An Optical Implementation of Associative Memory Based on Inner Product Neural Network Model

*S.K.Gil, J.J.Eun, H.K.Park

Dept. of Eletronic Eng., Yonsei Univ.

ABSTRACT

In this paper, we propose a hybrid optical/ digital version of the associative memory which improve hardware efficiency and increase convergence rates. Multifocus hololens are used as space- varient optical element for performing inner product and summation function. The real-time input and the stored states of memory matrix is formated using LCTV. One method of adaptively changing the weights of stored vectors during each iteration is implemented eletronically. A design for a optical implementation scheme is discussed and the proposed architecture is demonstrated the ability of retrieving with computer simmulation.

1. Introduction

Since Hopfield further extended to structure a computational model by a notion of energy function with an outer product model,[1] much recent works in optical computing have concentrated on the associative memory.[2-3] These efforts in the field of associative memory, however, suffer from low storage capacity, low convergence rates, etc.

To overcome these problems, higher-order nonlinear discriminant function models have been proposed without implementation.[4-5] And a simple implementation of the quadratic associative memory with outer product storage was reported.[6-7] Kung and Liu proposed an optical inner product array processor[8] for associative retrieval who reduced memory size compared to Hopfield's outer product model.

In our hybrid approach which takes the advantage of optics (linear transformation, massive inter-connectivity, parallelism, high speed) and eletronics (point nonlinearities, programmability), we implement an associative memory based on inner product neural network model using multifocus hololens and LCTV.

2. Inner product neural network model

The simplest model of a associative memory is replesented by Fig.1. This architecture is descrived mathematically in the following equations.

$$M_M = \sum_{i=1}^{M} V^{(i)} U^{(i)^T} \quad : \text{Recording} \qquad (2\text{-}1)$$

$$V = M_M \tilde{V} \quad : \text{Retrieval} \qquad (2\text{-}2)$$

where { $U^{(i)}$, $V^{(i)}$} is a set of the stored vectors for i=1,2,...,M, $\tilde{V}$ is imperfect input vector, V is retrieved output vector, and M_M is memory matrix.

If the data vectors are chosen to be binary pattern composed of N bits for all i,

$$V^{(i)} = (v_1^{(i)}, v_2^{(i)}, \ldots, v_N^{(i)}), \; i=1,2,\ldots,M \qquad (2-3)$$

where $v_j^{(i)} = 0$ or 1 for j=1,2,...,N.

We propose an modified autoassociative memory scheme in which Eq.(2-1) and Eq.(2-2) can be subsequently decomposed into two steps as follows.

Forward step;

$$X^{(i)} = V^{(i)}\tilde{V} \;, \; i=1,2,\ldots,M \qquad (2-4)$$

$$x^{(i)} = \sum_{j=1}^{N} X^{(i)} \qquad (2-5)$$

where $X^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_j^{(i)}, \ldots, x_N^{(i)})$

$$w = f[\, x \,] \qquad (2-6)$$

where f[ ]:nonlinear transformation function

Backward step;

$$y^{(i)} = w^{(i)} h \qquad (2-7)$$

where h:constant scalar function

$$Z = \sum_{i=1}^{M} V^{(i)} y^{(i)} \, (Z^{(i)} = V^{(i)} y^{(i)}) \qquad (2-8)$$

$$V = TH[Z] \qquad (2-9)$$

where TH[ ]:threshold operation

The resultant equation is given by

$$Z = \sum_{i=1}^{M} V^{(i)} f[\sum_{j=1}^{N} V^{(i)}\tilde{V}]h \qquad (2-10)$$

The forward step performs an inner product between the imperfect input vector $\tilde{V}$ and all the stored vectors $V^{(i)}$ in parallel. The resultant inner products $x^{(i)}$ are transformed via the nonlinear transformation function f[], that could be different and used as a coefficient $w^{(i)}$ in a linear summation of the corresponding stored vectors in backward step. After adaptive threshold operation TH[], the retrieved output vector is an estimate of the stored vector which is the closest to the partial or imperfect input vector $\tilde{V}$. This estimate is fed back to the forward step and the procedure is iterated until a stable state is reached.

A block diagram of our proposed associative memory is shown in Fig.2 which contains two steps connected in a loop with nonlinearities in between them.

3. Optical implementation

A detailed schematic of the hybrid autoassociative memory in Fig.2 can be implemented shown in Fig.3.

Its implementation depends on the physical availability of the correlation values and on the possibility of the point nonlinearities. The use of LCTV in an inner product associative memory simplifies interfacing to a computer and allows the implementation of nonlinear transformation function in real time.

For autoassociative memory retrieval the stored matrix formats $V^{(i)}$, $U^{(i)T}$ are identical. This fact can be utilized to design an optical autoassociative memory with bi-directional propagation of light and a common memory matrix. Thus the potential usage of LCTV is resonable as a memory on which different weights of nonlinearities can be displayed with different grey levels simultaneously. Also, an inner product and a linear summation can be performed using multifocus hololens which is capable of replicating an image. The fabrication of multifocus hololens is shown in Fig.4, and Fig.5 shows an 2 x 2 image pattern through multifocus hololens for example.

The procedure of associative retrieval is performed with forword and backward steps.

4. SNR improvement

The retrieved output vector before threshold operation is given by Eq.(2-10). If the k-th vector in the stored memory vectors is the desired vector which has the largest inner product or correlation with input vector, Eq.(2-10) can be expressed by

$$Z = Z^{(k)} + \sum_{i \neq k}^{M} Z^{(i)} \qquad (4\text{-}1)$$

Rewriting Eq.(4-1) in terms of matrix elements when h = 1,

$$\begin{aligned} z_j &= v_j^{(k)} f[\sum_{j=1}^{N} v_j^{(k)} \tilde{v}_j] + \sum_{i \neq k}^{M} v_j^{(i)} f[\sum_{j=1}^{N} v_j^{(i)} \tilde{v}_j] \\ &= v_j^{(k)} f[\, C_j^{(k)}] + \sum_{i \neq k}^{M} v_j^{(i)} f[\, C_j^{(i)}] \\ &= s_j + n_j \end{aligned} \qquad (4\text{-}2)$$

where $C_j^{(i)}$ is the correlation between $\tilde{V}$ and the stored vectors $V^{(i)}$.

According to this expansion, the first term is the "signal" and the second is the "noise" term. Convergence rate relys on the relative importance of the signal and the noise.

If the signal to noise ratio (SNR) is defined as

$$\begin{aligned} SNR &= E\{ s_j \} \,/\, E\{ | n_j |^2 \} \\ &= \eta \,/\, \sigma \end{aligned} \qquad (4\text{-}3)$$

We propose to improve the signal to noise ratio by suppressing the small correlations with respect to the large correlations. Since the cross-talk among stored vectors can be reduced to small value by increasing the nonlinearity in the correlation domain, a n-th power law nonlinearity increase the difference between the auto-correlation term and the cross-correlation terms. This nonlinear mechanism performs functions analogous to "winner-take-all" competitive neural networks.

In Eq.(4-2), then nonlinear transformation function f[] can be used to suppress the noise by applying a n-th power law which is chosen in Fig.6.

Thus Eq.(4-2) may be rewritten

$$z_j = v_j^{(k)} [\, C_j^{(k)}]^n + \sum_{i \neq k}^{M} v_j^{(i)} [\, C_j^{(i)}]^n \qquad (4\text{-}4)$$

Using of the central limit theorem leads to the following signal to noise ratio.

$$SNR = \left[\frac{n!}{(2n)!} \times \frac{N}{M-1} \right]^{1/2} \qquad (4\text{-}5)$$

This exppression for n = 1 is concurred with the SNR in the Hopfield model and a nonlinear transformation function using power law improves the SNR for larger n. Fig.7 (a) and (b) shows the SNR with respect to power n when the total number of bits is 16. But in our proposed architecture, the SNR is increased during each iteration. This is the reason why the continuing forward and backward procedures increase the difference of the weights among stored vectors. The SNR of our architecture is exppressed by

$$SNR = \left[\frac{n!}{(2n)!} \times \frac{N}{M-1} \right]^{t/2} \qquad (4\text{-}6)$$

where t is iteration times.

If we require that SNR $\geq$ 1, then we get the storage capacity bound of

$$M \leq 1 + \frac{n!}{(2n)!} \times N^n \qquad (4\text{-}7)$$

The maximum storage capacity of memory M when SNR = 1 is shown in Fig.8 (a) and (b).

5. Simulations

The proposed hybrid associative memory was simulated on a personal computer, which show the effects of various nonlinearities on associative memory convergence rates.

The four vectors, each 16-bit, to be stored are

$V^{(1)}$= (1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0)

$V^{(2)}$= (1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0)

$V^{(3)}$= (1,0,1,0,0,1,0,1,1,0,1,0,0,1,0,1)

$V^{(4)}$= (1,0,0,1,0,1,1,0,0,1,1,0,1,0,0,1)

The threshold operation TH[] is set to have a mean value of auto-correlation and cross-correlations, and it is altered adaptively by particular input.

An use of nonlinearities and an iterative procedure retrieve one of the stored vectors correctly. The weights of the stored vectors can be altered by changing the nonlinear transformation function. Even if the nonlinear transformation fuction is fixed, the weights of the stored vectors is altered via the forward and backward iterations. The convergence rate is increased if the nonlinear transformation function has higher order power law.

Some examples of computer simulation with our architecture is illustrated as follows:

(Example 1) power law ; n = 1

Input vector : 1111000011110001

1-st Iteration : 1011000011100000

6-th Iteration : 1111000011110000 ; $V^{(1)}$

(Example 2) power law ; n = 2

Input vector : 1111000011110001

1-st Iteration : 1011000011100000

2-nd Iteration : 1111000011110000 ; $V^{(1)}$

(Example 3) power law ; n = 1

Input vector : 1010001011101100

1-st Iteration : 1010001010101000

9-th Iteration : 1010101010101010 ; $V^{(2)}$

(Example 4) power law ; n = 2

Input vector : 1010001011101100

1-st Iteration : 1010001010101000

4-th Iteration : 1010101010101010 ; $V^{(2)}$

(Example 5) power law ; n = 2

Input vector : 0011000011101010

1-st Iteration : 1011001011101000

; false stable vector

6. Conclusions

We have proposed a hybrid optical/digital version of the associative memory based on inner product neural network model which consists of LCTV, multifocus hololens, photo-detector, CCD and computer. We have demonstrated the ability of retrieving the complete stored vector from memory by computer simulation. Considering the nonlinear weights, we have improved the signal to noise ratio and the maximum number of storage capacity. Our proposed architecture provides the flexibility of rapidly and arbitrary changing the weights of the stored vectors in an associative memory and the advantage of iteration time and memory size. And it can be applied to adaptive information processing and control system as well as pattern recognition.

References

1. J.J.Hopfield, Proc. Natl. Acad. Sci. U.S.A., 79, 2554, 1982
2. N.Fahrat, D.Psaltis et al, Appl. Opt. 24, 1469, 1985
3. B.H.Soffer, G.J.Dunning, et al, Opt. Lett., Vol.11, 118, 1986
4. Y.S.Abu-Mostafa and J.-M.S.Jacques, IEEE Trans. Inf. Theory IT-31, 461,1985
5. D.Psaltis and J.Hong, Opt.Eng. 26, 10, 1987
6. J.S.Jang, S.Y.Shin, S.Y.Lee, Opt. Lett. Vol.13, 8, 693, 1988
7. S.H.Song and S.S.Lee, Appl. Opt. 27, 3149,1988
8. H.K.Liu and S.Y.Kung, Opt. Eng. 25, 853, 1986
9. R.A.Athale, H.H.Szu and C.B.Friedlander, Opt. Lett. 11, 482, 1986

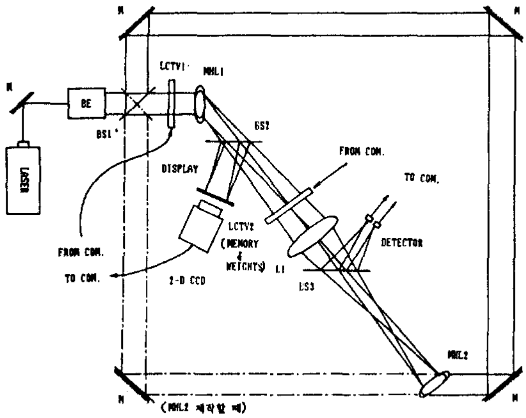
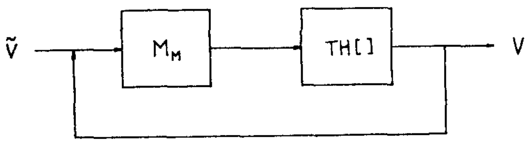


Fig.3 Optical implementation of hybrid associative memory using inner product



(M : memory matrix for associative retrieval
TH[ ] : thresholding operration)

Fig.1 Associative memory model

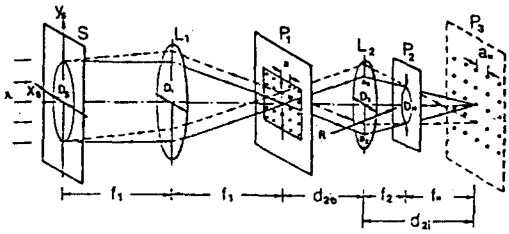


Fig.4 Fabrication of multifocus hololens

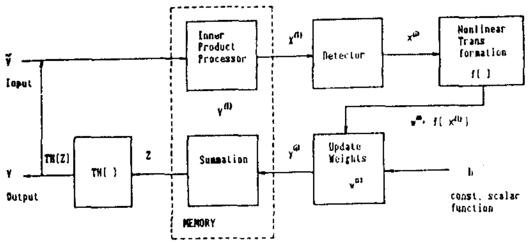


Fig.2 Block diagram of hybrid associative memory

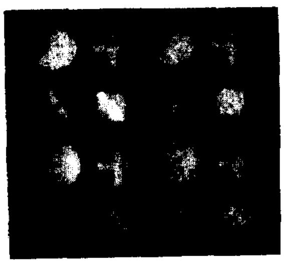


Fig.5 Multiple images from MHL

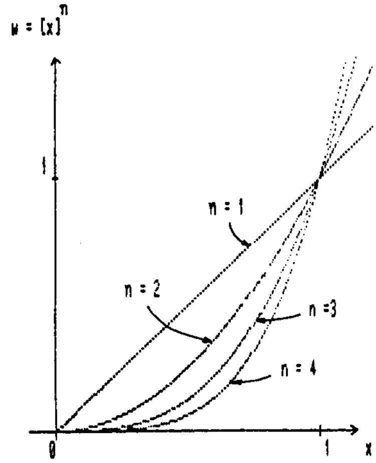
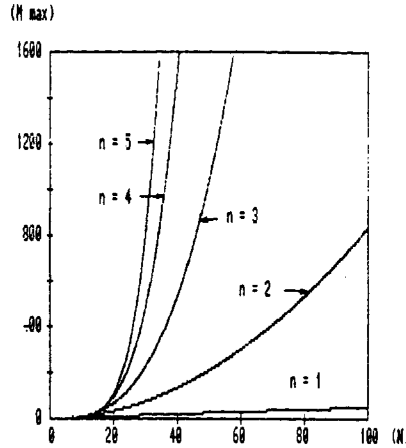
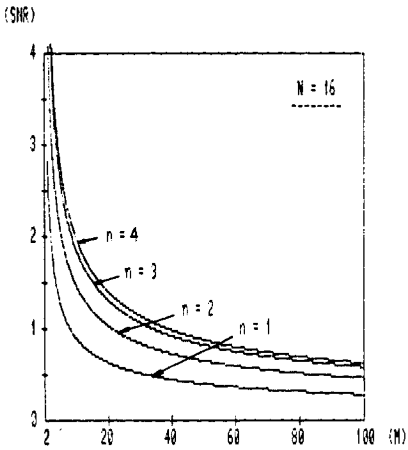


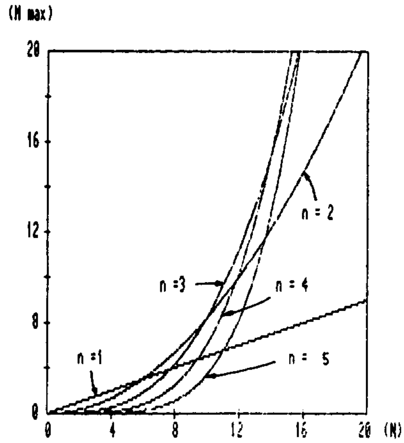
Fig.6 Nonlinear transformation function
with n-th power law



(a)

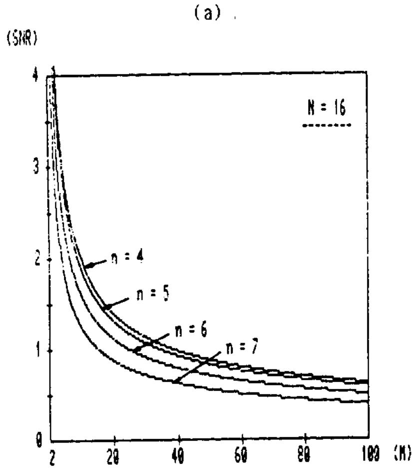


(a)



(b)

Fig.8 Maximum storage capacity of memory M



(b)

Fig.7 SNR with respect to power n