

High-Capacity Exponential Associative Memories

Tzi-Dar Chiueh and Rodney M. Goodman
Department of Electrical Engineering (116 - 81)
California Institute of Technology
Pasadena, California 91125

Abstract

In this paper we present a new generalized associative memory model with potentially high capacity. A memory of this kind with M stored vectors of length N , can be implemented with M nonlinear neurons, N ordinary thresholding neurons, and $2MN$ binary synapses. We show that special cases of this model include the Hopfield and high order correlation memories. We present a new special case of the model based on a neuron which can implement the exponentiation function. Such a neuron can be realized by a MOS transistor operating in the sub-threshold region. We analyze the capacity of this exponential associative memory and show that it scales exponentially with N . In any practical realization however, the dynamic range of the exponentiators will be constrained. We show that the capacity for networks with fixed dynamic range exponentiation circuits is proportional to the dynamic range.

I. Introduction

Since the seminal papers of Hopfield[1][2], there has been much interest in realizing high capacity associative memories. However, the capacity of Hopfield memory in terms of number of fixed points (exemplars), has been found to be severely constrained both empirically [1] and theoretically [3]. Much research has been done in improving the association ability of the Hopfield memory by raising the order of the correlation matrix or by using neurons with nonlinear discriminant functions [4][5][6]. All of them can be implemented by a three-layer feedback network, with the first layer computing the inner products of the input and all exemplars, the second layer raising these quantities to some power, and the third layer accumulating the weighted sum of all exemplars and thresholding the sum.

In section II, we propose a new model which is a generalization of the above associative memories. In particular, we propose to enhance the performance of the memory by using the exponential function instead of polynomial functions. Our motivation is that the exponentiation function is relatively easily done in VLSI, using the MOS transistor's exponential drain current dependence on the gate voltage in the subthreshold region [7][8]. We find in section III that the associative memory we propose exhibits very good performance, and has a capacity that is *exponential* in the length of the stored vectors. Next, we theoretically investigate the issue of constraining the dynamic range available for the exponentiation circuits. We find that in this case the capacity will also be limited, and will be proportional to the dynamic range both in the new associative memory and the high-order associative memory. Section IV presents simulation results which show the behavior of the new memory, and also confirms the theoretical results in section III.

II. A General Model of Associative Memories

In a Hopfield type associative memory [1], the connection matrix is derived from a set of exemplars (memory vectors) to be memorized. When given an initial input, such a network will evolve and become stable at the nearest exemplar (or fixed point) from the input, provided that there are not too many exemplars, and the input is not very far away from that fixed point (in the Hamming distance sense). Essentially, from these exemplars one can construct regions in the binary N -cube $\{1,-1\}^N$, so the network, once initialized with any vector inside a region, will evolve and become stable at the exemplar associated with that region. Thus each exemplar can be regarded as the centroid of a region in $\{1,-1\}^N$. The main evolution rule for a Hopfield memory (with binary neurons) is,

$$d_i' = \text{sgn} \left(\sum_{j=1}^N T_{ij} d_j \right) \quad (1)$$

where $\mathbf{d}' = [d_1' d_2' \dots d_N']$ is the next state of the network and $\mathbf{d} = [d_1 d_2 \dots d_N]$ is the current state. $[T_{ij}]$ is the $N \times N$ connection matrix.

We propose the following neural network associative memory given that $\mathbf{c}^{(k)}$ is to be associated with $\mathbf{e}^{(k)}$, and \mathbf{d}, \mathbf{d}' are the current state and next state of the network. The main evolution equation for this memory is

$$\mathbf{d}' = \text{sgn} \left(\sum_{k=1}^M f(\langle \mathbf{c}^{(k)}, \mathbf{d} \rangle) \mathbf{e}^{(k)} \right) \quad (2)$$

where $\langle \mathbf{a}, \mathbf{b} \rangle$ is the inner product of two vectors \mathbf{a} and \mathbf{b} , and the function $f(\cdot)$ is a nonlinear function. Here if $\mathbf{c}^{(k)}$'s and $\mathbf{e}^{(k)}$'s are the same then this is an autoassociative memory otherwise it is a heteroassociative memory.

Next we will show that some known associative memories are special cases of equation (2).

1. Hopfield Type Associative Memory

Suppose there are M exemplars $\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \dots, \mathbf{c}^{(M)}$ to be stored in the memory, then T_{ij} can be derived by,

$$T_{ij} = \sum_{k=1}^M c_i^{(k)} c_j^{(k)} - M \delta_{ij} \quad (3)$$

Combining equations (1) and (3), we have,

$$\begin{aligned} d_i' &= \text{sgn} \left(\sum_{j=1}^N \sum_{k=1}^M c_i^{(k)} c_j^{(k)} d_j - M d_i \right) \\ &= \text{sgn} \left(\sum_{k=1}^M \left[\sum_{j=1}^N c_j^{(k)} d_j \right] c_i^{(k)} - M d_i \right) \end{aligned}$$

$$d_i' = \text{sgn} \left(\sum_{k=1}^M \langle \mathbf{c}^{(k)}, \mathbf{d} \rangle c_i^{(k)} - M d_i \right) \quad (4)$$

Since equation (4) is true for $i = 1, 2, \dots, N$, we can rewrite (4) in the vector form,

$$\mathbf{d}' = \text{sgn} \left(\sum_{k=1}^M \langle \mathbf{c}^{(k)}, \mathbf{d} \rangle \mathbf{c}^{(k)} - M \mathbf{d} \right) \quad (5)$$

The argument of the sgn function in (4) can be interpreted as a weighted sum of all M exemplars minus a constant, and the weights are the inner products of the input \mathbf{d} and the corresponding exemplars $\mathbf{c}^{(k)}$. Now suppose the input \mathbf{d} is nearest a particular exemplar, say $\mathbf{c}^{(l)}$, that is,

$$\langle \mathbf{d}, \mathbf{c}^{(l)} \rangle > \langle \mathbf{d}, \mathbf{c}^{(k)} \rangle \quad \forall k \neq l$$

then ideally we would like to have the weight associated with $\mathbf{c}^{(l)}$ be the largest of all weights and preferably be the dominant one so that the next state \mathbf{d}' will be very similar to $\mathbf{c}^{(l)}$ if not identical.

Comparing equation (5) with the generalized evolution equation (2), we see that the Hopfield type memory (with non-zero diagonal) is a special case of the general model which has a linear function for $f(\cdot)$. By choosing other $f(\cdot)$ we can construct several known memories.

2. High Order Inner Product Associative Memory

If $f(\cdot)$ is chosen to be a power function, i.e.

$$f(x) = x^p$$

then equation (2) becomes the evolution equation of a high-order associative memory [5][6].

3. One Shot Associative Memory using N-flops

If we let f be an N-flop or winner-take-all function, that is if

$$f(x_k) = \begin{cases} 1 & x_k > x_l \quad l = 1, 2, \dots, M, l \neq k \\ 0 & \text{otherwise} \end{cases}$$

where $x_l = \langle \mathbf{c}^{(l)}, \mathbf{d} \rangle$, then obviously no iteration is necessary and the correct memory vector is retrieved at once.

4. Exponential Inner Product Associative Memory

We now propose a new *exponential associative memory* by choosing $f(\cdot)$ as

$$f(x) = b^x$$

where b is a fixed number and $b > 1$. Intuitively, the performance of such a associative memory should be better than a high-order correlation memory since the weight associated with the nearest exemplar will be more dominant than in the high-order correlation memory case.

Next we shall consider the complexity of such a memory (see Fig. 1). First, we need M neurons to calculate the inner products of the input and M exemplars, and to perform the nonlinear function $f(\cdot)$. Each of these M neurons has N synapses (matrix W) carrying the signals from input. After obtaining the weights we have to compute the weighted sum and threshold - this can be done by N ordinary neurons, each with M synapses (matrix W^T). We therefore need a total of M nonlinear neurons, N ordinary neurons and $2MN$ binary synapses (with value 1 or -1).

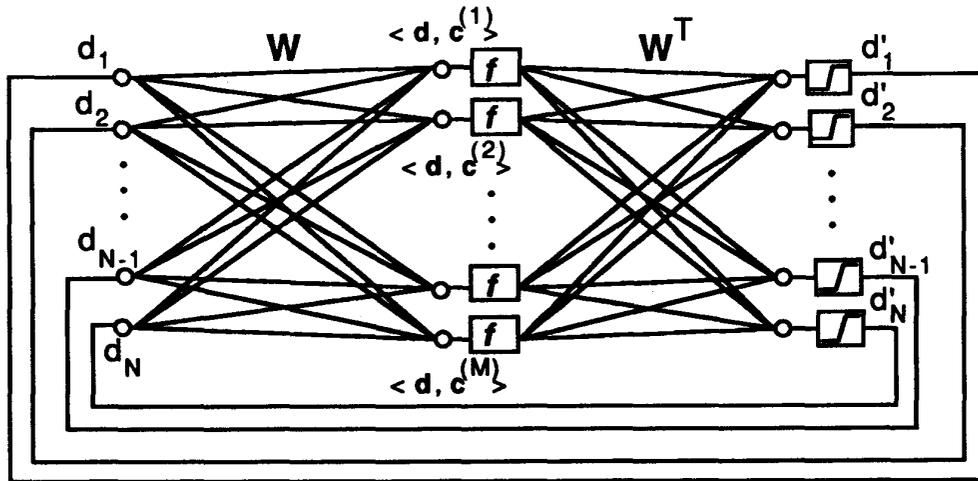


Figure 1 Associative memory constructed with inner product method

III. Capacity of the exponential memory

From now on, we will consider only the case that $f(x) = b^x$, where b is a fixed constant and $b > 1$. Thus equation (2) becomes (in the autoassociative memory case)

$$d' = \operatorname{sgn} \left(\sum_{k=1}^M b \langle c^{(k)}, d \rangle c^{(k)} \right) \quad (6)$$

Furthermore, suppose the input d is one of the exemplars say $c^{(h)}$, then

$$\begin{aligned} d' &= \operatorname{sgn} \left(\sum_{k=1}^M b \langle c^{(k)}, c^{(h)} \rangle c^{(k)} \right) \\ &= \operatorname{sgn} \left(b^N c^{(h)} + \sum_{\substack{k=1 \\ k \neq h}}^M \beta_k c^{(k)} \right) \end{aligned}$$

where β_k is b raised to the power of the inner product of $c^{(k)}$ and $c^{(h)}$.

Now consider the i^{th} component of \mathbf{d}' , we have

$$d'_i = \text{sgn} \left(b^N c_i^{(h)} + \sum_{\substack{k=1 \\ k \neq h}}^M \beta_k c_i^{(k)} \right) \quad (7)$$

In order to compute the capacity of this model, we make the following assumption, each exemplar is randomly generated, i.e. every component of a memory vector is the outcome of a Bernoulli trial (-1 or 1). We will use a signal to noise ratio (SNR) approach [3][9] to calculate the capacity. The signal in equation (7) corresponds to the first term, hence the signal power is

$$S = b^{2N}$$

The second term is actually a sum of $M-1$ i.i.d. (independent identically distributed) random variables (x_1, x_2, \dots, x_M except x_h). Therefore the variance of the second term is $M-1$ times the variance of a single random variable. Suppose

$$x_1 = \beta_1 c_i^{(1)}, \quad x_2 = \beta_2 c_i^{(2)}, \quad \dots, \quad x_M = \beta_M c_i^{(M)}$$

After some derivation the following results are in order (assuming, without loss of generality, $c_i^{(h)} = +1$),

$$\begin{aligned} \Pr(x_1 = b^{N-2k}) &= 2^{-N} \binom{N-1}{k} & k = 0, 1, \dots, N-1 \\ \Pr(x_1 = -b^{N-2k-2}) &= 2^{-N} \binom{N-1}{k} & k = 0, 1, \dots, N-1 \end{aligned}$$

and the expectation of x_1 is

$$E[x_1] = b^N 2^{-N} (1 - b^{-2})(1 + b^{-2})^{N-1}$$

The expectation of $(x_1)^2$ and the variance of x_1 are

$$\begin{aligned} E[x_1^2] &= [(1 + b^{-4})(b^2/2)]^N \\ \text{Var}[x_1] &= E[x_1^2] - (E[x_1])^2 \\ &= (b/2)^{2N} [(2 + 2b^{-4})^N - (1 + b^{-2})^{2(N-1)}(1 - b^{-2})^2] \end{aligned}$$

Therefore, the signal to noise ratio is

$$\text{SNR} = \frac{S}{(M-1)\text{Var}[x_1]} = \frac{2^{2N}}{(M-1)[(2 + 2b^{-4})^N - (1 + b^{-2})^{2(N-1)}(1 - b^{-2})^2]} \quad (8)$$

From eq. (8), one can obtain an estimate of the capacity of this model

$$M \sim \frac{4^N}{\text{SNR} (2 + 2b^{-4})^N} = \frac{[2/(1 + b^{-4})]^N}{\text{SNR}} \quad (9)$$

Equation (9) shows that at a specific SNR, the capacity scales *exponentially* with N, the length of the stored vectors.

This exponential capacity is very attractive, however, the dynamic range required of the exponentiators grows exponentially with N. In any real implementation, the dynamic range of the exponentiation circuit will be constrained. (In our proposed VLSI implementations to approximately 10^5 to 10^7 [8]). We therefore need to analyze the performance of this model in the case of fixed dynamic range. Suppose the dynamic range of the exponentiators is fixed at D, where

$$D = b^{2N}$$

Then as N grows, b decreases, so M does not scale exponentially with N (see eq. (9)). We will deal with the case where N is very large with D fixed. Let

$$b = 1 + \mu$$

where $\mu \ll 1$. Since $N \gg 1$, then

$$\ln D = 2N \ln b = 2N \ln (1 + \mu) \approx 2N\mu$$

and

$$\begin{aligned} (1 + b^{-4})^N &\approx (1 + 1 - 4\mu)^N \approx 2^N (1 - (\ln D)/N)^N \\ &\approx 2^N e^{-(\ln D)} = 2^N / D \end{aligned}$$

Substituting the above equation in equation (9), we have

$$M \sim D/\text{SNR} \quad (10)$$

We thus conclude that with the dynamic range fixed, the capacity of the new exponential associative memory is proportional to the dynamic range. This agrees with known results for the high order associative memory [5] which show the capacity proportional to a power of N, which is also equal to the dynamic range.

IV. Simulation Results

We have simulated the new exponential memory (we choose $b = 2$) in order to confirm the theoretical results presented above. In each case, we randomly generate 10 sets of M N-bit exemplars. For each of these, 10 input vectors are generated by randomly picking a particular exemplar and flipping T bits. Then the network is allowed to run until it becomes stable at a fixed point. That point is compared to the original exemplar and if they match this run is called a success. We then collect the number of successes out of 100 trials and if this is greater than 97, then we say that with M exemplars this network can tolerate T errors. The largest T for each M we call the *attraction radius* (denoted by R). In figure 2, we illustrate how the attraction radius scales with N. One can see that the four curves are almost parallel to each other, thus the

capacity, which is the value of M where the curve meets the horizontal axis, scales approximately exponentially with N - which confirms equation (9). In figure 3, we compare the third order correlation memory (which has $p = 2$) with the new memory. One can easily see that the new memory has a much larger capacity than the third order memory. Figure 4 shows the attraction radius (R) of memories with $N=32$ and $N=64$, and with four different dynamic ranges 2^8 , 2^{10} , 2^{12} , 2^{14} . Again the curves are parallel to each other, thus we conclude that the capacity of the exponential memory with limited dynamic range scales with the dynamic range - again confirming our theoretical prediction.

V. Conclusions

In this paper we have presented a new high capacity associative memory based on using the exponential function in the evolution equation. In a practical implementation the capacity of the memory is proportional to the dynamic range of the exponentiation circuits. Future work will report on the results of implementing these memories in VLSI.

References

- [1] J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities", *Proc. Nat. Acad. Sci. USA*, Vol. 79 (1982), pp. 2554-2558.
- [2] J. J. Hopfield, "Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons", *Proc. Nat. Acad. Sci. USA*, Vol. 81 (1984), pp. 3088-3092.
- [3] R. J. McEliece, et. al. "The Capacity of The Hopfield Associative Memory", *IEEE Tran. on Information Theory*, Vol. IT-33 (1987) pp. 461-482.
- [4] Y.C. Lee et. al., "Machine Learning using a Higher Order Correlation Network", *Physica*, 22D, (1986), pp. 276-306.
- [5] B. Soffer, "Holographic Associative Memories", *Proc. of Workshop on Neural Network Devices and Applications*, Jet Propulsion Lab. Feb. 1987, pp. 125-146.
- [6] D. Psaltis and Cheol Hoon Park, "Nonlinear Discriminant Functions and Associative Memory", *AIP Conf. Proc.*, no. 151, *Neural Network Computing*, 1986, pp. 370-375.
- [7] C. A. Mead, Course Notes of "Analog Integrated Circuit Design", Caltech, 1987.
- [8] L. A. Glasser and D. W. Dobberpuhl, *The Design and Analysis of VLSI Circuits*, Addison-Wesley, 1985.
- [9] T. D. Chiueh and R. M. Goodman, "A Neural Network Classifier Based on Coding Theory", *Proc. of IEEE Conference on Neural Information Processing Systems*, Denver, Colorado, Nov., 1987.

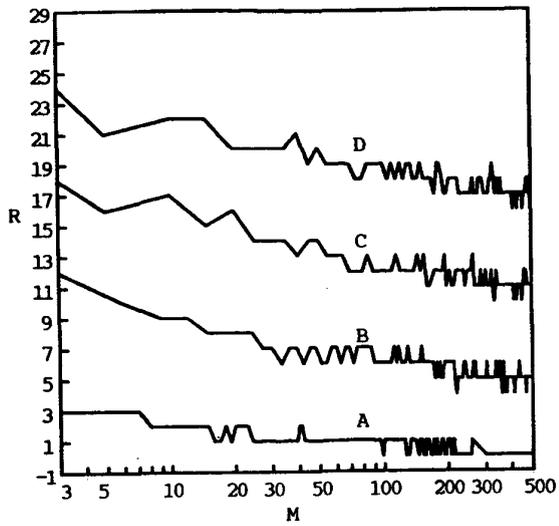


Figure 2 Attraction radius (R) vs. the number of exemplars stored (M) in the new associative memory
 curve A : N = 16
 curve B : N = 32
 curve C : N = 48
 curve D : N = 64

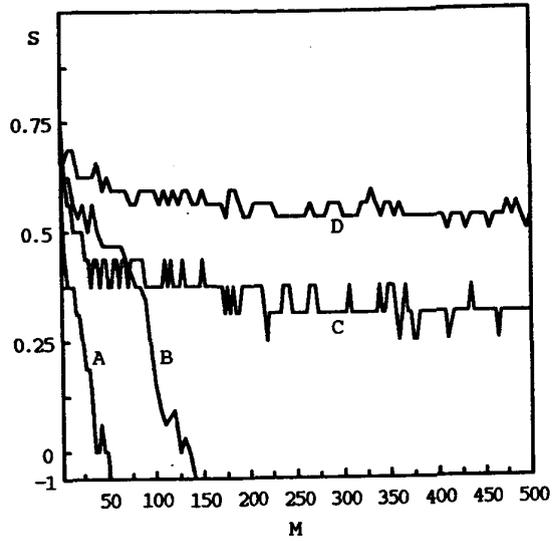
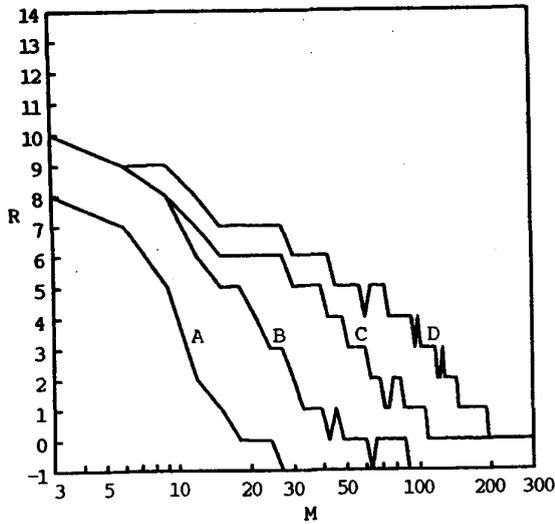
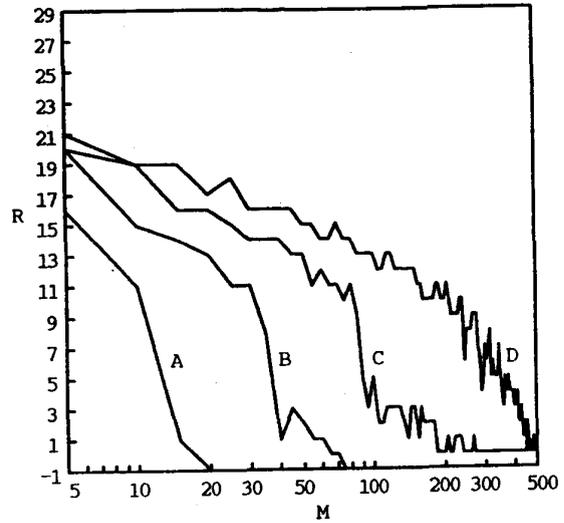


Figure 3 Normalized attraction radius (S) vs. the number of exemplars stored (M)
 curve A : N = 32, third order correlation memory
 curve B : N = 64, third order correlation memory
 curve C : N = 32, new associative memory
 curve D : N = 64, new associative memory



curve A : N = 32, D = 28
 curve B : N = 32, D = 210
 curve C : N = 32, D = 212
 curve D : N = 32, D = 214



curve A : N = 64, D = 28
 curve B : N = 64, D = 210
 curve C : N = 64, D = 212
 curve D : N = 64, D = 214

Figure 4 Attraction radius(R) vs. the number of exemplars stored(M) in memories with different dynamic ranges