen from (1) and (2), the state of the \( i \)-th neuron is equal to \( x_{mi} \) if the total input signal \( S_i \) and \( x_{mi} \) have the same sign. This means that if \( x_{mi} S_i > 0 \), the component is recognized correctly, while \( x_{mi} S_i \leq 0 \) suggests failure of recognition. This way, the error in recognizing component \( x_{mi} \) can be defined as

\[ P_l = \Pr \{ X_i(t+1) \neq x_{mi} \} = \Pr \{ S_i x_{mi} \leq 0 \} \]

(9)
or, in view of (2) and (6), as

\[ P_i = \Pr \left\{ \sum_{j=1}^{N} T_{ij} x_{mj} x_{mj} \leq 0 \right\} = \Pr \left\{ \sum_{j \neq i}^{N} \sum_{k=1}^{M} x_{ki} x_{kj} x_{mi} x_{mj} \leq 0 \right\} \quad (10) \]

As the diagonal elements of the interconnection matrix are zero and we deal with binary vectors \((x_{ij} = 1)\), expression (10) can be rewritten as

\[ P_i = \Pr \left\{ \sum_{j \neq i}^{N} \sum_{k=1}^{M} x_{ki} x_{kj} x_{mi} x_{mj} \leq 0 \right\} = \Pr \left\{ N - 1 + \sum_{j \neq i}^{N} \sum_{k \neq m}^{M} x_{ki} x_{kj} x_{mi} x_{mj} \leq 0 \right\} \quad (11) \]

Before we start to determine the error probability, note that the double summation in the right side of (11) can be represented as a sum \(L\) of independent random quantities \(q_r\) which have the same distributions and take either +1 or –1 with equal probability:

\[ q_r = \begin{cases} -1, & \frac{1}{2} \geq r \\ +1, & \frac{1}{2} < r \end{cases}, \forall r \in \overline{1,L}, \quad L = (N - 1)(M - 1) \quad (12) \]

In view of (12) we obtain

\[ P_i = \Pr \left\{ \sum_{r=1}^{L} q_r \geq N - 1 \right\} \quad (13) \]

Let us look at (13) in the terms of the problem of large deviations of the sum of random quantities. To find the probability of large deviations, we employ the Chebyshev-Chernov estimation [25]. For this purpose we adapt the derivation of this estimate to our case. Let us have a nonnegative quantity \(z\) \((z \geq 0)\). Then we obtain from (13):

\[ \Pr \left\{ \sum_{r=1}^{L} q_r \geq N - 1 \right\} \equiv \Pr \left\{ \exp \left( \sum_{r=1}^{L} q_r \right) \geq e^{z(N-1)} \right\} \leq \exp \left( -z(N-1) + z \sum_{r=1}^{L} q_r \right) = e^{-z(N-1)} \prod_{r=1}^{L} (\overline{e^{q_r}}) = e^{-z(N-1)} \left( \overline{e^{q}} \right)^L \quad (14) \]

where the line stands for averaging over the distribution of random qualities (12). Note that the upper estimate (14) is true for any \(z \geq 0\). This fact allows us to write that

\[ P_i = \Pr \left\{ \sum_{r=1}^{L} q_r \geq N - 1 \right\} \leq \min_{z \geq 0} e^{-z(N-1)} \left( \overline{e^{q}} \right)^L \quad (15) \]

This is the Chebyshev-Chernov estimate of the large deviation probability [25]. In our case, random quality \(q\) takes either +1 or –1 with probability \(\frac{1}{2}\), and, therefore, we have

\[ \overline{e^{q}} = \frac{1}{2} e^{-z} + \frac{1}{2} e^{z} \quad (16) \]
Going back to (15), we can now obtain that

\[
P_i \leq \min_{z \geq 0} e^{-z(N-1)} \left( \frac{e^z + e^{-z}}{2} \right)^L = \left[ \left[ \min_{z \geq 0} e^{-z} \left( \frac{e^z + e^{-z}}{2} \right)^{M-1} \right]^{N-1} \right]^{1/L} \tag{17}
\]

Find the minimum of the expression in the square brackets. Differentiating the expression with respect to \( z \), we obtain the condition of the extremum:

\[
(M - 1)\left( e^z - e^{-z} \right) = \left( e^z + e^{-z} \right) \tag{18}
\]

Solving (18) for \( e^z \), we find

\[
e^{2z} = \frac{M}{M - 2} \tag{19}
\]

Expression (19) defines the value of \( z \) that minimizes the estimate (17). Substituting (19) in (17) we obtain the final formula:

\[
P_i \leq \left[ \left( 1 - \frac{1}{M - 1} \right)^{M-2} \left( 1 + \frac{1}{M - 1} \right)^M \right]^{(N-1)/2} \tag{20}
\]

Concluding this paragraph, we should point out that expressions (19) and (20) make sense when \( M \geq 2 \). Particularly, when \( M = 2 \), (18) reaches the minimum at \( z \rightarrow +\infty \) when \( P_i \rightarrow 0 \). If \( M = 1 \) (the network stores only one vector), it does not make sense to tell about the failure of recognition. Note once again that this paragraph has analyzed the recognition failure in the case when the network is fed with a vector identical to one of the vectors stored in the network, which means that the recognition failure probability is connected to the dissatisfactory memory organization (weight matrix).

4. Recognition of noisy vectors

Let us consider the recognition process when the network is fed with a distorted vector. In this case erroneous recognition is largely connected with the size of the attraction area and the extent of dissimilarity of the input vector from the reference vector stored in the memory. We assume that distortions are of random nature. Let the network be fed with a vector \( x'_k \) which can be regarded as a noisy reference vector \( x_k \in \{ x_m \} \ ( m \in \overline{1, M} ) \). We can represent vector \( x'_k \) as

\[
x'_k = (\theta_1 x_{k1}, \theta_2 x_{k2}, \cdots, \theta_N x_{kN}) \tag{21}
\]

where

\[
\theta_i = \begin{cases} -1, & p \\ +1, & 1 - p \end{cases}, \quad i \in \overline{1, N} \tag{22}
\]

here \( p \) is the probability of the distortion of a single vector component. The probability of recognition failure for the \( i \)-th component of the input vector can be written in the form similar to (13):
where \( q_r \in \{+1, -1\} \) is the random quantity defined in (12). In the absence of noise (when \( \theta_i = 1 \)) expression (23) changes into (13). Repeating the technique of estimating the error probability described in the previous paragraph, we get for any \( z \geq 0 \):

\[
P_i = \Pr \left\{ \sum_{r=1}^{L} q_r \geq \sum_{j=1}^{N} \theta_j \right\} \leq \Pr \left\{ \exp \left( z \sum_{r=1}^{L} q_r \right) \geq \exp \left( z \sum_{j=1}^{N} \theta_j \right) \right\} \leq \\
\leq \exp \left( z \sum_{r=1}^{L} q_r \right) \geq \exp \left( z \sum_{j=1}^{N} \theta_j \right) = \left( \frac{e^{zq}}{e^{-z\theta}} \right)^{L} \left( \frac{e^{-z\theta}}{e^{-z\theta}} \right)^{N-1} \tag{24}
\]

where the line stands for averaging over the distribution of random quantities \( q \) and \( \theta \), and \( L = (N-1)(M-1) \) as was defined earlier. In view of evident relations

\[
e^{zq} = \frac{e^z + e^{-z}}{2}, \quad e^{-z\theta} = p e^z + (1-p)e^{-z}
\]

expression (24) takes the form:

\[
P_i \leq \min_{z \geq 0} \left[ \left( pe^z + (1-p)e^{-z} \right) \left( \frac{e^z + e^{-z}}{2} \right)^{M-1} \right] \tag{26}
\]

It is easy to show that the expression in the square brackets reaches the minimum at \( z \) meeting the following condition:

\[
Mpe^{4z} + (M - 2)(1 - 2p)e^{2z} - M(1 - p) = 0 \tag{27}
\]

Solving this equation with respect to \( e^z \) and substituting the result in (26), we get the probability of recognition failure for the \( i \)-th vector component:

\[
P_i \leq e^{-\mu N} \tag{28}
\]

where

\[
\mu = \frac{M}{2} \ln K - (M - 1) \ln \frac{K + 1}{2} - \ln(pK + 1 - p) \tag{29}
\]

\[
K = \frac{-(M - 2)(1 - 2p) + \sqrt{(M - 2)^2 + 16p(1 - p)(M - 1)}}{2pM} \tag{30}
\]

This way, the error probability \( P \) in the randomized Hopfield network (i.e. the chance of the output vector \( \mathbf{X}_m \) differing from the reference vector \( \mathbf{x}_m \) in at least one of \( N \) features) is found to be:
\[ P \equiv \Pr\{X_m \neq x_m\} = \Pr\left\{ \bigcup_{i=1}^{N} \{X_{mi} \neq x_{mi}\} \right\} \leq \sum_{i=1}^{N} \Pr\{X_{mi} \neq x_{mi}\} = \sum_{i=1}^{N} P_i \]  

(31)

In view of (28), this formula can be represented as:

\[ P \leq Ne^{-\mu N} \]  

(32)

Expression (32) sets the upper limit for the probability of erroneous recognition of the whole input vector.

5. Discussion of the results

Note that relations (28)-(32) are obtained in the asymptotic approximation \( N \gg 1 \). It is, however, very important that they are true for any number of patterns \( M > 1 \) memorized by the network. This fact allows the relations to be used for refining the HAM capacity estimates obtained in papers [12, 26-29].

Let us consider the limit case \( \mu \rightarrow 1 \) for which the authors of [12, 26-29] find the evaluation formulae of the \( M \approx N/\ln N \) type. For this purpose, we can expand \( \mu \) in the exponent (32) in small parameter \( M^{-1} \). Leaving only two first terms of the expansion, we get:

\[ \mu = \frac{(1-2p)^2}{2M} (1 + \delta) \]  

(33)

where \( \delta = o(1) \sim 2M^{-1} \). Then expression (32) takes the simple enough form:

\[ P \leq N \exp \left[ -\frac{N(1-2p)^2}{2M} (1 + \delta) \right] \]  

(34)

In the limit \( \delta \rightarrow 0 \), the expression takes the same form as the corresponding relations in [12, 26-29]. However, quantity \( \delta \) in the exponent can be neglected only if \( N\delta / 2M < 1 \) (i.e. for \( N<<M^2 \)). This way, the well-known estimation \( M \approx N/\ln N \) for the memory capacity of HAM neural networks is true only when \( N \gg 1 \) and \( M >> \sqrt{N} \) at the same time.

On the other hand, the results we obtained above allow us to correctly enough introduce such a definition as “memory capacity \( M_{\text{max}} \) given error \( P_{\text{max}} \)”, which is determined by the following formulae:

\[ M_{\text{max}} = \frac{N(1-2p)^2}{2 \ln(N / P_{\text{max}})} \]

\[ P \leq N \exp \left[ -\frac{N(1-2p)^2}{2M} \right] \]  

(35)

The capacity of the neural network memory \( M_{\text{max}} \) should be regarded in the following way: \( P \leq P_{\text{max}} \) at \( M \leq M_{\text{max}} \), i.e. the recognition failure probability does not exceed the specific level \( P_{\text{max}} \) if the number of memorized patterns \( M \) does not exceed the upper limit \( M_{\text{max}} \).
In conclusion, let us refine once again the application area of the asymptotic estimations (35) that agree with the results of earlier papers (see [12, 28, 29] and the references in them). For this purpose, we consider the expressions for the recognition failure probability given in [29] the results of which best agree with the simulation results. We can also represent the expression given in [29] as $P = N \exp(-\mu N)$. Fig.1 gives the dependence of $\mu$ on the number of memorized patterns $M$: curve 1 relies on the exact expressions (30); curve 2 – on asymptotic results of paper [29]. As seen from the figure, when the number of patterns stored in the neural network is large enough, the asymptotic estimates describe the corresponding relations correctly enough. However, for $M \leq 0.15N$ the discrepancy between the exact expression (30) and its asymptotic representation [29] becomes so large that the use of the asymptotic gives a fortiorti wrong result: the well-known asymptotic relations [26-29] for small $M$ give the understated estimation of recognition errors.

Expressions (31) and (32) give an inflated value of the upper limit of the recognition error probability. However, it is possible to obtain more exact expressions which can be compared to the experiment. For this purpose, following the calculation method given in [29], we find from [31] that

$$P = \sqrt{\frac{NM}{1 + M \ln N}} \exp\left[-\frac{N(1 - 2p)^2}{2M}\right] \quad (36)$$

As the exact formula is rather cumbersome, we give only expression (32) derived for $M \leq 2\sqrt{N}$. Fig.2 gives the error probability $P$ as a function of the number $M$ of patterns stored: the dashed line represents the results of the computer simulation experiment, the solid line is drawn in accordance with (36). The agreement grows with the increasing network dimension $N$: the theoretical and experimental curves are almost indistinguishable at $N=100$.

The work is partially funded by the Russian Foundation for Basic Research (grant No.99-01-00325).
References