Neural Networks, Vol. 1, pp. 149–163, 1988
Printed in the USA.

0893-6080/88 $3.00 + .00

ORIGINAL CONTRIBUTION

Higher Order Associative Memories and Their Optical Implementations*

DEMETRI PSALTIS, CHEOL HOON PARK, AND JOHN HONG†

California Institute of Technology

(*Received and accepted* 22 *December* 1987)

Abstract—*The properties of higher order memories are described. The non-redundant, up to Nth order polynomial expansion of N-dimensional binary vectors is shown to yield orthogonal feature vectors. The properties of expansions that contain only a single order are investigated in detail and the use of the sum of outer product algorithm for training higher order memories is analyzed. Optical implementations of quadratic associative memories are described using volume holograms for the general case and planar holograms for shift invariant memories.*

1. INTRODUCTION

An associative memory can be thought of as a system that stores a prescribed set of vector pairs $(\mathbf{x}^m, \mathbf{y}^m)$ for $m = 1, \ldots, M$ and also produces $\mathbf{y}^m$ as its output when $\mathbf{x}^m$ becomes its input. We denote by N and N_0 the dimensionalities of the input and output vectors, respectively. When the output vectors are stored as binary N_0-tuples, the associative memory can be implemented as an array of discriminant functions, each dichotomizing the input vectors into two classes. This type of associative memory is shown schematically in Figure 1. In evaluating the effectiveness of a particular associative memory we are concerned with its ability to store a large number of associations (capacity), the ease with which the parameters of the memory can be set to realize the prescribed mappings (learning), and how it responds to inputs that are not members of its training set (generalization). In this paper we discuss a class of associative memories known as higher order memories that have been recently investigated by a number of separate research groups (Baldi & Venkatesh, 1987; Chen *et al.*, 1986; Giles & Maxwell, 1987; Maxwell, Giles, Lee, & Chen, 1986; Newman, 1987; Poggio, 1975; Psaltis & Park, 1986; Sejnowski, 1986). Our motivation for investigating these memories was the increase in storage capacity that results from the increase in the number of independent parameters or degrees of freedom that is needed to describe a higher order associative mapping. The relationship between the degrees of freedom of a memory and its ability to store associations (Abu-Mostafa & Psaltis, 1985) is fundamental to this work and we state it in the following subsection as a theorem.

1.1 Degrees of Freedom and Storage Capacity

Let D be the number of independent variables (degrees of freedom) we have under our control to specify input–output mappings and let each parameter have K separate levels or values that it can assume. We define the storage capacity C to be the maximum number of arbitrary associations that can be stored and recalled without error.

Theorem 1.

$$C \leq \frac{D \log_2 K}{N_0}. \tag{1}$$

Proof: The number of different states of memory is given by K^D and the total number of outputs that a given set of M input patterns can be mapped to is $2^{N_0 M}$. If the number of mappings were larger than the number of distinct states of the memory, then mappings would exist that are not implementable. Requiring that all mappings can be done leads to the relationship of the theorem.

The equality in (1) is achieved by Boolean circuits such as programmable logic arrays and an extreme case of a higher order memory we will discuss later. When the equality holds, resetting any one bit in any one of

* Funded by the Air Force Office of Scientific Research, the Army Research Office and the Defense Advanced Research Projects Agency.

† Dr. Hong is now with the Rockwell Science Center, Thousand Oaks, CA 91360.

Requests for reprints should be sent to Demetri Psaltis, Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125.

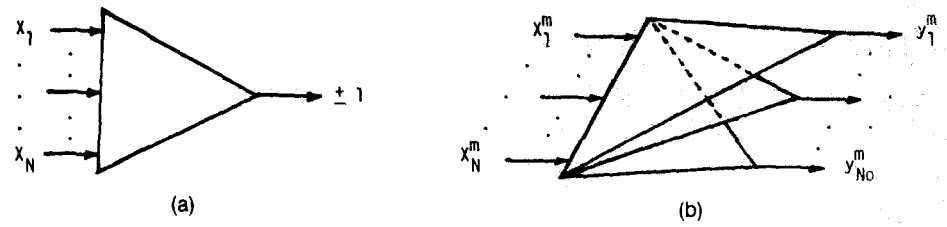


FIGURE 1. (a) Discriminant function; (b) Associative memory constructed as an array of discriminant functions.

the parameters of the memory gives a different mapping. Such a memory cannot learn from the training set to respond in some desirable way to inputs that it has never seen before. The only way to get generalization when $C = D \log_2 K/N_0$ is to impose on it the overall structure of the memory before learning begins. One of the appealing features of neural architectures is the considerable redundancy in the degrees of freedom that is typically available. Therefore, there is hope that while a memory learns specific input–output correspondences it can also discover the underlying structure that may exist in the problem and learn to respond correctly for a set of inputs much larger than the training set. Moreover, the same redundancy is responsible for the error tolerance that is evident in many neural architectures. Higher order memories are generally redundant and they can provide us with a methodology for selecting the degree of redundancy along with the number of degrees of freedom and the associated capacity to store random problems.

It is important to keep in mind that (1) holds for *arbitrary* mappings. If the input and output vectors are restricted in some way that happens to be matched to the architecture of a particular associative memory then it may be possible to overcome this limit. However, selecting the architecture of the associative memory such that it optimally implements only a subset of all possible associations is basically equivalent to choosing the architecture so that it generalizes in a desirable way. For instance suppose that we design an associative memory so that it is shift invariant (i.e., the output is insensitive to a change in the position of the input) (Maxwell *et al.*, 1986; Psaltis & Hong, 1987). Then this system will respond predictably to all the shifted versions of the patterns that were used to train it. We can equivalently think of this system as having a larger storage capacity than the limit of (1) over the set of shift invariant mappings. If we can identify a priori the types of generalization we wish the memory to exhibit, and we can find ways to impose these on the architecture, then this is certainly a sensible thing to do. Higher order memories can also provide a convenient framework within which this can be accomplished.

The penalty we must pay for the increase in the storage capacity that is afforded by the increase in the degrees of freedom in a higher order associative memory is increase in implementation complexity. The computer that implements a higher order memory must have sufficient storage capacity to store a very large number of parameters. Moreover it must be capable of addressing the stored information with a high degree of parallelism in order to produce an output quickly. We will discuss in this paper optical implementations of second order memories and we will show a remarkable compatibility between the computational requirements of these memories and the ability of optics to store information in three dimensions.

1.2 Linear Discriminant Functions and Associative Memories

We will consider as a precursor the most familiar associative memories that are constructed as arrays of linear discriminant functions (Kohonen, 1984). A linear discriminant function is a mapping from the sample space X, a subset of R^N, to 1 or -1.

$$y = \operatorname{sgn}\{\mathbf{w}^t \cdot \mathbf{x} + w_0\}$$

$$= \operatorname{sgn}\{w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_N x_N\} \quad (2)$$

where sgn is the signum function, $\mathbf{w}$ is a weighting vector and w_0 is a threshold value. In this case the capacity is upperbounded by $(N + 1)\log_2 K$ according to our definition of capacity. In this relatively simple case the exact capacity is known to be equal to $C = N + 1$ assuming the input points are in general position and $K = \infty$ (Cover, 1965). An associative memory is constructed by simply forming an array of linear discriminant functions each mapping the same input to a different binary variable. Several algorithms exist for training such memories including the perceptron, Widrow-Hoff, sum of outer products, pseudoinverse, and simplex methods (Duda & Hart, 1973; Hopfield, 1982; Kohonen, 1984; Venkatesh & Psaltis, in press). This memory can be thought of as the first order of the broader class of higher order memories that contain not only a linear expansion of the input vector but also quadratic and higher order terms. We will see in Section 3 that the learning methods that are applicable to the linear memories generalize directly to the higher order memories. First, however, we will describe the properties of the mappings that are implementable with higher order memories in Section 2. Finally, in Section 4 we will describe optical implementations of quadratic optical memories (Psaltis, Park, & Hong, 1986).

2. PROPERTIES OF HIGHER ORDER MEMORIES

A Φ-function is defined to be a *fixed* mapping of the input vector **x** to an L-dimensional vector **z** followed by a linear discriminant function.

$$y = \text{sgn}\{\mathbf{w}^t \cdot \mathbf{z}(\mathbf{x}) + w_0\}$$
$$= \text{sgn}\{w'_1 z_1 + w'_2 z_2 + \cdots + w'_L z_L + w_0\} \quad (3)$$

where $\mathbf{z}(\mathbf{x}) = (z_1(\mathbf{x}), z_2(\mathbf{x}), \ldots, z_L(\mathbf{x}))$, $\mathbf{w}'$ is an L dimensional weighting vector and $\mathbf{z}(\mathbf{x})$ is an L dimensional vector derived from **x**. The storage capacity in this case is equal to the capacity of the second layer $L + 1$ (Cover, 1965) if the samples **z** are in general position whereas the upper bound on the capacity from (1) is $(L + 1)\log_2 K$. The inefficiency in this case is $\log_2 K$ bits, the same as for the linear discriminant function even though the capacity can be raised arbitrarily by increasing L. It is not known what the exact relationship between L and K is, that is, we do not know whether for higher dimensions we need better resolution for the values of the weights to be capable of implementing a fixed fraction of the linear mappings. Recently, Mok and Psaltis (personal communication) have found the asymptotic (large N) statistical capacity to be $C = N$ for a linear discriminant function with binary weights. This result implies that even for large N, for the vast majority of linear dichotomies, a large number of levels is not required. Therefore a Φ-function is an effective and straightforward method for increasing the capacity of an associative memory without loss in efficiency.

A higher order associative memory is an array of Φ-functions with the mappings **z**(**x**) being polynomial expansions of the vector **x**. The schematic diagram of a higher order associative memory is shown in Figure 2. When the polynomial expansion is of the rth order in **x** then the output vector **y** is given by

$$y_l = \text{sgn}\{W_l^r(\mathbf{x}, \mathbf{x}, \ldots, \mathbf{x}) + W_l^{r-1}(\mathbf{x}, \ldots, \mathbf{x}) + \cdots + W_l^2(\mathbf{x}, \mathbf{x}) + W_l^1\mathbf{x} + w_{l0}\} \quad (4)$$

where $l = 1, \ldots, N_0$, W_l^k is a k-linear symmetric mapping and W_l^1 is equivalent to $\mathbf{w}^l$ in (2). According to (3)

$$z_j(\mathbf{x}) = x_{p_1(j)}^{n_1} x_{p_2(j)}^{n_2} \cdots x_{p_r(j)}^{n_r} \quad (5)$$

where $j = 1, 2, \ldots, L$, $p_i(j) \in \{1, 2, \ldots, N\}$, such that all the $_j$ are distinct, and $n_1, n_2, \ldots, n_r = 0, 1$. Then L is $\binom{N+r}{r}$ (Cover, 1965), and hence the capacity bound is $(\binom{N+r}{r} + 1)\log_2 K$ as before. For example, if $r = 2$, the function becomes quadratic and has the form $y_l = \mathbf{x}^t W_l^2 \mathbf{x} + W_l^1 \mathbf{x} + w_{l0}$ and the number of non-redundant terms in the quadratic expansion is $L = (N + 1)(N + 2)/2$.

The components of the vector **z** are binary-valued if **x** is binary. In this case, the samples cannot be assumed to be in general position since there are at most $N + 2$ binary vectors in N dimensional space which lie in general position. We will evaluate the effectiveness of higher order mappings in producing representations **z**(**x**) that are separable by the second layer of weights by calculating the Hamming distance between **z** vectors given the Hamming distance between the corresponding **x** vectors. We expect that if the Hamming distance between two binary vectors is large then they are easy to distinguish from one another.

2.1 Complete Polynomial Expansion of Binary Vectors

There are at most 2^N non-redundant terms in any polynomial expansion (4) of a binary vector **x** in N dimensions. First, we will consider the following Nth order expansion (or equivalently bit production) for the bipolar vectors **x** in N dimensional binary space $\{1, -1\}^N$:

$$\mathbf{z} = \mathbf{z}(\mathbf{x})$$
$$= (1, x_1, x_2, \ldots, x_N, x_1x_2, \ldots, x_1x_2 \cdots x_N)^t. \quad (6)$$

If we apply a linear discriminant function to the new vectors **z**, then the capacity becomes 2^N which is equal to the total number of possible input vectors (Psaltis & Park, 1986). In other words this memory is capable of performing *any* mapping of N binary variables to any binary output vector **y**. Of course the number of weights that are needed to implement this memory grows to 2^N times N_0, the number of bits at the output. In what follows we show that in this extreme case the vectors **z** become orthogonal to each other.

Theorem 2. If we expand binary vectors $\mathbf{x}^m$ ($m = 1, 2, \ldots, 2^N$) in $X_B = \{1, -1\}^N$ to 2^N dimensional binary

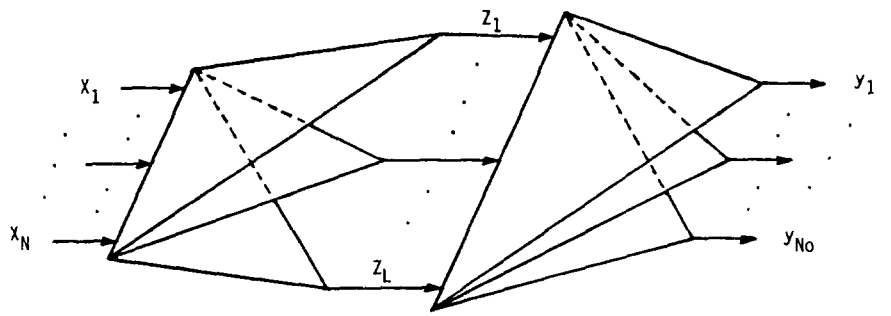


FIGURE 2. Higher order associative memory.

vectors $\mathbf{z}^m$ according to (6), where N is the dimensionality of the original feature vectors, then (a) $\langle \mathbf{z}^{m_1}, \mathbf{z}^{m_2} \rangle = 2^N \delta_{m_1 m_2}$ where $\langle \cdot, \cdot \rangle$ is an inner product, (b) $\sum_i z_i^m = 0$, (c) $\sum_m z_{j_1}^m z_{j_2}^m = 2^N \delta_{j_1 j_2}$ and $\sum_m z_j^m = 0$.

Example: Table 1 is for the case of $N = 3$. Note the orthogonality and the numbers of 1s and -1s in the new vectors and the set of each component of them except the first vector and the set of the first components.

Proof: (a) Let us consider any two different binary vectors in the binary space of $\{1, -1\}^N$ whose Hamming distance is n ($1 \le n \le N$). When they are expanded to two 2^N dimensional binary vectors, the number of kth order terms that have opposite signs in the two expansions is

$$\binom{n}{1}\binom{N-n}{k-1} + \binom{n}{3}\binom{N-n}{k-3} + \binom{n}{5}\binom{N-n}{k-5} + \cdots. \quad (7)$$

Notice that two polynomials have different values if, and only if, they have an odd number of terms whose signs are opposite. The Hamming distance between the two fully expanded (up to order 2^N) vectors can be calculated by adding the number of terms that have different signs over all the orders of the expansion:

$$\begin{aligned}
&\binom{n}{1}\binom{N-n}{0} + \binom{n}{1}\binom{N-n}{1} \\
&\quad + \left\{\binom{n}{1}\binom{N-n}{2} + \binom{n}{3}\binom{N-n}{0}\right\} \\
&\quad + \left\{\binom{n}{1}\binom{N-n}{3} + \binom{n}{3}\binom{N-n}{1}\right\} + \cdots \\
&\quad + \left\{\binom{n}{1}\binom{N-n}{N-n} + \binom{n}{3}\binom{N-n}{N-n-2}\right. \\
&\quad \left. + \binom{n}{5}\binom{N-n}{N-n-4} + \cdots\right\} \\
&\quad + \left\{\binom{n}{3}\binom{N-n}{N-n-1} + \binom{n}{5}\binom{N-n}{N-n-3}\right. \\
&\quad \left. + \binom{n}{7}\binom{N-n}{N-n-5} + \cdots\right\} \\
&\quad + \cdots \\
&= \sum_{i=\text{odd}} \binom{n}{i} \sum_{j=0}^{N-n} \binom{N-n}{j} = 2^{n-1} 2^{N-n} \\
&= 2^{N-1}. \qquad (8)
\end{aligned}$$

The fact that the Hamming distance is 2^{N-1} for any two expanded vectors (for any n) proves that all of the 2^N vectors become orthogonal and that $\langle \mathbf{z}^{m_1}, \mathbf{z}^{m_2} \rangle = 2^N \delta_{m_1 m_2}$. (b) Just think of the cases where one of the two vectors is $(1, 1, \ldots, 1)$. Then, all the other vectors $\mathbf{z}$ have equal number of 1s and -1s because their Hamming distances are all 2^{N-1} from the $(1, 1, \ldots, 1)$ vector. (c) See Duda and Hart (1973, p. 109).

Slepian has discussed this orthogonalization property as a method for designing orthogonal codes and has given a different proof for it (Slepian, 1956). The proof presented here is useful for characterizing higher order memories because it allows us to trace the contribution of each order of the expansion to the orthogonalization and immediately derive results about the properties of quadratic and cubic memories. The output vector $\mathbf{y}$ is

$$y_l = \operatorname{sgn}\{W_l \cdot \mathbf{z}\} = \operatorname{sgn}\left\{\sum_{i=1}^{2^N} W_{li} z_i\right\} \quad (9)$$

where $l = 1, \ldots, N_0$ and W_l is a 2^N dimensional weighting row vector. The matrix W_{li} that can implement the $\mathbf{x}^m \mapsto \mathbf{y}^m$ mapping for $m = 1$ to 2^N can be formed in this case simply as the sum of outer products of $\mathbf{y}^m$ and $\mathbf{z}^m$:

$$W_{li} = \sum_{m=1}^{2^N} y_l^m z_i^m. \quad (10)$$

2.2 Expansions of a Single Order

The orthogonalization property of the full expansion is interesting because it shows that higher order memories provide a complete framework that takes us from the simplest "neuron," the linear discriminant function, to the full capability of a Boolean look-up table. Higher order memories can indeed provide a valuable tool for designing digital programmable logic arrays. In this paper, however, we are interested in associative memories that are capable of accepting inputs with

TABLE 1

x_1	x_2	x_3	1	x_1	x_2	x_3	x_1x_2	x_2x_3	x_3x_1	$x_1x_2x_3$
1	1	1	1	1	1	1	1	1	1	1
1	1	−1	1	1	1	−1	1	−1	−1	−1
1	−1	1	1	1	−1	1	−1	−1	1	−1
1	−1	−1	1	1	−1	−1	−1	1	−1	1
−1	1	1	1	−1	1	1	−1	1	−1	−1
−1	1	−1	1	−1	1	−1	−1	−1	1	1
−1	−1	1	1	−1	−1	1	1	−1	−1	1
−1	−1	−1	1	−1	−1	−1	1	1	1	−1

large N (e.g., if $N = 10^3$ then $2^N \approx 10^{300}$) in which case considering a full expansion of the input data is completely out of the question. In such cases we are really interested in an expansion that contains a large enough number of terms to provide the capacity needed to learn the problem at hand. In this subsection we analyze the properties of partial expansions that include all the terms of one order.

We will first consider the memory consisting of all the terms of a quadratic expansion with binary input vectors.

$$y_l = \text{sgn}\{\sum_i \sum_j w_{lij} x_i x_j\} = \text{sgn}\{\sum_{k=1}^{L} w'_{lk} z_k\}. \quad (11)$$

The number of non-redundant terms in a quadratic expansion of a binary vector is $L = N(N-1)/2$. Let two input vectors have a Hamming distance n. The angle between these two vectors is given by the relation $\cos\theta_1 = 1 - (2n/N)$. The angle θ_2 between the corresponding $\mathbf{z}(\mathbf{x})$ vectors can be readily calculated since we know their Hamming distance from the proof of Theorem 2(a):

$$\cos\theta_2 = 1 - \frac{4n(N-n)}{N(N-1)} \approx 1 - 4\rho + 4\rho^2 = (1-2\rho)^2 \quad (12)$$

where $\rho = n/N$. θ_2 and θ_1 are plotted versus ρ in Figure 3a. For $\rho < .5$, θ_2 is always larger than θ_1. Specifically for $\rho \ll 1$, $\theta_2 = \sqrt{2} \times \theta_1$. We see therefore that the quadratic mapping not only expands the dimensionality which provides capacity but also spreads the input samples apart, a generally desirable property. For $\rho > .5$ the quadratically expanded vectors are closer to each other than the original vectors and in the extreme case $n = N$, θ_2 becomes zero. This insensitivity of the quadratic mapping to a change in sign of all the bits is a property that is shared by all even order expansions. Next we consider a cubic memory

$$y_l = \text{sgn}\{\sum_i \sum_j \sum_k w_{lijk} x_i x_j x_k\} = \text{sgn}\{\sum_{n=1}^{L} w'_{ln} z_n\} \quad (13)$$

where $L = \binom{N}{3} + N$. In Figure 3b we plot θ_3, the angle between two cubically expanded binary vectors as a function of ρ. For convenience, θ_1 is also plotted in the same figure. In this case θ_3 increases faster with ρ for $\rho < .5$. For $\rho \ll 1$, $\theta_3 = \sqrt{3} \times \theta_1$. At $\rho \approx .4$ the cubic expansion gives essentially perfectly orthogonal vectors while for $\rho > .5$, θ_3 remains smaller than θ_1 and in the limit $\rho = 1$, $\theta_3 = \pi$. Thus the cubic memory discriminates between a vector and its complement.

The basic trends that are evident in the quadratic and cubic memories generalize to any order r. The number of independent terms in the rth order expansion of a binary vector is $\binom{N}{r}$ which is maximum for $r \approx N/2$. Again this is not of practical importance because the number of terms in a full expansion of this sort is prohibitively large. What is of interest however is the effectiveness with which relatively small order expansions can orthogonalize a set of input vectors. The angle θ_r between two vectors that have been expanded to the rth order is given by the following relation:

$$\cos\theta_r = \frac{\binom{N}{r} - 2\sum_{i=\text{odd}} \binom{n}{i}\binom{N-n}{r-i}}{\binom{N}{r}}. \quad (14)$$

We can obtain a simpler expression for the interesting case $r \ll N$ and for small ρ, $\theta_r \approx \sqrt{r} \times \theta_1$.

Proposition 3: *For* $r \ll N$,

$$\cos\theta_r \approx (1-2\rho)^r. \quad (15)$$

Moreover, for small ρ,

$$\theta_r \approx \sqrt{r}\theta_1 \quad (16)$$

where $\theta_1 \approx 2\sqrt{\rho}$.

Proof: For a small r, we can make the approximations $\binom{N}{r} \approx N^r/r!$, $\binom{n}{i} \approx n^i/i!$, and $\binom{N-n}{r-i} \approx (N-n)^{r-i}/(r-i)!$. Then, $\cos\theta_r$ is approximated as follows:

$$\cos\theta_r \approx 1 - 2\sum_{i=\text{odd}} \frac{r!}{i!(r-i)!}\rho^i(1-\rho)^{r-i} = (1-2\rho)^r$$

because of these relationships:

$$\sum_{i=\text{odd}} + \sum_{i=\text{even}} = (1-\rho+\rho)^r = 1,$$

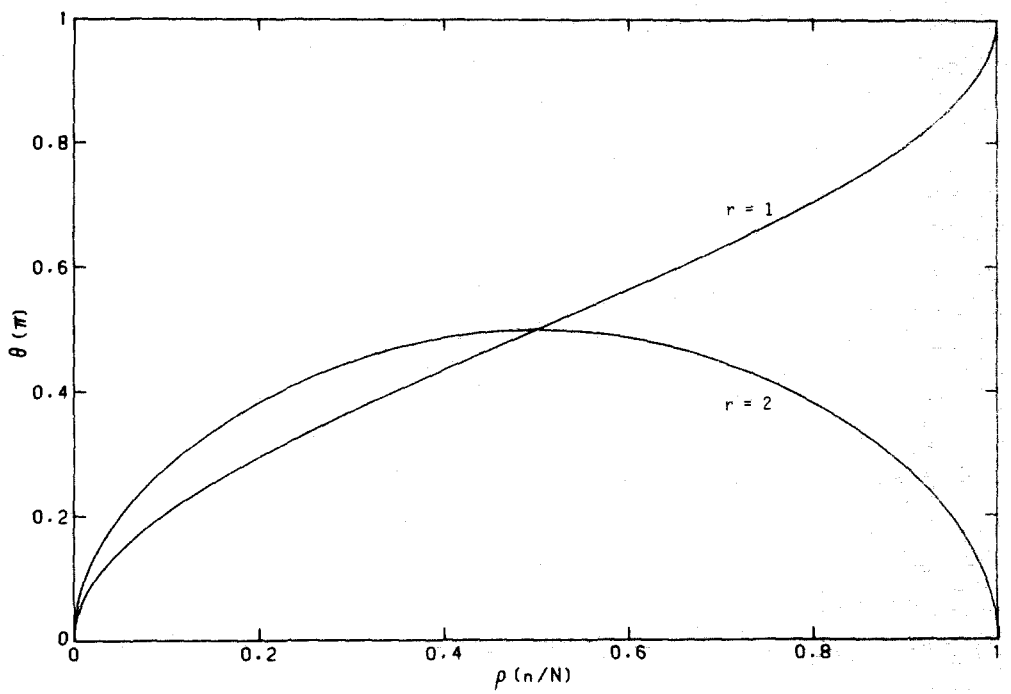
$$\sum_{i=\text{odd}} - \sum_{i=\text{even}} = -(1-\rho-\rho)^r = -(1-2\rho)^r.$$

When $\rho \ll 1$, $\cos\theta_r$, which is approximately $1 - \theta_r^2/2!$, is approximated by $1 - 2r\rho$ directly from (14) or from (15). Therefore, it is followed by (16) that $\theta_r \approx 2\sqrt{r\rho}$.

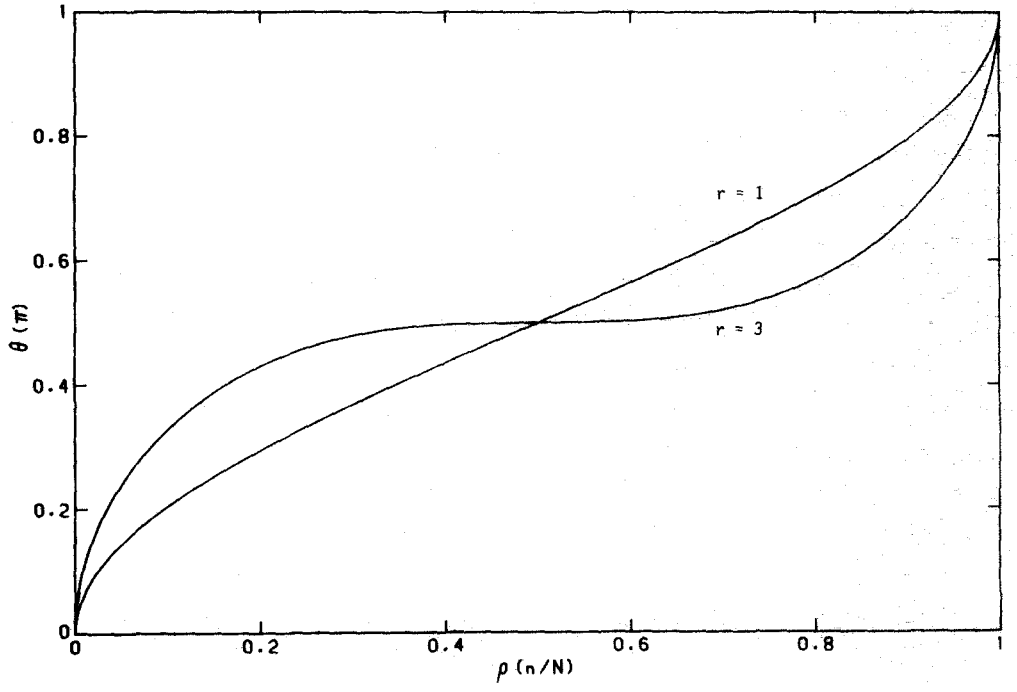
We plot θ_r versus ρ for selected orders in Figure 4 using (15). It is evident that increasing r results in better separated feature vectors. Polynomial mappings act as an effective mechanism for increasing the dimensionality of the space in which inputs are classified because they guarantee a very even distribution of the samples in this new space.

3. TRAINING OF HIGHER ORDER MEMORIES

Once the initial polynomial mapping has been selected, the rest of the system in a higher order memory is simply a linear discriminant function. As such it can be trained by any of the existing methods for training



(a)



(b)

FIGURE 3. (a) The angle between linearly and quadratically expanded vectors as a function of the hamming distance at the input; (b) The angle between linearly and cubically expanded vectors as a function of the hamming distance at the input.

linear discriminant functions. For instance the pseudoinverse (Kohonen, 1984; Venkatesh & Psaltis, in press) can be used to calculate the set of weights that will map a set of L-dimensional expanded vectors $\mathbf{z}^m$ to the associated output vectors $\mathbf{y}^m$. Alternatively, error driven algorithms such as the perceptron or adaline can be used to iteratively train the memory by repeatedly presenting the input vectors to the system, monitoring the output to obtain an error signal, and modifying the weights so as to gradually decrease the error.

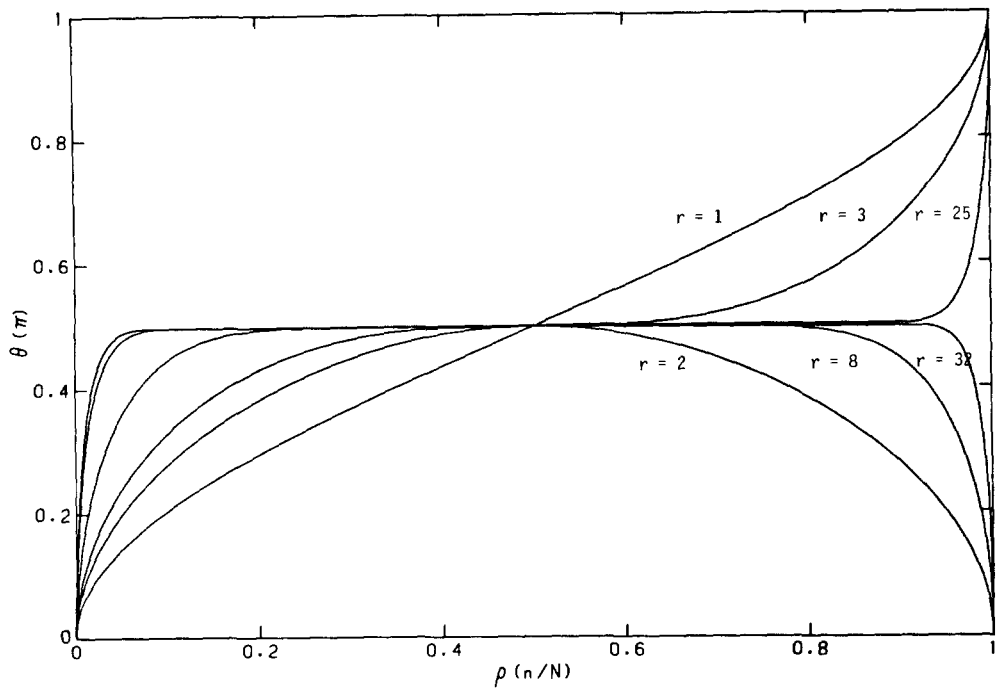


FIGURE 4. The angle between expanded vectors for selected orders.

The relative ease with which higher order memories can be trained is a very important advantageous feature of this approach. A higher order memory is basically a multilayered network where the first layer is selected a priori. In terms of capacity alone, there is no advantage whatsoever in having multiple layers with modifiable weights. From Theorem 1 we know that at best the capacity is determined by the number of modifiable weights. For a higher order memory we get the full advantage of the available degrees of freedom whereas if we put the same number of weights in multiple layers the resulting degeneracies will decrease the capacity. The relative advantage of trainable multiple layers is the potential for generalization that emerges through the learning process. The generalization properties of higher order memories on the other hand are mostly determined by the choice of the terms used in the polynomial expansion in the fixed first layer. Thus the generalization properties of these memories as described in this paper are imposed a priori by the designer of the system.

The sum of outer products algorithm that has been used extensively for training linear associative memories can also be used for training the higher order memories and this algorithm generalizes to the higher order case in particularly interesting ways. In addition, this particular learning algorithm is predominantly used for the holographic optical implementations that are described in the following section. Therefore we will discuss in some detail the properties of higher order memories that are trained using this rule.

3.1 The Outer Product Rule

Let us consider associative memories constructed as an expansion of the r-order only with input samples in an N dimensional binary space and $r \geq 1$.

$$y_l = \operatorname{sgn}\{\sum_{j_1 j_2 \cdots j_r} W_{l j_1 j_2 \cdots j_r} x_l x_{j_1} x_{j_2} \cdots x_{j_r}\}, \quad (17)$$

where $1 \leq j_1, j_2, \ldots, j_r \leq N$, $1 \leq l \leq N_0$. The number of independent terms L in the rth order expansion is $\binom{N+r-1}{r}$ which for $r \ll N$ can be approximated by $N^r/r!$

The expression for the weights of the rth order expansion using the sum of outer products algorithm is (Chen et al., 1986; Psaltis & Park, 1986)

$$W_{l j_1 j_2 \cdots j_r} = \sum_{m=1}^{M} y_l^m x_{j_1}^m x_{j_2}^m \cdots x_{j_r}^m \quad (18)$$

where M is the number of vectors stored in the memory, $\mathbf{y}^m$ is an output vector associated with an input vector $\mathbf{x}^m$ as before. With the above expression for the weight tensor (17) can be rewritten as follows

$$y_l = \operatorname{sgn}\{\sum_{m=1}^{M} y_l^m (\sum_{j=1}^{N} x_j^m x_j)^r + w_l^0\}. \quad (19)$$

The above equation suggests an alternate implementation for higher order memories that are trained using the outer product rule. This is shown schematically in Figure 5. The inner products between the input vector and all the stored vectors $\mathbf{x}^m$ are formed first, then raised to the rth power, and the signal from the mth unit is

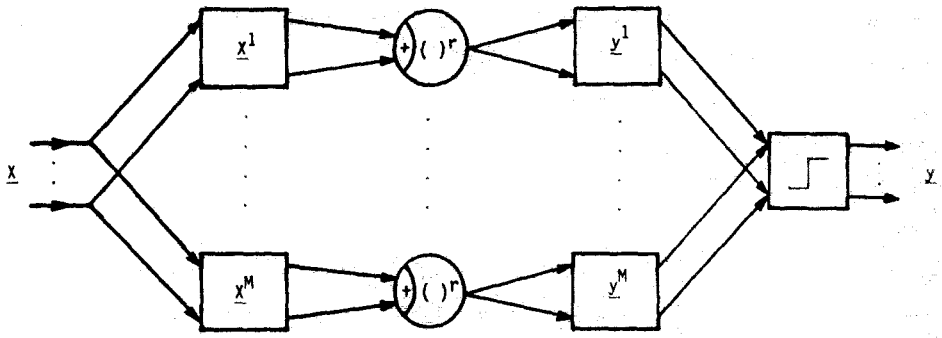


FIGURE 5. Outer product, *r*th order associative memory.

connected to the output through interconnective weights y_l^m. If $\mathbf{y}^m = \mathbf{x}^m$ then the memory is autoassociative, and in this case the output can be fed back to the input resulting in a system whose stable states are programmed to be the vectors $\mathbf{x}^m$. This becomes a direct extension of the Hopfield network (Anderson, 1983; Hopfield, 1982; Nakano, 1972) to the higher order case. Assuming that $\mathbf{x} = \mathbf{x}^n$ is one of the stored vectors, y_l becomes

$$y_l = \operatorname{sgn}\{N^r y_l^n + \sum_{m \neq n} y_l^m (\sum_{j=1}^{N} x_j^m x_j^n)^r + w_l^0\}$$

$$= \operatorname{sgn}\{N^r y_l^n + n_l(\mathbf{x}^n)\} \tag{20}$$

where the first term is the desired signal term and n_l is a noise term. The threshold weight is set to zero.

The expectation value of $n_l(\mathbf{x}^n)$ is zero if the bits that comprise the stored binary input and output vectors are drawn randomly and independently having equal probability of being +1 or −1. If this is the case then

$$E(\sum_{tt'} x_t^m x_t^{m'}) = \sum_{tt'} \delta_{tt'},$$

$$E(\sum_{mm'} x_t^m x_t^{m'}) = \sum_{mm'} \delta_{mm'} \tag{21}$$

where δ_{ij} is the Kronecker delta function. The variance of n_l is calculated as follows:

$$E(n_l^2)$$

$$= E(\sum_{m \neq n} \sum_{m' \neq n} y_l^m y_l^{m'} \sum_{j_1 j_2 \cdots j_r} \sum_{s_1 s_2 \cdots s_r} x_{j_1}^m x_{j_2}^m \cdots$$

$$x_{j_r}^m x_{j_1}^n x_{j_2}^n \cdots x_{j_r}^n x_{s_1}^{m'} x_{s_2}^{m'} \cdots x_{s_r}^{m'} x_{s_1}^n x_{s_2}^n \cdots x_{s_r}^n)$$

$$= E(\sum_{m \neq n} \sum_{j_1 j_2 \cdots j_r} \sum_{s_2 \cdots s_r} x_{j_1}^m x_{j_2}^m \cdots x_{j_r}^m x_{s_1}^m x_{s_2}^m \cdots$$

$$x_{s_r}^m x_{j_1}^n x_{j_2}^n \cdots x_{j_r}^n x_{s_1}^n x_{s_2}^n \cdots x_{s_r}^n). \tag{22}$$

In the above we used the facts that different stored vectors are uncorrelated (i.e., for $m \neq m'$) and $y_l^2 = 1$. Then, the variance becomes $(M-1)Q(N, r)$, where $Q(N, r)$ is the number of possible permutations such that

$$\delta_{i_1 t_1} \delta_{i_2 t_2} \cdots \delta_{i_r t_r} = 1 \tag{23}$$

where the set of variables $\{i_1, \ldots, i_r, t_1, \ldots, t_r\}$ spans all the combinations produced by the set of variables $\{j_1, \ldots, j_r, s_1, \ldots, s_r\}$. The variance can be calculated exactly for the cases $r = 1$, 2, and 3 and it is $(M-1)N$, $(M-1)(3N^2 - 2N)$ and $(M-1)(15N^3 - 30N^2 + 16N)$, respectively. For the general case we will derive lower and upper bounds which for large N provide us with a good estimate of the variance for any order r.

Proposition 4: The total number of permutations, $Q(N, r)$, for which (23) holds, satisfies the following relationship:

$$P(N, r)\frac{(2r)!}{2^r r!} + \binom{2r}{4} P(n, r-1)\frac{(2r-4)!}{2^{r-2}(r-2)!}$$

$$\leq Q(N, r) \leq N^r \frac{(2r)!}{2^r r!} \tag{24}$$

where $P(m, n) \equiv m!/(m-n)!$.

Proof: The number of ways of making r pairs of $2r$ items is $(2r-1)(2r-3)\cdots(3)(1) = (2r)!/2^r r!$. The items that we are concerned with are the variables i_j, t_j and each of these variables can take one of N values. We can only select the values of half these variables (N^r possibilities) and for each of these choices we can create r pairs. Hence the upperbound is $N^r(2r)!/2^r r!$. This is an upper bound because we have overcounted for different pairings of variables that have the same value.

The initial lower bound is derived if each pair has a different value from all others, which eliminates the possibility of overcounting. The number of possible ways to satisfy (23) with the variables in any two pairs not taking the same values is $P(N, r)(2r)!/2^r r!$. This is an underestimate because all pairs that contain variables taking the same value should be counted once. We can thus improve the lower bound by counting the number of ways these degenerate pairings occur and adding them into the previous bound. For example when two pairs out of r have the same values with $\binom{2r}{4}$ choices, there are $\binom{2r}{4}NP(N-1, r-2)(2r-4)!/2^{r-2}(r-2)!$ possible permutations where $(2r-4)!/2^{r-2}(r-2)!$ is the number of ways of making $r-2$ pairs of $2r-4$ items. Therefore, $Q(N, r)$ is lower bounded by $P(N, r)(2r)!/2^r r! + \binom{2r}{4}P(N, r-1)(2r-4)!/2^{r-2}(r-2)!$, since $NP(N-1, r-2) = P(N, r-1)$.

We can get a very good approximation to the *SNR* using the approximations of $M - 1 \approx M$ and $Q(N, r) \approx N^r(2r)!/2^r r!$ which are very nearly true for the interesting case $r \ll N$:

$$SNR \approx \frac{N^r}{\{MN^r(2r)!/2^r r!\}^{1/2}} = \left\{\frac{N^r}{M}\frac{2^r r!}{(2r)!}\right\}^{1/2}. \quad (25)$$

For example, the linear memory, $r = 1$, has a *SNR* $\approx (N/M)^{1/2}$, the quadratic memory, $r = 2$, a *SNR* of $N/(3M)^{1/2}$ and the cubic memory, $r = 3$, a *SNR* of $(N^3/15M)^{1/2}$. We can obtain an estimate for the capacity of an rth order memory by equating the signal to noise ratios of the linear and rth order memories and solving for M_r, the number of stored vectors that will yield the equality. For r, small compared to N, we obtain

$$\frac{M_r}{M_1} = N^{r-1}\frac{2^r r!}{(2r)!}. \quad (26)$$

Comparing its value with the capacity M_1 of a linear memory we can obtain the relationship between the capacities, that is, $M_r/M_1 = N^{r-1}2^r r!/(2r)!$. For example M_2 of a quadratic memory is $M_1 N/3$ and M_3 of a cubic memory is $M_1 N^2/15$.

The diagonal terms in a high order memory $W_{ij_1j_2\cdots j_r}$ can be defined as those of which all the indexes j are not different. We form the weight tensor with zero diagonal as follows:

$$W_{ij_1j_2\cdots j_r} = \begin{cases} \sum_m y_i^m x_{j_1}^m x_{j_2}^m \cdots x_{j_r}^m & \text{if } j\text{s are all different,} \\ 0 & \text{otherwise.} \end{cases} \quad (27)$$

When the input is one of the stored vectors $\mathbf{x}^n$ and the weight tensor has zero diagonal, the output y_l becomes

$$y_l = \operatorname{sgn}\Big\{\sum_{\text{different } j} W_{lj_1j_2\cdots j_r}x_{j_1}^n x_{j_2}^n \cdots x_{j_r}^n + w_l^0\Big\}$$
$$= \operatorname{sgn}\Big\{P(N, r)y_l^n + \sum_{m\neq n} y_l^m \sum_{\text{different } j} x_{j_1}^m x_{j_2}^m \cdots x_{j_r}^m x_{j_1}^n x_{j_2}^n \cdots x_{j_r}^n + w_l^0\Big\} \quad (28)$$

where the first term is a signal term and the second a noise term as before. The variance of the noise term is easily shown to be $(M - 1)P(N, r)r!$ using (21). Therefore, the *SNR* becomes

$$SNR = \left\{\frac{P(N, r)}{(M-1)r!}\right\}^{1/2} \approx \left\{\binom{N}{r}\Big/M\right\}^{1/2} \quad (29)$$

which can be approximated as $(N^r/Mr!)^{1/2}$ for $r \ll N$.

Chen and his coworkers (1986) introduced an energy function (Cohen & Grossberg, 1983; Hopfield, 1982) for the rth order autoassociative memory with feedback and outer products as follows:

$$E_r = -\sum_{m=1}^{M} \langle \mathbf{x}^m, \mathbf{x}\rangle^{r+1} \quad (30)$$

where $\langle \cdot, \cdot \rangle$ denotes an inner product of two vectors. The change in the energy due to a change $\delta\mathbf{x}$ in the state of the network was shown by Chen *et al.* (1986) to be decreasing for odd r.

$$\Delta E_r \equiv E_r(\mathbf{x} + \delta\mathbf{x}) - E_r(\mathbf{x}) = -(r+1)\sum_l \delta x_l \sum_{j_1\cdots j_r} W_{lj_1j_2\cdots j_r} \times x_{j_1}x_{j_2}\cdots x_{j_r} - R_r \quad (31)$$

where

$$R_r \equiv \sum_m \sum_{j=2}^{r+1} \binom{r+1}{j} \langle \mathbf{x}^m, \mathbf{x}\rangle^{r+1-j}\langle \mathbf{x}^m, \delta\mathbf{x}\rangle^j. \quad (32)$$

The first term in (31) is always nonpositive because of the specification of the update rule: $\delta x_l \geq 0$ if $\sum_{j_1\cdots j_r} \times W_{lj_1j_2\cdots j_r}x_{j_1}x_{j_2}\cdots x_{j_r} \geq 0$ and vice versa. Chen *et al.* (1986) showed that the second term is also nonpositive by showing that R_r is an increasing function of r for r odd and $R_1 > 0$.

For r even it is possible to prove the autoassociative memory converges only for asynchronous updating even though in simulations even order autoassociative memories consistently converge as well. The fact that the energy is not always decreasing when r is even may actually be helpful for getting out of local minima and settling in the programmed stable state which are global minima in a region of the energy surface. A descent procedure that is always decreasing in energy cannot escape local minima since there is no mechanism for climbing out of them. As an example, consider a quadratic memory, that is, $r = 2$ (even), whose energy function is given by

$$E_2 = -\sum_{ijk} W_{ijk}x_i x_j x_k \quad (33)$$

$$\Delta E_2 = -3\sum_{ijk} W_{ijk}x_j x_k \delta x_i - 3\sum_{ijk} W_{ijk}x_k \delta x_i \delta x_j - \sum_{ijk} W_{ijk}\delta x_i \delta x_j \delta x_k. \quad (34)$$

The first term is nonincreasing but the second and third terms can be increasing. If the vector $\mathbf{x}$ is very close to one of the stored vectors $\mathbf{x}^n$ then the first term becomes dominant and the energy will be very likely to be nonincreasing causing the system to settle at $\mathbf{x} = \mathbf{x}^n$. If $\mathbf{x}$ is not close to any of the stored vectors, then all three terms in the above equations are on the average comparable to each other and since two of them are not nondecreasing the energy function may be increasing and it is possible to escape from local minima.

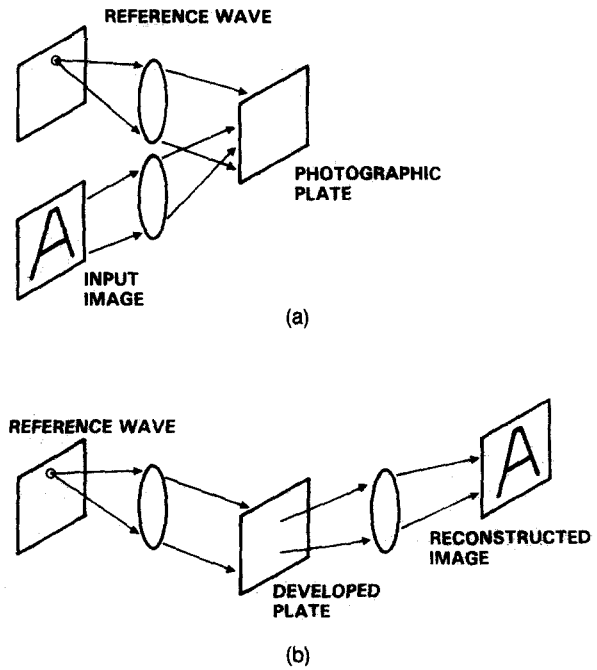


FIGURE 6. Holographic recording and reconstruction. (a) Recording, (b) reconstruction.

4. OPTICAL IMPLEMENTATIONS OF QUADRATIC ASSOCIATIVE MEMORIES

The outer product quadratic associative memories described in the previous section require three basic components for their implementation: interconnective weights, a square-law device, and a threshold nonlinearity. In this section, we present a variety of optical implementations using either planar or volume holograms to provide the interconnection pathways and optical or electro-optical devices to provide the required nonlinearities.

Since holographic techniques are used to implement the required interconnections, we will first briefly discuss holography (Collier, Burkhardt, & Lin, 1971) and in particular the distinction between the use of planar versus volume holograms. The holographic process is shown schematically in Figure 6. In the recording step (Figure 6a) the interference between the reference plane wave that is created by collimating the light from a point source using a lens and the wave originating from the object "A" is recorded on a planar light sensitive medium such as a photographic plate. When the developed plate is illuminated with the same reference wave, the field that is diffracted by the recorded interference pattern gives a virtual image of the original object which can be converted to a real image with a lens. The reconstruction of the hologram is thus equivalent to interconnecting the single point from which the plane wave reference is derived to all the points that comprise the reconstructed image. The weight of each interconnection is specified by the interference pattern stored in the hologram.

Volume holograms are prepared and used in the same manner except that whereas a planar hologram records the interference pattern as a two dimensional pattern on a plane, a volume hologram records the interference pattern throughout the volume of a three dimensional medium. The disparity in the dimensionalities of the two storage formats results in marked differences in the capabilities of the two processes. This difference is explained with the aid of Figures 7a and

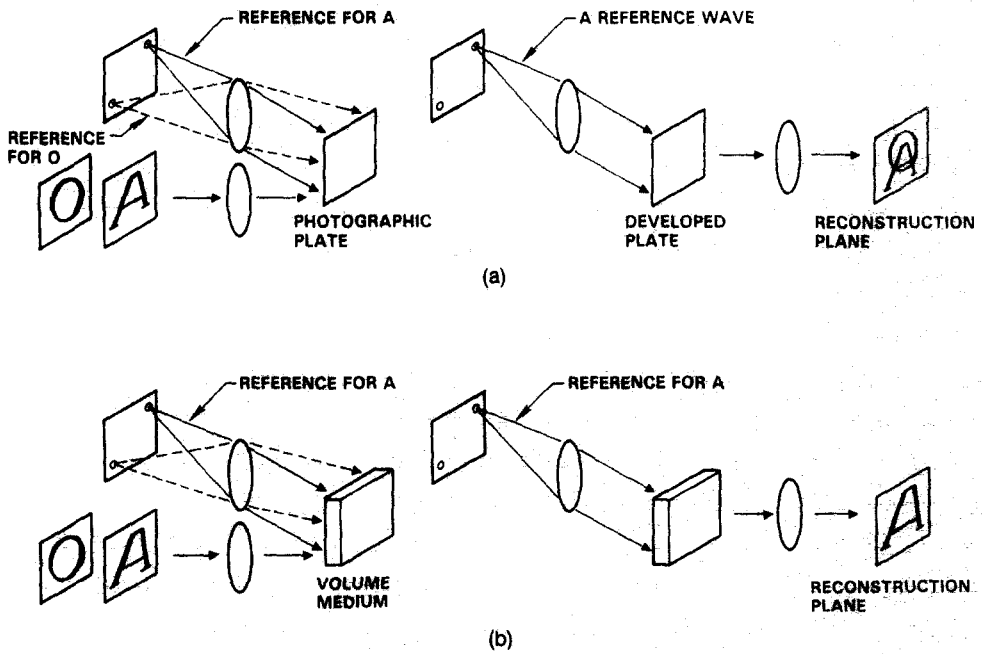


FIGURE 7. Holographic interconnections using (a) planar versus (b) volume holograms.

7b where the reconstruction of both a planar and a volume hologram are shown. Each hologram is prepared to store the two images "A" and "O" by double exposure with each image being associated with a reference plane wave that is incident on the hologram at a different angle. Each reference plane wave is generated by a separate point source and thus the reconstruction of a hologram with the two reference waves is equivalent to interconnecting multiple input points to all the points on the plane of the reconstructed image. In the case of the planar hologram, however, when either one of the reference waves is incident both images are reconstructed. This implies that we cannot in this case independently specify how each of the input points is connected to the output. In contrast, because of the interaction of the fields in the third dimension (Kogelnik, 1969) the volume hologram is able to resolve the differences in the angle of incidence of the reference beam and upon reconstruction when the reference for "A" illuminates the medium, only "A" is reconstructed and similarly for the second pattern. When both input points are on simultaneously then each is interconnected to the output independently according to the way it was specified by the recording of the two holograms. Thus volume holograms provide more flexibility for implementing arbitrary interconnections which translates to efficient three dimensional storage of the interconnective weights needed to specify the quadratic memory.

Another way in which we can draw the distinction between planar and volume holograms is in terms of the degrees of freedom. The implementation of a quadratic memory whose input word size is N bits requires approximately N^3 interconnections for the three dimensional interconnection tensor. The number of degrees of freedom of the planar hologram of area A is upper bounded by A/δ^2 while that of a volume hologram is limited to V/δ^3, where V is the volume of the crystal and δ is the minimum detail that can be recorded in any one dimension (Psaltis, Yu, Gu, & Lee, 1987; Van Heerden, 1963). Equating the degrees of freedom that are required to do the job to those that are available, the crystal volume is determined to be at least $V = N^3\delta^3$ whereas a planar hologram to do the same job would require a hologram of area $A = N^3\delta^2$. For comparison, a network with $N = 10^3$ can in principle be implemented using a cubic crystal with the length of each side being $l_v = N\delta = 1$ cm, but a square planar hologram is required to have the length of each side be at least $l_p = N^{3/2}\delta = 0.33$ m at $\delta = 10$ μm. Thus, the volume hologram offers a more compact means of implementing large memory systems.

4.1 Volume Hologram Systems

There are several schemes for fully utilizing the interconnective capability of volume holograms (Psaltis

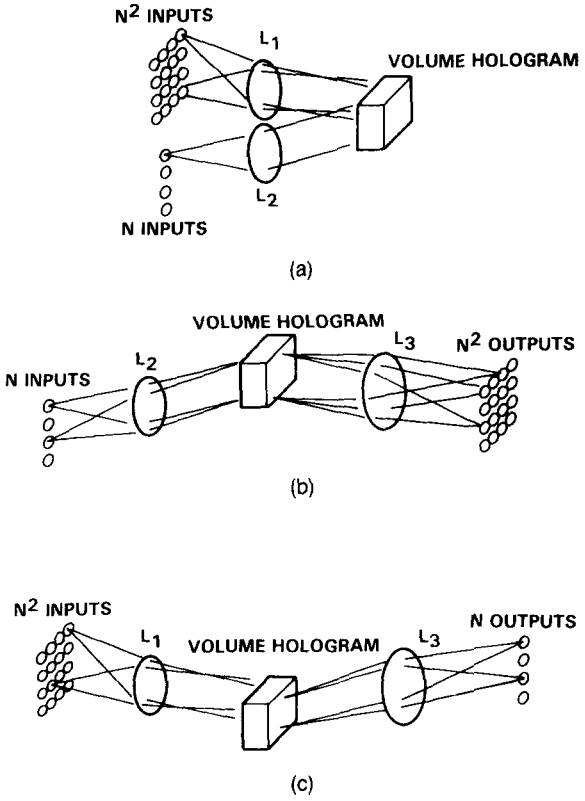


FIGURE 8. Optical interconnections using volume holograms. (a) Recording apparatus; (b) $N \mapsto N^2$ mapping; (c) $N^2 \mapsto N$ mapping.

et al., 1987; Psaltis, Brady, & Wagner, in press). For the implementation of quadratic memories we use volume holograms to fully interconnect a 2-D pattern to a 1-D pattern ($N^2 \mapsto N$ mappings) and also the reverse ($N \mapsto N^2$). The geometry for recording the weights for both cases is shown in Figure 8a and the reconstruction geometries are illustrated in Figures 8b and 8c. The circles represent the resolvable spots at the various planes in the system. The waves emanating from each point at the input planes are transformed into plane waves by the Fourier transform lenses L_1 and L_2 and interfere within the crystal, creating volume gratings.

The weights are loaded into the volume hologram with multiple holographic exposures in the system of Figure 8a. In the following subsections we will describe several specific procedures for doing so. For the $N \mapsto N^2$ mapping (Figure 8b) in reading out the stored information, a single source in the input array reconstructs one of the N 2-D images consisting of N^2 pixels that it is associated with. The rest of the images, which belong to the other input points, are not read out because of the angular discrimination of volume holograms. The counterpart to this scheme, shown in Figure 8c, implements an arbitrary $N^2 \mapsto N$ mapping. This setup is basically the same as that of Figure 8b except that the roles of the input planes have been interchanged or equivalently the direction in which light propagates has been reversed.

4.1.1 $N^2 \mapsto N$ *Schemes.* First, we consider a method by which the full three dimensional interconnection tensor is implemented directly with a volume hologram. Recall that if the weight tensor is trained using the sum of outer products then it is given by

$$w_{ijk} = \sum_{m=1}^{M} y_i^m x_j^m x_k^m, \tag{35}$$

where x_i^m represents the mth input memory vector and y_i^m represents the associated output vector. Such a memory is accessed by first creating an outer product of the input vector and multiplying it with w_{ijk} as follows:

$$y_i = \operatorname{sgn}\{\sum_{j=1}^{N} \sum_{k=1}^{N} w_{ijk} x_j x_k\}. \tag{36}$$

The volume hologram is prepared using the setup in Figure 8a. First, the outer produet matrix of the mth memory input vector, $x_j^m x_k^m$, is formed on an electronically addressed spatial light modulator (SLM) (Warde & Fisher, 1987). Another one-dimensional SLM whose transmittance represents the mth output vector y_i^m is placed in the other input plane, and the two SLMs are illuminated by coherent light. The transmitted waves are then Fourier transformed by lenses L_1 and L_2 to interfere within the crystal volume to create index gratings. This procedure is repeated for all M associated input–output pairs so that a sum of M holograms is created in the crystal. For the quadratic outer product memory whose capacity is fully expended, this involves on the order of $N^2/\log N$ exposures.

We will now describe another method for recording the weight vector in the volume hologram that involves fewer exposures and can also be used not only for the outer product scheme but for recording any given weight tensor as well. The same basic recording architecture of Figure 8a is used in this case also. In the first exposure, the top light source in the linear array is turned on while the SLM is programmed with the matrix w_{1jk}, where w_{ijk} is the interconnection tensor. When the SLM is illuminated with light coherent with that of the point source, the crystal records the mutual interference pattern as a hologram of the image w_{1jk} with a reference beam that is the plane wave generated from the top light source. In the next step, the second source is turned on while the SLM is programmed with the matrix w_{2jk}. In this manner the connectivity for all the points in the linear array at the input are sequentially specified and the memory training is completed when all N exposures have been made. The disadvantage of this method relative to the outer product recording is the need to precalculate electronically the weight tensor but it has the advantage of fewer exposures (N versus $N^2/\log N$) and greater flexibility in choosing the training method.

The architecture in Figure 8c is used to access the data stored in the hologram by either one of the recording methods described above. The electronically addressed 2-D SLM is placed at the input plane and it is programmed with the outer product matrix $x_k x_j$ of the input vector. The light from the N^2 input points is interconnected with the N output points via the recorded w_{ijk} interconnect kernel. A linear array of N photodetectors is positioned to sample the output points.

It is important to restate at this juncture that this particular implementation achieves the quadratic interconnections by first transforming the N input features (i.e., the N elements of the input vector x_j) into a set of N^2 features via the outer product operation. The result is that although the interconnections are quadratic with respect to the N original feature points, they are linear with respect to the N^2 transformed features. This allows the application of error driven learning algorithms for linear networks such as the *Adaline* (Widrow & Hoff, 1960) where the interconnections are developed by an iterative training process. The operation of such a learning scheme is illustrated in Figure 9 which is the same basic architecture as Figure 8c with feedback from the output back into one of the input ports. Each iteration consists of a reading and a writing phase. During the reading phase, the interconnections present in the crystal are interrogated with a particular item to be memorized by illuminating the 2-D SLM which contains the outer product matrix $\mathbf{x}^m\mathbf{x}^{mt}$ and the output is formed on the detector array. In the subsequent writing phase, the error pattern generated by subtracting the actual output from the desired output pattern is loaded into the 1-D SLM and both SLMs (the 2-D SLM still contains $\mathbf{x}^m\mathbf{x}^{mt}$) are illuminated with coherent light, forming a set of gratings in addition to the previously recorded gratings. The procedure is iteratively repeated for each item to be memorized until the output error is sufficiently small. This algorithm is a descent procedure designed to minimize the mean squared cost $\epsilon = \frac{1}{M} \sum_{m=1}^{M} [\sum_{j=1}^{N} \sum_{k=1}^{N} w_{ijk} x_j^m x_k^m - y^m]^2$ by iteratively updating the interconnection values.

4.1.2 $N \mapsto N^2$ *Schemes.* The $N \mapsto N^2$ mapping capability of the volume hologram which is the inverse of that required for the architectures just described can be used also to implement quadratic memories and can be generalized for higher order memories. The basic idea behind this scheme is illustrated in Figure 10 which shows the interconnection between the ith and jth neurons whose weight w_{ij} is a linear combination of all of the inputs and is described by

$$w_{ij} = \sum_{k=1}^{N} \hat{w}_{ijk} x_k. \tag{37}$$

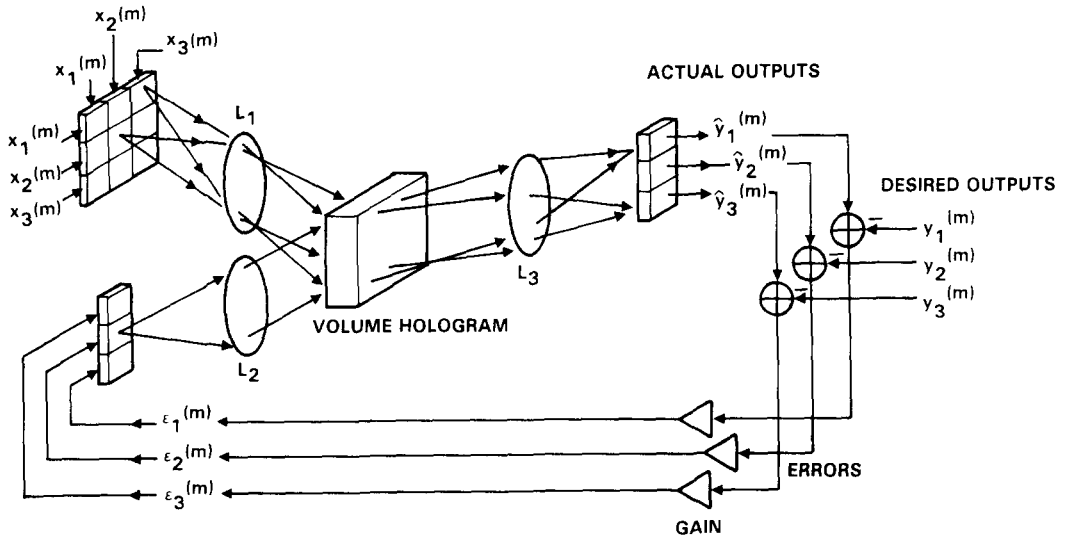


FIGURE 9. Optical system for performing error driven learning in a higher order memory.

The overall result is, of course, recognized to be the equation describing the quadratic memory, but the notion of an input dependent weight suggests the implementation shown in Figure 11. The system is basically an optical vector matrix multiplier (Goodman, Dias, & Woody, 1978) in which the matrix is created on an optically addressed SLM by multiplying the input vector with the three-dimensional tensor stored in a volume hologram. The input vector is represented by a one dimensional array of light sources. The portion of the system on the left side of the SLM is the vector matrix multiplier and it works as follows. Light from each input point is imaged horizontally but spread out vertically so that each source illuminates a narrow, vertical area on the 2-D SLM. The reflectance of the SLM corresponds to the matrix of weights w_{ij} in (37). The reflected light from the SLM travels back towards the input and a portion of it is reflected by a beam splitter and then imaged horizontally but focused vertically onto a 1-D output detector array. The output from the detector array represents the matrix vector product between the input vector and the matrix represented by the 2-D reflectance of the SLM. The matrix of weights, in this case, is not fixed but rather computed from the input via a volume hologram by exposing the righthand side of the SLM as shown in the figure. The optical system to the right of the 2-D SLM in Figure 11 is the same as the $N \mapsto N^2$ system of Figure 8b. The volume hologram which has been prepared to perform the appropriate dimension increasing operation ($N \mapsto N^2$), transforms the light distribution given by its one dimensional array of sources into the input dependent matrix of weights given by (37). This system is functionally equivalent to the previous system except it does not require the use of a 2-D electronically addressed input SLM. The 1-D devices utilized in this architecture are easier and faster to use in practice. Instead a 2-D optically addressed SLM is needed which in practice is simpler to use compared to electronically addressed devices (requires less electronics), typically has more pixels, and is potentially much higher speed. A disadvantage of this method, however, is that it does not lend itself for the direct implementation of the simple outer product training method without the use of an electronically addressed 2-D SLM.

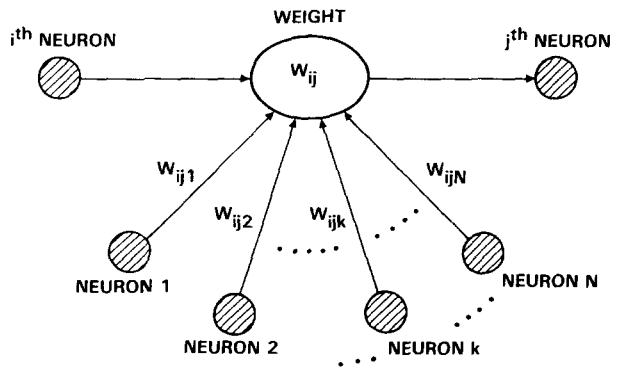


FIGURE 10. Quadratic mappings implemented as nonlinear interconnections.

The $N \mapsto N^2$ mapping technique can be used in conjunction with its inverse, the $N^2 \mapsto N$ mapping, to implement the quadratic outer product memory using two volume holograms, a 1-D electronically addressed SLM, and an optically addressed 2-D SLM. Shown in Figure 12 is a schematic diagram of such a system. The first hologram is prepared with the multiple exposure scheme discussed earlier (Figure 8a) where for each exposure, a memory vector in the one-dimensional input array and one point in the two-dimensional ($\sqrt{M} \times \sqrt{M}$) input training array are turned on simultaneously. The second hologram is prepared by a similar procedure except that the associated output vectors are

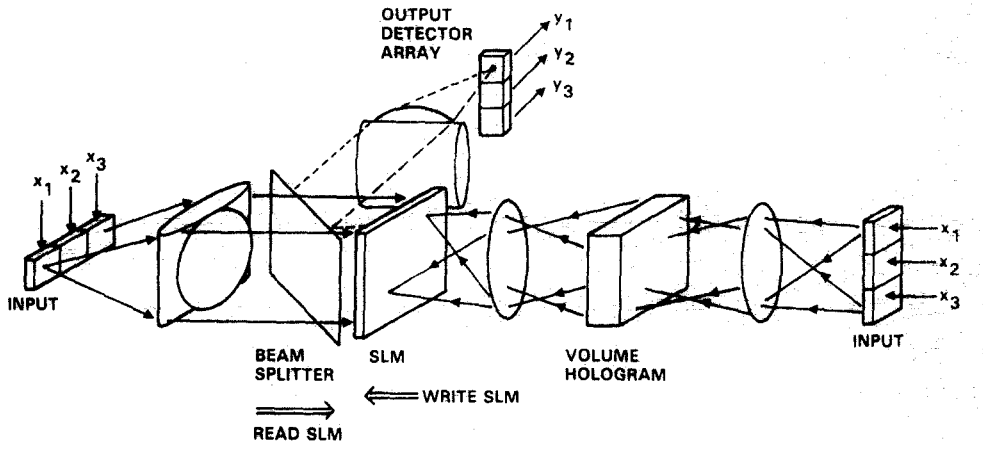


FIGURE 11. Optical architecture for the implementation of the nonlinear interconnections of Figure 10.

recorded in correspondence to each point in the two dimensional training plane. After the holograms are thus prepared, an input vector is loaded into the one-dimensional input array and the correlations between it and the M memory vectors are displayed in the output plane (Athale, Szu, & Friedlander, 1986; Owechko, Dunning, Marom, & Soffer, 1987; Paek & Psaltis, 1987). An optically addressed SLM can be used to produce an amplitude distribution which is the square of the incident correlation amplitudes. The processed light then illuminates the second hologram which serves as an $M \mapsto N$ interconnection, each correlation peak in the SLM plane reading out its corresponding memory vector and forming a weighted sum of the stored memories on the one dimensional output detector array. This is a direct optical implementation of the system shown in block diagram form in Figure 5 with the 2-D SLM performing the square law nonlinearity at the middle plane and the two-volume holograms providing the interconnections to the input and output.

4.2 Planar Hologram Systems

While not having the extra dimension to directly implement the three dimensional interconnection tensor for general quadratic memories, planar holograms can nevertheless implement the outer product quadratic memory in a way similar to the one used in the system just described. The planar holographic system is shown in Figure 13. Here, the information is stored in the two multichannel 1-D Fourier transform (FT) holograms, the first of which contains the 1-D FTs of the M memory input vectors and the other, the FTs of the associated output vectors (Psaltis & Hong, 1987). The first part of the system is a multichannel correlator which correlates the input against each of the M memory vectors. At the correlation plane, the M correlation functions stacked up vertically are sampled at $x = 0$ with a slit to obtain the required inner products which are then squared by the SLM. Each resulting point source of light is then collimated horizontally and imaged vertically onto the second hologram to illuminate that portion which contains the corresponding output vector. The final stage computes the FT of the light distribution just following the second hologram to produce the weighted sum of the vectors at the output detector array. It is interesting to note that if the SLM is removed from the correlation plane, this system reduces to the linear outer product memory.

Notice that in this system if the input pattern shifts horizontally then the correlation peak also shifts in the correlation plane and it is blocked by the slit that is placed there. Therefore shifted versions of the input vector are not recognized, as expected. Shift invariance where the shifted versions of the memory vectors are recognized and their associated outputs, shifted by the

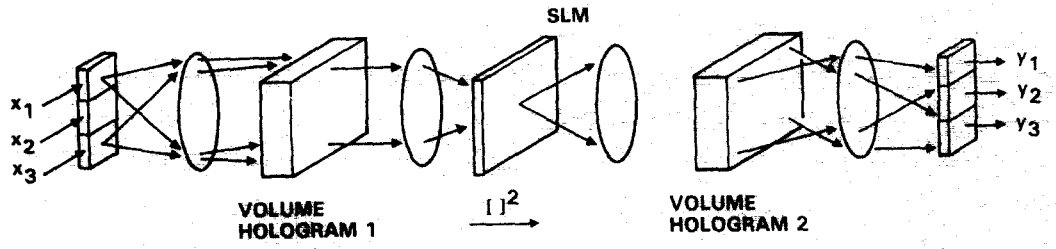


FIGURE 12. Optical higher order associative memory implemented with volume holograms.

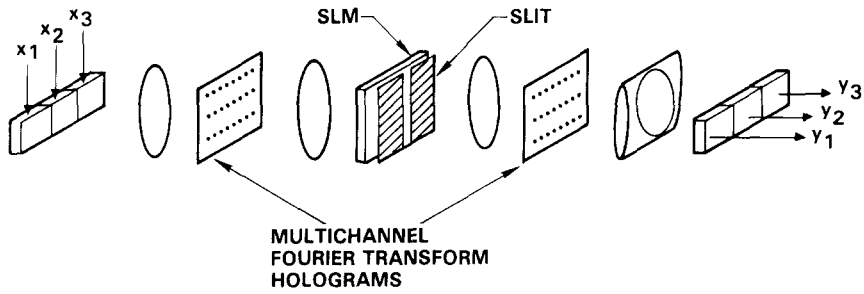


FIGURE 13. Optical implementation of the outer product higher order memory.

same amount as the input, are retrieved can be built into this system by simply lengthening the input SLM and the output detector array to accommodate the shifts and removing the slit in the correlation plane. The resulting system treats each of the $2N - 1$ shifted versions of the memory vectors as a new memory and as a result, the increased capacity of the quadratic memory over the linear one (by a factor of N) is expended to provide invariant operation.

REFERENCES

Abu-Mostafa, Y., & Psaltis, D. (1985). Computation power of parallelism in optical architectures. In *IEEE Computer Society Workshop on Computer Architecture for Pattern Analysis and Image Database Management* (p. 42). Silver Spring, MD: IEEE Computer Society Press.

Anderson, J. A. (1983). Cognitive and psychological computation with neural models. *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-13**, 799.

Athale, R. A., Szu, H. H., & Friedlander, C. B. (1986). Optical implementation of associative memory with controlled nonlinearity in the correlation domain. *Optics Letters*, **11**(7), 482.

Baldi, P., & Venkatesh, S. S. (1987). Number of stable points for spin-glasses and neural networks of higher orders. *Physics Review Letters*, **58**(9), 913.

Chen, H. H., Lee, Y. C., Maxwell, T., Sun, G. Z., Lee, H. Y., & Giles, C. L. (1986). High order correlation model for associative memory. In J. Denker (Ed.), *AIP Conference Proceedings* (p. 86). New York: American Institute of Physics.

Cohen, M., & Grossberg, S. (1983). Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-13**, 815.

Collier, R. J., Burkhardt, C. B., & Lin, L. H. (1971). *Optical holography*. New York: Academic Press.

Cover, T. M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, **EC-14**, 326.

Duda, R., & Hart, P. (1973). *Pattern classification and scene analysis*. New York: Wiley.

Giles, C. L., & Maxwell, T. (1987). Learning and generalization in higher order networks. *Applied Optics*, **26**(23), 4972.

Goodman, J. W., Dias, R. A., & Woody, L. M. (1978). Fully parallel, high speed incoherent optical method for performing discrete Fourier transforms. *Optics Letters*, **2**(1), 1.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, **79**, 2554.

Kogelnik, H. (1969). Coupled theory for thick hologram gratings. *Bell Systems Technical Journal*, **48**, 2909.

Kohonen, T. (1984). *Self-organization and associative memory*. New York: Springer Verlag.

Maxwell, T., Giles, C. L., Lee, Y. C., & Chen, H. H. (1986). Nonlinear dynamics of artificial neural systems. In J. Denker (Ed.), *AIP Conference Proceedings* (p. 299). New York: American Institute of Physics.

Nakano, K. (1972). Association—A model of associative memory. *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-2**, 380–388.

Newman, C. M. (1987, November). *Memory capacity in symmetric neural networks: Rigorous bounds*. Paper presented at the IEEE Conference on "Neural Information Processing Systems—Natural and Synthetic," Denver, CO.

Owechko, Y., Dunning, G. J., Marom, E., & Soffer, B. H. (1987). Holographic associative memory with nonlinearities in the correlation domain. *Applied Optics*, **26**(10), 1900.

Paek, E. G., & Psaltis, D. (1987). Optical associative memory using Fourier transform holograms. *Optical Engineering*, **26**(5), 428.

Poggio, T. (1975). On optimal nonlinear associative recall. *Biological Cybernetics*, **19**, 201.

Psaltis, D., Brady, D., & Wagner, K. (in press). Adaptive optical networks using photorefractive crystals. *Applied Optics*.

Psaltis, D., & Hong, J. (1987). Shift-invariant optical associative memories. *Optical Engineering*, **26**(1), 10.

Psaltis, D., & Park, C. H. (1986). Nonlinear discriminant functions and associative memories. In J. Denker (Ed.), *AIP Conference Proceedings* (p. 370). New York: American Institute of Physics.

Psaltis, D., Park, C. H., & Hong, J. (1986). Quadratic optical associative memories. *Journal of the Optical Society of America—A*, **3**(13), 32.

Psaltis, D., Yu, J., Gu, X. G., & Lee, H. (1987). Optical neural nets implemented with volume holograms. In *Proceedings of OSA Second Topical Meeting on Optical Computing* (p. 129). Incline Village, NV: Optical Society of America.

Sejnowski, T. (1986). High-order Boltzmann machines. In J. Denker (Ed.), *AIP Conference Proceedings* (p. 398). New York: American Institute of Physics.

Slepian, D. (1956). A class of binary signaling alphabets. *Bell Systems Technical Journal*, **35**, 203.

Van Heerden, P. J. (1963). Theory of optical information storage in solids. *Applied Optics*, **2**(4), 393.

Venkatesh, S. S., & Psaltis, D. (in press). Linear and logarithmic capacities of associative memories. *IEEE Transactions on Information Theory*.

Warde, C., & Fisher, A. D. (1987). Spatial light modulators: Applications and functional capabilities. In J. L. Horner (Ed.), *Optical signal processing* (p. 477). San Diego: Academic Press.

Widrow, B., & Hoff, M. E. (1960). Adaptive switching circuits. *IRE Wescon Convention Record, Pt. 4*, 96.