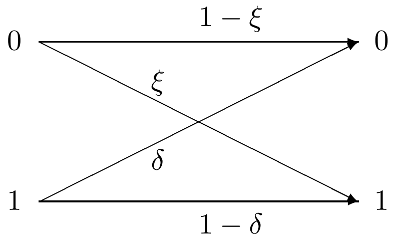
The Capacity of Matcher Neural Networks[*]

Kun-Liang Lu   Peter Z. Revesz
Department of Computer Science and Engineering
University of Nebraska–Lincoln, Lincoln, NE 68588, USA

Abstract: This paper analyzes the asymptotic capacity of Hamming Neural Networks [8, 9, 13] and Matcher Neural Networks [18, 19] in the case of unidirectional errors. It is shown that for any omission error rate, the asymptotic capacity of Matcher Neural Networks is much greater than that of Hamming Neural Networks.

1 Introduction

In many applications binary patterns may suffer omission errors (corruption of 1's to 0's) and commission errors (corruption of 0's to 1's) with different probabilities. The figure below shows the various probabilities of bit corruption. In the symmetric case $\delta = \xi$, in the asymmetric case $\delta \neq \xi$, and in the unidirectional case $\xi = 0$.



Unidirectional errors occur for example in magnetic and optical storage media, in fiber

[*] An abstract of this paper appeared in [14]. This work was supported in part by NSF grant IRI-9632871.

optical transmission, and in sign language communication [11, 12, 21, 20, 23]. Despite this range of applications, few papers consider the performance of neural networks in the unidirectional case. Among the few exceptions is the Matcher Neural Network (MNN) which was specifically designed for the unidirectional case [18, 19]. Among the more popular neural networks MNNs are closest to the Hamming Neural Networks (HNNs) [8, 9, 13].

The present paper proves that unidirectional applications are more suitable for matcher neural networks than for Hamming neural networks. The proof relies on arguments from information theory to show that in the unidirectional case for any δ MNN can store and recall correctly more random vectors than HNN can.

This result explains previous computer experiments that compared MNN, HNN and other neural networks in the nearly unidirectional environment of the CyberGlove system [22] when it was used for hand sign recognition [20].[1] There MNNs were used to recognize hand signs from a dictionary of 395 root signs with 96% accuracy while HNNs performed much poorer with about 33% accuracy.

Our study complements various capacity analyses for other neural networks. Hopfield [7] suggested by simulation that the relative capacity in Hopfield associative memory is $m = 0.15n$ where m is the number of patterns learned correctly and n is the size of the memory. McEliece et al. [16] proved that the absolute capacity in Hopfield associative memory is $m = \frac{n}{4\log(n)}$. Amit et al. [2] proved that for the spin-glass model $m = 0.14n$. Amari and Yanai [1] present a unified approach to the analysis of various architectures of associative memory models: the cascade model, the cyclic model, BAM, and the autocorrelation model. Chou [**?**] analyzed the capacity of Sparse Distributed Memories which were introduced in Kanerva [10].

This paper is organized as follows. Section 2 lists basic definitions. Section 3 presents the main statistical analysis for MNNs. Section 4 generalizes the analysis in Section 3 for the case when each vector has several parts that have different omission error probabilities. Section 5 compares the capacity of MNNs with that of HNNs. Finally, Section 6 lists some open problems.

[1] The CyberGlove system is nearly unidirectional because various muscle movements or elementary features of signs are more likely to be omitted than extra ones introduced.

2 Basic Definitions

Let V_n denote the binary n-cube $\{0,1\}^n$. We may think that each vector X in V_n is a feature list. That is it describes an object by its features with the ith feature being present in the object if and only if the ith bit in X is a 1.

The neural networks considered in this paper memorize a subset $X_1, \ldots, X_M$ of V_n in an autoassociative way. By *autoassociative* we mean that if a corrupted version of any X_i is input to the network, then the network should give as output X_i. Of course, how well a network performs depends to a large extent on the degree of corruption in the input.

There are several ways to measure the difference between two feature lists. Below we define formally both the Hamming distance and the asymmetric distance.

Definition 2.1 Let V_n denote the binary n-cube $\{0,1\}^n$, and let $X, Y \in V_n$. We call the *Asymmetric Distance* of X and Y

$$h(X,Y) \stackrel{def}{=} card(\{i \;:\; (x_i = 1) \wedge (y_i = 0), 1 \leq i \leq n\}).$$

the *Hamming Distance*

$$d_h(X,Y) \stackrel{def}{=} h(X,Y) + h(Y,X)$$

□

We say that Y *matches* X if $h(X,Y) = 0$. In that case whenever a feature is present in X then it is also present in Y.

Remark: The Hamming distance is the square of the Euclidean distance, which is used in nearest neighbor pattern classification [4]. Hence a HNN corresponds to a nearest neighbor classifier where each category contains only one pattern.

Definition 2.2 Let $v(X)$ be the number of 1's in the vector $X \in V_n$. □

Example 2.1 Let $X = (0,0,1,1,1)$ and $Y = (0,1,0,1,0)$. Then $h(X,Y) = card(\{3,5\}) = 2$ because the third and the fifth digits are 1s in X and 0s in Y. Also, $h(Y,X) = card(\{2\}) = 1$ because only the second digit is 1 in Y and 0 in X. We also have $d_h(X,Y) = h(X,Y) + h(Y,X) = 3$. Finally, $v(X) = 3$ and $v(Y) = 2$. □

This paper analyzes the capacity of Matcher Neural Networks [18, 19]. The recall algorithm of MNN is the following:[2]

(1) Find those memorized vectors that match the input.
(2) Select from the matching vectors those with the smallest asymmetric distance from the input.
(3) If several vectors remain, then select randomly any one of them.

For example, suppose that the vector $X_1 = (1,0,0,0,0)$ and $X_2 = (1,0,1,1,1)$ are the stored vectors in a MNN and the input vector is $X_0 = (1,0,0,1,0)$. Then the output vector is X_2 because in step (1) out of the two stored vectors only X_2 will be found to match all the 1s in X_0, because $h(X_0, X_1) = 1$ but $h(X_0, X_2) = 0$.

The recall algorithm of HNN is similar to that of MNN but step (1) is skipped and in step (2) the Hamming distance measure is used.

For example, suppose now that we have the same situation as in the previous example but we use a HNN instead of a MNN. Then the output vector will be X_1 because $d_h(X_0, X_1) = 1$ while $d_h(X_0, X_2) = 2$.

The capacity of a neural network is intuitively the number of vectors in V_n that it can correctly memorize. We define the capacity formally as follows.

Definition 2.3 Let ϵ be any fixed number between 0 and 1 and let $\mathcal{N}$ be any autoassociative neural network that memorizes vectors of length n that have been generated at random. The *capacity* of $\mathcal{N}$ with respect to ϵ, n, δ and ξ is the maximum number of memorized vectors $X_1, \ldots, X_M$ such that for any X_i for $1 \leq i \leq M$ if vector X is within a fraction of δ omission and ξ commission error from X_i, and the output of the network is Y, then the probability that $X_i \neq Y$ is less than ϵ. □

Definition 2.4 Let δ be the fraction omission and ξ be the fraction commission error allowed during recall. Let for each integer $n \geq 1$ $\mathcal{N}_n$ be the set of neural networks that memorize vectors of length n. Let $\{\epsilon_n, n \geq 1\}$ be a sequence of positive numbers with $\lim_{n \to \infty} \epsilon_n = 0$ and $\{M_n, n \geq 1\}$ a sequence of integers. If for each n, M_n is the maximum capacity of a network in $\mathcal{N}_n$ with respect to ϵ_n and n and δ and ξ, then we define the *asymptotic capacity* with respect to δ, ξ, $\{\epsilon_n, n \geq 1\}$ to be $\{M_n, n \geq 1\}$. □

[2] In this paper we omit describing the architecture of the MNN and the adaptive learning procedure (see [15]) because these are not needed for the main proof of the paper.

Absolute capacity is asymptotic capacity with $\epsilon_n = 0$ for each $n \geq 1$. *Relative capacity* is asymptotic capacity with $\delta = \xi = 0$. *Symmetric capacity* is asymptotic capacity with $\delta = \xi$. *Unidirectional capacity* is asymptotic capacity with $\xi = 0$.

It is intuitive that the asymptotic capacity is a monotone increasing sequence for all reasonable sets of neural networks. Therefore, what we will be interested in seeing is how fast this sequence increases. The faster the rate of increase the better the memory of a set of neural networks. This rate of increase is sometimes called the exponential rate and is defined as follows.

Definition 2.5 For any fixed set of neural networks and $\{\epsilon_n, n \geq 1\}$ we define the *exponential rate* function to be

$$\gamma(\delta) \overset{def}{=} \limsup_{n\to\infty} \ \frac{1}{n} \log_2(M_n)$$

where $\{M_n, n \geq 1\}$ is the asymptotic capacity with respect to δ, $\xi = 0$ and $\{\epsilon_n, n \geq 1\}$. □

We will see in this paper that the exponential rate for Matcher Neural Networks is much higher than for Hamming neural networks. Hence for sufficiently large vectors to be memorized the former provides a more efficient memory than the latter.

For the technical analysis in the paper we will also need to define various *information rate* functions usually used for channel models [17]. The information rate functions provide approximations for the exponential rate function and vice versa.

If $\xi = \delta$ we talk about a binary symmetric channel (BSC). The information rate of BSCs is

$$I_h(\delta) \overset{def}{=} 1 - h_2(\delta)$$

where

$$h_2(\delta) \overset{def}{=} -\delta \log_2(\delta) - (1-\delta)\log_2(1-\delta)$$

is Shannon's entropy function. If $\xi = 0$ we talk about an ideal binary asymmetric channel (IBAC or Z-channel). The information rate of IBAC is

$$I_a(\delta) \overset{def}{=} 1 + \frac{\delta}{2}\log_2(\delta) - \frac{1+\delta}{2}\log_2(1+\delta).$$

Note that by Shannon's Information Theorem, I_a is an upper bound of the exponential rate of the MNN when the input vectors have only omission errors.

Furthermore, we define two additional information rates I_m and I_r. For the first $I_m(\delta) = 1 - 2\delta$. This is the theoretical maximum information rate that can be achieved. This follows from [5].

For the second let $\beta(\delta) \overset{def}{=} \max\{\delta, (3 + \delta - \sqrt{\delta^2 + 6\delta + 1}\ )/4\}$. Then

$$I_r(\delta) \overset{def}{=} I_h(\beta(\delta)) + 1 - (1 - \beta(\delta) + \delta)h_2\left(\frac{\delta}{1 - \beta(\delta) + \delta}\right).$$

3 A Capacity Analysis of Matcher Neural Networks

Suppose that the stored patterns are selected at random and that the input patterns are different from the stored patterns only with d omission bits when the stored pattern has at least d number of 1's and with k omission bits when the stored pattern has k $(k < d)$ number of 1's. Under the above assumption, we say that the input pattern has at most d omission bits.

Lemma 3.1 For MNN, let M_n be the number of stored patterns selected at random. Without loss of generality, we assume that X_1 is the pattern to be retrieved. Then the probability of retrieving a wrong pattern Y given an input with at most d bit omissions is bounded by

$$P(Y \neq X_1) \leq \frac{(M_n - 1)}{2^{2n}} \sum_{k=0}^{n} \binom{n}{k} \sum_{i=0}^{\min(d,k)} \binom{n-k+d}{i}.$$

Proof. When a wrong pattern $Y = X_j$ $(j \neq 1)$ is retrieved, this pattern X_j is at least as close to the input pattern X_0 as X_1 is, where the input pattern X_0 is a vector satisfying the conditions of $h(X_0, X_1) = 0$ and $h(X_1, X_0) = \min(d, v(X_1))$. If $v(X_1) = k \geq d$ and the input X_0 has d omission errors, then $v(X_0) = k - d$. In this case the vector X_j has a bit 1 if the corresponding bit vector X_0 is 1, and X_j has at most d more 1's among the $n - (k - d)$ other bits. (If there were more than d such 1's in X_j then X_j would be more than d distance from X_0. Hence X_1 would be closer to X_0 than X_j.) Similarly, if $v(X_1) = k < d$ then X_j has at most k more 1's than X_0 has. The probability $P(Y \neq X_1)$ is bounded by the sum of the probabilities for $2 \leq j \leq M_n$

$$P(Y = X_j) \leq \sum_{k=0}^{n} P(v(X_1) = k) \cdot P(h(X_j, X_0) \leq h(X_1, X_0) \textit{ and } h(X_0, X_j) = 0 | v(X_1) = k),$$

hence by the foregoing argument,

$$P(Y = X_j) \leq \sum_{k=0}^{n} P(v(X_1) = k) \cdot P(h(X_j, X_0) \leq \min(d, k) \textit{ and } h(X_0, X_j) = 0 | v(X_1) = k),$$

which implies the inequality in the Lemma. □

Theorem 3.1 For MNN, let M_n be the number of stored patterns selected at random. Without loss of generality, we assume that X_1 is the pattern to be retrieved. If δ is any number such that

$$\limsup_{n \to \infty} \frac{1}{n} \log_2(M_n) < I_r(\delta),$$

then for all $\epsilon > 0$, there exists an n_0 such that for all $n \geq n_0$, and for all input X with $h(X_1, X) = \min(\delta n, v(X_1))$ and $h(X, X_1) = 0$,

$$P(Y \neq X_1) < \epsilon,$$

where Y is the retrieved pattern for input X from $\{X_1, \cdots, X_{M_n}\}$.

Proof. See Appendix. □

Theorem 3.2 For MNN, let M_n be the number of stored patterns selected at random. Without loss of generality, we assume that X_1 is the pattern to be retrieved. If δ is any number such that

$$\limsup_{n \to \infty} \frac{1}{n} \log_2(M_n) > I_r(\delta)$$

then for all $\epsilon > 0$, there exists an n_0 depending on ϵ such that for all $n \geq n_0$ and for the input X with $h(X_1, X) = \min(\delta n, v(X_1))$ and $h(X, X_1) = 0$,

$$P(Y \neq X_1) > 1 - \epsilon,$$

where Y is a retrieved pattern from $\{X_1, \cdots, X_n\}$.

Proof. See Appendix. □

Theorems 3.1 and 3.2 together imply that for MNN the exponential rate is equivalent to the information rate for large n. This can be seen as follows.

Assume that there is some δ such that $\gamma(\delta) > I_r(\delta)$. Therefore for the capacity sequence $\{M_n, n \geq 1\}$ the supper limit is larger than $I_r(\delta)$. Therefore by Theorem 3.2 we cannot retrieve correctly, which is a contradiction because by Definition 2.4 for the capacity sequence we can retrieve correctly.

Now assume that there is some δ such that $\gamma(\delta) < I_r(\delta)$. Let $\{M_n, n \geq 1\}$ be the capacity sequence. Then define a new sequence each of whose elements is $2^{n\frac{I_r(\delta)+\gamma(\delta)}{2}}$. For this new sequence the supper limit is also less than $I_r(\delta)$. Therefore, by Theorem 3.1 we can retrieve correctly as many vectors as the new sequence shows. This contradicts the maximum assumption within Definition 2.4.

4 Multiple Omission Errors

In many practical applications the probability of making an omission error in the input will be slightly different for different bits of a large pattern. For example, Revesz & Veera [20] used MNN for sign language translation where the input binary pattern was generated by a CyberGlove device. The CyberGlove gave a one-bit output with "1" for each bent and "0" for each straight joint in the hand. The experiments show that the omission errors are much more frequent for the bits describing the thumb finger joints than for the other bits. Some other applications may also use simultaneously several different types of sensors, for example tactile, visual, auditory sensors with potentially large differences in reliabilities. In this section, we analyze the capacity of MNN considering two different groups of bits within the input pattern, with each group having a different omission error.

Suppose that the input vector $X \in V_n$ can be broken down into 2 subvectors

$$X = (X^{(1)}, X^{(2)})$$

where the length of $X^{(i)}$ is n_i, for $i = 1, 2$ and $n = n_1 + n_2$. Let $\lambda_i = n_i/n$ denote the proportional length of the two subvectors. (Here each subvector $X^{(i)}$ describes a group of bits as explained earlier.) We assume that the different subvectors have different probabilities of omission errors. Let δ_i be the probability of omission error for the subvector i where $i = 1, 2$.

Definition 4.1 The *multiple information rate* I_R is defined by

$$I_R(\delta_1, \delta_2) = \lambda_1 I_r(\delta_1) + \lambda_2 I_r(\delta_2)$$

Lemma 4.1 Let M_n be the number of stored patterns selected at random and learned by a MNN. Without loss of generality, we assume that X_1 is a learned pattern to be retrieved. Then the probability of retrieving a wrong pattern given an input with at most $d_i = n_i\delta_i$ bit omissions distance from $X^{(i)}$ is bounded by

$$P(Y \neq X_1) \leq \frac{(M_n - 1)}{2^{2n}} \sum_{k_1=0}^{n_1} \sum_{k_2=0}^{n_2} \left\{ \prod_{i=1}^{2} \binom{n_i}{k_i} \left[\sum_{j=0}^{\min(d_i, k_i)} \binom{n_i - k_i + d_i}{j} \right] \right\}$$

□

Theorem 4.1 For MNN, let $M_n, n \geq 1$ be the number of stored patterns selected at random. Without loss of generality, we assume that X_1 is the pattern to be retrieved. If δ_1 and δ_2 are any numbers such that

$$\limsup_{n \to \infty} \frac{1}{n} \log_2(M_n) < I_R(\delta_1, \delta_2),$$

then for all $\epsilon > 0$, there exists an n_0 such that for all $n \geq n_0$, and for all input X with $h(X_1, X) = \min(\bar{\delta} n, v(X_1))$ and $h(X, X_1) = 0$,

$$P(Y \neq X_1) < \epsilon,$$

where $\bar{\delta} = \lambda_1 \delta_1 + \lambda_2 \delta_2$ and Y is the retrieved pattern for input X from $\{X_1, \cdots, X_{M_n}\}$.

Proof. Similar to the proof of Theorem 3.1. □

Theorem 4.2 For MNN, let $M_n, n \geq 1$ be the number of stored patterns selected at random. Without loss of generality, we assume that X_1 is the pattern to be retrieved. If δ_1 and δ_2 are any number such that

$$\limsup_{n \to \infty} \frac{1}{n} \log_2(M_n) > I_R(\delta_1, \delta_2)$$

then for all $\epsilon > 0$, there exists an n_0 depending on ϵ such that for all $n \geq n_0$ and for the input X with $h(X_1, X) = \min(\bar{\delta} n, v(X_1))$ and $h(X, X_1) = 0$,

$$P(Y \neq X_1) > 1 - \epsilon,$$

where $\bar{\delta} = \lambda_1 \delta_1 + \lambda_2 \delta_2$ and Y is a retrieved pattern from $\{X_1, \cdots, X_n\}$.

Proof. Similar to the proof of Theorem 3.2. □

Consider the special case in which there are multiple omission errors. Suppose that the two subvectors have equal length $n_1 = n_2 = n/2$, and $\delta_1 = 0$ and $\delta_2 = 2\delta$, where δ is the percent omission of the single omission error case discussed in Section 3. To compare the single omission case and the multiple omission case, we assume that $\bar{\delta} = \delta$. The two information rates are shown in the Figure 1. The information rate in the multiple omission case is higher then that in the single omission case.

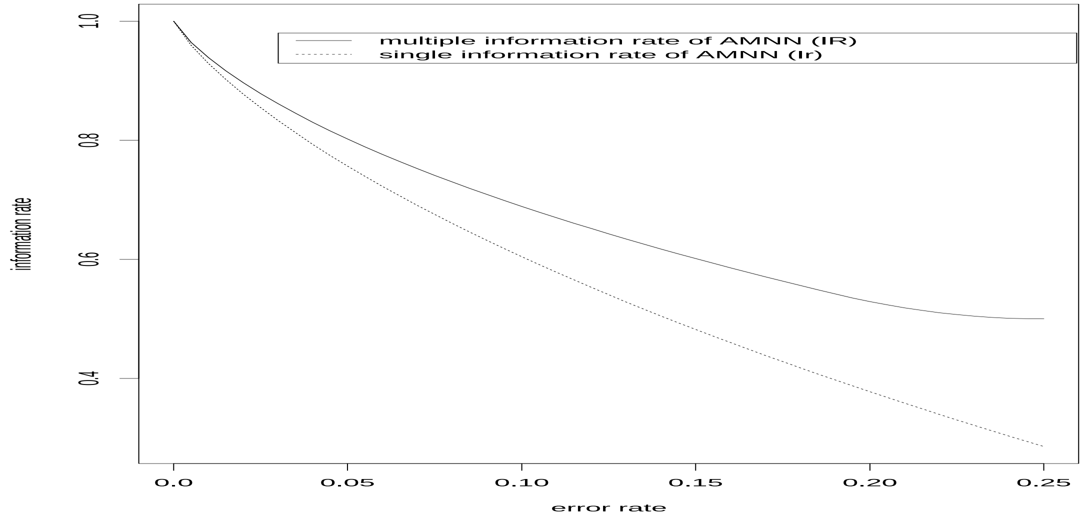


Figure 1: The information rates for multiple omission errors

As a more concrete example, lets assume that $n = 100$ and $\delta = 0.1$. Then in both the single and the multiple omission case we expect 10 bits to be in error. But, whereas in the single omission case the errors can occur anywhere, in the multiple omission case they can occur only among the last 50 bits. Figure 1 reveals that despite some superficial similarity, the second case is really better.

5 Comparison with Hamming Neural Networks

The Hamming Neural Network (HNN) was proposed by Jackel et al. [8, 9]. It consists of two parts. The first part calculates matching scores, and the second part is the MAXNET, which picks the maximum of all the matching scores. The matching score computed by HNNs is the Hamming distance between the input vector and the stored vectors. The capacity of the Hamming networks can be defined in the same way as for MNNs.

Lemma 5.1 For HNN, let M_n be the number of stored patterns selected at random. Without loss of generality, we assume that X_1 is the pattern to be retrieved. Then the probability of retrieving a wrong pattern Y given an input with at most d bit errors is bounded by

$$P(Y \neq X_1) \leq \frac{(M_n - 1)}{2^n} \sum_{k=0}^{d} \binom{n}{k}$$

Proof. When a wrong pattern $Y = X_j$ $(j \neq 1)$ is retrieved, this pattern X_j is at least as close to the input pattern X_0 as X_1 is, Then $d_h(X_j, X_0) \leq d_h(X_1, X_0) \leq d$. The probability $P(Y \neq X_1)$ is bounded by the sum of the probabilities for $2 \leq j \leq M_n$

$$\begin{aligned} P(Y = X_j | X_0, X_1) &= P(d_h(X_j, X_0) \leq d_h(X_1, X_0) | X_0, X_1) \\ &\leq P(d_h(X_j, X_0) \leq d | X_0, X_1) \\ &= \frac{1}{2^n} \sum_{k=0}^{d} \binom{n}{k}. \end{aligned}$$

then $P(Y = X_j) \leq \frac{1}{2^n} \sum_{k=0}^{d} \binom{n}{k}$. Thus,

$$P(Y \neq X_1) \leq \sum_{j=2}^{M_n} P(Y = X_j) \leq \frac{(M_n - 1)}{2^n} \sum_{k=0}^{d} \binom{n}{k}$$

which implies the inequality in the Lemma. □

Theorem 5.1 For HNNs, let M_n be the number of stored patterns selected at random. Without loss of generality, we assume that X_1 is the pattern to be retrieved. If δ is any number such that $d = \delta n$ is an integer and

$$\limsup_{n \to \infty} \frac{1}{n} \log_2(M_n) < I_h(\delta),$$

then for all $\epsilon > 0$, there exists an n_0 such that for all $n \geq n_0$, and for all input X_0 with $d_h(X_1, X_0) = d$,

$$P(Y \neq X_1) < \epsilon,$$

where $d = \delta n$ is an integer and Y is the retrieved pattern for input X_0 from $\{X_1, \cdots, X_{M_n}\}$. □

Proof. Definition 2.5 implies that $M_n \approx 2^{n\gamma(\delta)}$ for large n. We have $\gamma(\delta) < I_h(\delta)$ by the condition of the theorem. Hence, for large n

$$M_n < 2^{n\left[\gamma(\delta) + \frac{I_h(\delta) - \gamma(\delta)}{2}\right]}$$

By the bound of the binomial coefficient (see [6], page 530),

$$\sum_{k=0}^{d} \binom{n}{k} 2^{-n} \leq 2^{-nI_h(\delta)}. \tag{1}$$

By the Lemma 5.1,

$$P(Y \neq X_1) \leq \frac{(M_n - 1)}{2^n} \sum_{k=0}^{d} \binom{n}{k} \leq 2^{n\left[\gamma(\delta) + \frac{I_h(\delta) - \gamma(\delta)}{2}\right]} 2^{-nI_h(\delta)} = 2^{-n\frac{I_h(\delta) - \gamma(\delta)}{2}}. \tag{2}$$

Thus, for large $n > n_0$, $P(Y \neq X_1) < \epsilon$. □

Theorem 5.2 For HNNs, let M_n be the number of stored patterns selected at random. Without loss of generality, we assume that X_1 is the pattern to be retrieved. If δ is any number such that

$$\limsup_{n \to \infty} \frac{1}{n} \log_2(M_n) > I_h(\delta)$$

then for all $\epsilon > 0$, there exists an n_0 depending on ϵ such that for all $n \geq n_0$ and for the input X_0 with $d_h(X_1, X_0) = d$,

$$P(Y \neq X_1) > 1 - \epsilon,$$

where $d = \delta n$ is an integer and Y is a retrieved pattern from $\{X_1, \cdots, X_n\}$.

Proof. See Appendix. □

Theorems 5.1 and 5.2 together imply that for HNNs the exponential rate is equivalent to the information rate for large n. This can be seen as follows.

Assume that there is some δ such that $\gamma(\delta) > I_h(\delta)$. Therefore for the capacity sequence $\{M_n, n \geq 1\}$ the upper limit is larger than $I_h(\delta)$. Therefore by Theorem 3.2 we cannot retrieve correctly, which is a contradiction because by Definition 2.4 for the capacity sequence we can retrieve correctly.

Now assume that there is some δ such that $\gamma(\delta) < I_h(\delta)$. Let $\{M_n, n \geq 1\}$ be the capacity sequence. Then define a new sequence each of whose elements is $2^{n\frac{I_h(\delta)+\gamma(\delta)}{2}}$. For this new sequence the supper limit is also less than $I_h(\delta)$. Therefore, by Theorem 5.1 we can retrieve correctly as many vectors as the new sequence shows. This contradicts the maximum assumption within Definition 2.4.

Figure 2 summarizes the relation among the various information rate functions.

For comparison with HNNs, we have the following theorems:

Theorem 5.3 For $0 < \delta < 0.5$,

$$I_h(\delta) < I_r(\delta) < 2I_h(\delta).$$

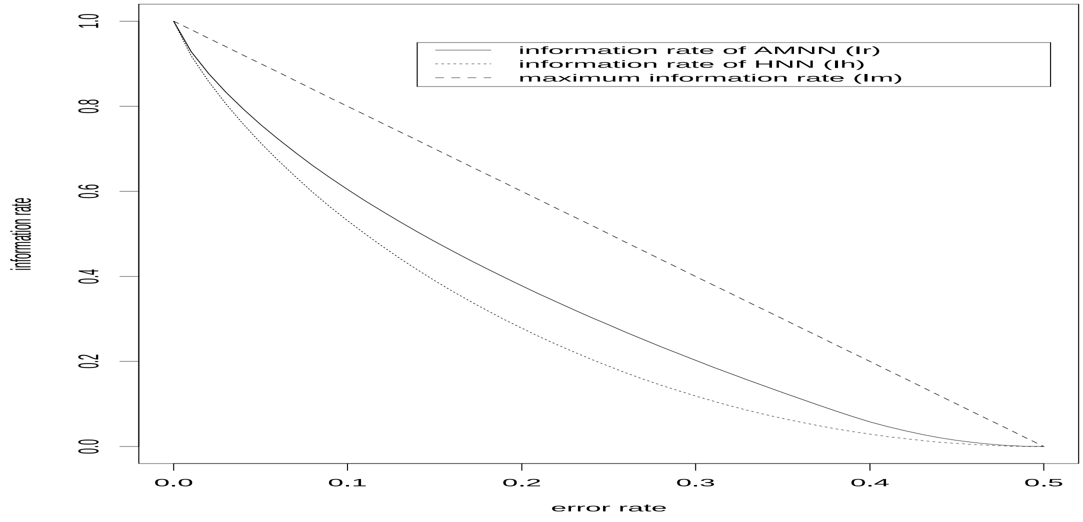


Figure 2: The information rates

□

Proof. It is easy to show that the function $g_1(\delta) \overset{def}{=} I_r(\delta) - I_h(\delta)$ and the function $g_2(\delta) \overset{def}{=} 2I_h(\delta) - I_r(\delta)$ are positive in the interval $(0, 1)$. Then the theorem is proved. □

The above theorem shows that MNNs have smaller error probability than HNNs, when they have the same number of stored patterns. The following plot shows a comparison of MNN and HNN. In the plot, the x-axis is the number of bits for memories of size $N = 2^x$ bits, i.e. $x = \log_2(N)$, and the y-axis is the error rate δ, which is the percentage of the omission bits in the vector of n-bits. From the Figure 3, it can be seen that the MNN uses the memory more efficiently.

As an example, suppose that the length of the stored vector is $n = 1000$, and the maximum percent of omission errors is 40%. The MNN can memorize $M = 2^{55}$ (i.e. $36 \times 10^{15} Mb$) vectors without any significant errors, while the HNNs can memorize only $M = 2^{26}$ (i.e. about $67Mb$) vectors.

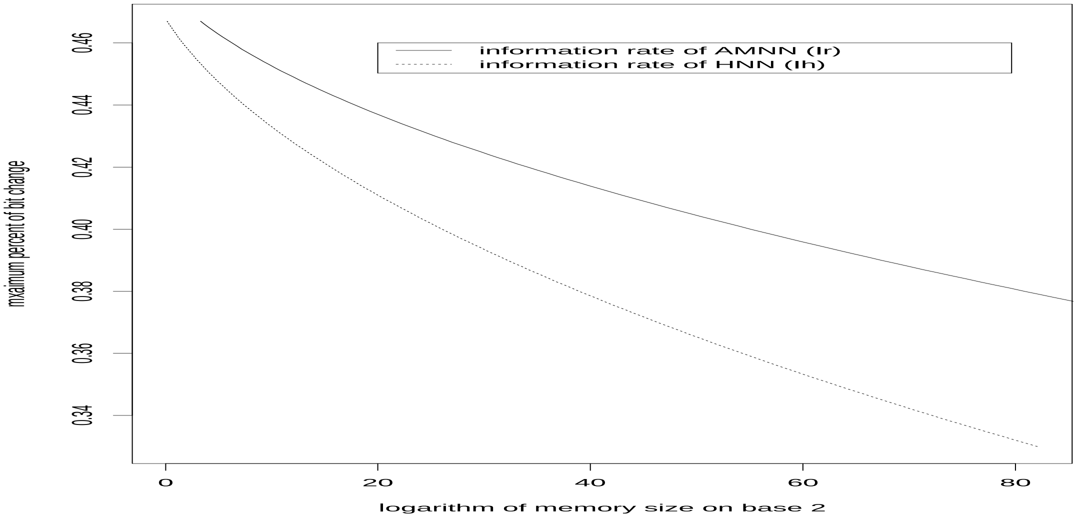


Figure 3: The error rates when $n = 1000$

6 Conclusion

This paper considered the case of unidirectional errors. This case occurs in many applications (e.g. [11, 12, 21, 20, 23]), but it is largely neglected in the neural networks literature.

This paper supports the idea that the unidirectional and the symmetric cases should be handled by different neural networks. It remains an interesting research issue to design more neural networks for the unidirectional case and to compare their capacity with that of Matcher Neural Networks.

There is also work to be done on the asymmetric case. We are currently extending our analysis to the asymmetric case. In the extension, step (1) of the MNN recall algorithm would be modified. Instead of requiring exact matches, imperfect matches would be also allowed. That is, if X_0 is the input, then for some fixed constant c each stored vector X_i will be selected for which $h(X_0, X_i) \leq c$. We also consider extending the analysis from binary to analog data.

References

[1] Amari, S. and Yanai, H. Statistical Neurodynamics of Various Types of Associative Nets, In: Hassoun (ed.), *Associative Neural Memories*, pp. 169–183, (Oxford University Press, 1993).

[2] Amit, D. J., Gutfreund, H., and Sompolinsky, H. Spin-glass Model of Neural Networks, *Phys. Rev.*, **A32**, 1007-1018, (1985).

[3] Chou Chou, P. A. The Capacity of the Kanerva Associative Memory, *IEEE Trans. Info. Theory.* **35**(2), 281-298, (1989).

[4] Cover, T. M. and Hart, P. E. Nearest Neighbor Pattern Classification, *IEEE Trans. Info. Theory.* **13**(1), 21-27, (1967).

[5] Fang, G. *Binary Block Code for Correcting Asymmetric or Unidirectional Errors*, Ph.D. thesis, Eindhoven University of Technology, (1993).

[6] Gallager, R. G. *Information Theory and Reliable Communication*, (John Wiley and Sons, 1968).

[7] Hopfield, J. J. Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *Proc. Nat. Acad. Sci. U.S.A.*, **79**, 2445-2458, (1982).

[8] Jackel, L. D. Howard, R. E., Graf, H. P., Straughn, B., Denker, J. S. Artificial Neural Networks for Computing, *J. Vac. Sci. Technol.*, B4, 61-63, (1986).

[9] Jackel, L. D., Graf, H. P., Howard, R. E. Electronic Neural Network Chip, *Appl. Opt.* 26, 5077-5080, (1987).

[10] Kanerva, P. *Sparse Distributed Memory*, (MIT Press, 1988).

[11] Knuth, D.E. Efficient Balanced Codes. *IEEE Transactions on Information Theory*, vol. 32, pp. 51-53, 1986.

[12] Leiss, E.L. Data Integrity in Digital Optical Disks. *IEEE Transactions on Computers*, vol. 33, pp. 818-827, 1984.

[13] Lippmann, R. P. An Introduction to Computing with Neural Nets, *IEEE ASSP Magazine*, 4, 4-22, (1987).

[14] Lu, K-L. and Revesz, P. Z. The Capacity of Matcher Neural Networks. *Proc. Artificial Neural Networks in Engineering*, (St. Louis, MI, 1995).

[15] Lu, K-L. *The Capacity of Matcher Neural Networks.* M.S. Thesis, University of Nebraska-Lincoln, 1996.

[16] McEliece, R. J., Posner, E. R., and Venkatesh, S. S. The Capacity of the Hopfield Associative Memory, *IEEE Trans. Info. Theory.* **IT-33**, 461-482, (1987).

[17] MacWilliams, F. J. and Sloane, N. J. A. *The Theory of Error-Correcting Codes*, (Amsterdam: North-Holland, 1977).

[18] Revesz, P. Z. Matcher Neural Networks, *1st International Joint Conference on Neural Networks*, vol. 1, 767-772, (Washington D.C, 1989).

[19] Revesz, P. Z. Functional Interpretations of Neocortical Modules, *2nd International Joint Conference on Neural Networks*, vol. 2, 509-514, (San Diego, CA, 1990).

[20] Revesz, P. Z. and Veera, R. R. A Sign-To-Speech Translation System Using Matcher Neural

Networks, *Proc. Artificial Neural Networks in Engineering*, (St. Louis, MI, 1993)..

[21] Takasaki, Y. et al. Optical Pulse Formats for Optic Digital Communications. *IEEE Transactions on Communications*, vol. 24, pp. 404-413, 1976.

[22] Virtual Technologies Inc., *CyberGlove^{TM} User's manual,* 1992.

[23] Widmer, A.X. and Franaszek. A Debalanced, Partitioned Block, 8B/10B Transmission Code. *IBM Journal of Research and Development*, vol. 27, no. 5, pp. 440-451, 1983.

A Appendix

Proof of Theorem 3.1. Let $d = \delta n$. Definition 2.5 implies that $M_n \approx 2^{n\gamma(\delta)}$ for large n. We have $\gamma(\delta) < I_r(\delta)$ by the condition of the theorem. Hence, for large n

$$M_n < 2^{n\left[\gamma(\delta)+\frac{I_r(\delta)-\gamma(\delta)}{2}\right]}$$

We consider the case of $v(X_1) \geq d$, first. For $k \geq d$, $\binom{n}{k}\binom{n-k+d}{d}$ is maximum when $k = k_1 \overset{def}{=} \beta(\delta)n$. Then by the bound of the binomial coefficient (Gallager (1968), page 530),

$$\begin{aligned} \sum_{k=d}^{n}\binom{n}{k}2^{-n}\sum_{i=0}^{d}\binom{n-k+d}{i}2^{-n} &\leq n(d+1)\binom{n}{k_1}2^{-n}\binom{n-k_1+d}{d}2^{-n} \\ &\leq n(d+1)2^{-n\left[I_h(\beta(\delta))+1-(1-\beta(\delta)+\delta)h_2(\frac{\delta}{1-\beta(\delta)+\delta})\right]} \\ &\leq n(d+1)2^{-nI_r(\delta)}. \end{aligned}$$

Now, we consider the case of $v(X_1) < d$.

$$\sum_{k=0}^{d-1}\binom{n}{k}2^{-n}\sum_{i=0}^{k}\binom{n}{i}2^{-n} \leq \left(\sum_{k=0}^{d}\binom{n}{k}2^{-n}\right)^2 \leq 2^{-2nI_h(\delta)}.$$

By Lemma 3.1, we have from the above inequality,

$$\begin{aligned} P(Y \neq X_1) &\leq M_n\left[n(d+1)2^{-nI_r(\delta)} + 2^{-2nI_h(\delta)}\right] \\ &\leq 2^{n\left[\gamma(\delta)+\frac{I_r(\delta)-\gamma(\delta)}{2}\right]}\left[n(d+2)2^{-nI_r(\delta)}\right] \\ &\leq n(d+2)2^{-n\frac{I_r(\delta)-\gamma(\delta)}{2}}, \end{aligned}$$

since $I_r(\delta) < 2I_h(\delta)$. Thus, for large $n > n_0$, $P(Y \neq X_1) < \epsilon$. □

Proof of Theorem 3.2. Let $d = \delta n$. Let the event $A_j = \{h(X_j, X) < d\}$, $j = 2, \cdots, M_n$. It is clear that $A_2, \cdots, A_{M_n}$ are independent and have same probability, say p_n. By step (2) of the recall algorithm for MNN, and the fact that X is d distance apart from X_1, any event A_j implies that a wrong pattern will be retrieved. Hence,

$$\begin{aligned}
P(Y \neq X_1) &\geq P(\cup_{j=2}^{M_n} A_j) \\
&= 1 - P(\cap_{j=2}^{M_n} \bar{A}_j) \\
&= 1 - \prod_{j=2}^{M_n} (1 - p_n) \\
&= 1 - e^{(M_n - 1)\log(1-p_n)} \\
&= 1 - e^{-(M_n-1)\log(\frac{1}{1-p_n})} \\
&\geq 1 - \epsilon,
\end{aligned}$$

where the last inequality is given by the inequality

$$(M_n - 1)\log(\frac{1}{1-p_n}) > -\log \epsilon,$$

which is proved as follows.

Let $k_1 = \beta(\delta)n$. If k_1 is not a integer, let k_1 be its closest integer. Then

$$\begin{aligned}
p_n &\geq 2^{-2n} \sum_{k=d}^{n} \binom{n}{k} \sum_{i=0}^{d-1} \binom{n-k+d}{i} \\
&\geq 2^{-2n} \binom{n}{k_1} \binom{n-k_1+d}{d-1} \\
&\geq \frac{1}{8n^2} 2^{-nI_r(\delta)}
\end{aligned}$$

Using the given condition

$$\limsup_{n\to\infty} \frac{1}{n}\log_2(M_n) > I_r(\delta),$$

Then for sufficiently large n,

$$-M_n \log(1-p_n) > M_n p_n > -\log \epsilon.$$

□

Proof of Theorem 5.2. Let the event $A_j = \{d_h(X_j, X_0) < d\}$, $j = 2, \cdots, M_n$. It is clear that $A_2, \cdots, A_{M_n}$ are independent and have same probability, say p_n. By step (2) of the recall algorithm for HNNs, and the fact that X is d distance apart from X_1, any event A_j implies that a wrong pattern will be retrieved. Hence,

$$\begin{aligned}
P(Y \neq X_1) &\geq P(\cup_{j=2}^{M_n} A_j) = 1 - P(\cap_{j=2}^{M_n} \bar{A}_j) = 1 - \prod_{j=2}^{M_n}(1-p_n) \\
&= 1 - e^{(M_n-1)\log(1-p_n)} = 1 - e^{-(M_n-1)\log(\frac{1}{1-p_n})} \geq 1 - \epsilon,
\end{aligned}$$

where the last inequality is given by the inequality

$$(M_n - 1)\log(\frac{1}{1-p_n}) > -\log\epsilon,$$

which is proved as follows.

$$\begin{aligned} p_n &= P(A_j) = P(d_h(X_j, X) < d) \geq 2^{-n}\sum_{k=0}^{d-1}\binom{n}{k} \geq 2^{-n}\binom{n}{d-1} \\ &= \frac{n+d-1}{d}2^{-n}\binom{n}{d} \geq \frac{1-\delta}{2\delta n}2^{-nI_h(\delta)} \end{aligned}$$

Using the given condition

$$\limsup_{n\to\infty}\ \frac{1}{n}\log_2(M_n) > I_h(\delta),$$

Then for sufficiently large n,

$$(M_n - 1)\log(\frac{1}{1-p_n}) = -(M_n-1)\log(1-p_n) > (M_n-1)p_n > -\log\epsilon.$$

□