Associative Memory for Gray-scale Images

Sitabhra Sinha Sambhunath Biswas
Machine Intelligence Unit
Indian Statistical Institute
203 B. T. Road, Calcutta-700035, INDIA
E-mail: res9425@isical.ernet.in, sambhu@isical.ernet.in

Abstract

Associative memory for gray-level images with bit-plane representation has been considered. Faithful recall is achieved from the noisy and the occluded version of the stored images. The method is suitable for parallel implementation and superior to some existing techniques.

1 Introduction

Associative memory, also referred to as content-addressable memory is the storage and recall of information by association with other semantically related information. An example is the way the human brain manages to recall some memory on the basis of only partial or imperfect knowledge of its content [1]. Human memory is not organized in the same way as the traditional computers, where the recall of information is based on the precise knowledge of memory address and the data base management for handling these memory addresses.

Let us assume that m binary patterns, each having N bits of information, ξ_i^μ $(i = 1, \ldots, N;\ \mu = 1, \ldots, m)$ are stored in an associative memory. Then, for an input pattern, ζ_i $(i = 1, \ldots, N)$, the stored pattern, ξ^μ, is recalled for which the mean square deviation is minimal. Note that, due to the binary representation of patterns, the mean square deviation is the Hamming distance d_H for the pair (ξ^μ, ζ) defined as

$$d_H = \sum_i (\zeta_i - \xi_i^\mu)^2 \qquad (1)$$

This problem can be solved in principle, in a conventional computer, by calculating all possible values of d_H and then searching for the smallest one. However, the computational complexity of this procedure increases exponentially with the number of stored patterns, and the time required for searching out the smallest d_H becomes impractical even for a modest value of m. In neural networks, this process is sought to be achieved by the evoution of the network state from the initial pattern to the final desired pattern under its own dynamics.

Neural network models of computation have already been used with some success in associative recall of *binary* images [2], and the Hopfield model [3], being very popular for its ease, is the most commonly used model for such problems. However, although Hopfield's network is most appropriate for binary images (including ASCII text), it is unsuitable for gray-images due to the representatation problem. In particular, it is not possible to use the standard connection weight matrix computation technique if the images are not binary. In this paper, we have attempted to overcome this representation problem by the application of gray-scale decomposition scheme.

The paper is organized as follows. We provide a brief description of the model in section 2. The gray-level image decomposition technique, and the algorithm for storage of patterns in the network and their subsequent recall are also provided. In Section 3, the simulation results of the proposed method are described.

2 Description of Model

The Hopfield model is a globally coupled network of N threshold activated nodes having binary-valued outputs (0 or 1), whose state at time t is denoted by $x_i(t)$ $(i = 1, \ldots, N)$. The discrete time-evolution equation of the nodes is

$$x_i(t) = \mathbf{F}\Big(\sum_j w_{ij} x_j(t-1) - \theta_i\Big) \qquad (2)$$

where, w_{ij} is the connection weight between the ith and jth nodes, and θ_i is the threshold for the ith node. $\mathbf{F}$ is the input-output transfer function. Usually it is taken to be a step function :

$$\mathbf{F}(z) \begin{array}{ll} = 0 & \text{if } z < 0 \\ = 1 & \text{otherwise} \end{array}$$

2.1 Pattern Storage

Binary patterns ξ_i^μ $(i = 1, \ldots, N; \mu = 1, \ldots, m)$ are embedded in the network through the tuning of connection weights w_{ij} using the following "learning rule":

$$w_{ij} = \frac{1}{N} \sum_\mu \xi_i^\mu \xi_j^\mu \qquad (3)$$

0-8186-8183-7/97 $10.00 © 1997 IEEE

This rule identifies the dynamical attractors of the network evolution equation (2) with the patterns stored in the network.

2.2 Pattern Recall

Recall is accomplished when the network converges to the original stored pattern from an initial state of imperfect information (corresponding to a noisy version of the stored image), evolving with time according to eqn. (2). If the initial state is denoted by

$$x_i(0) = \xi_i^\mu + \xi_i' \qquad (i = 1, \ldots, N)$$

where ξ_i' is the perturbation given to the original image, then the global stability of the Hopfield network implies

$$Lt_{t\to\infty}\ x_i(t) = \xi_i^\mu \qquad (i = 1, \ldots, N)$$

provided d_H is minimal for the pair $(x(0), \xi^\mu)$ and the loading fraction, $f < f_c$. Loading fraction is the ratio of the number of patterns stored, m, to the number of nodes, N. For the Hopfield network with stored orthogonal uncorrelated binary patterns, f_c=0.14 [4].

2.3 Gray-Level Decomposition

The gray levels of a s-bit gray scale image can be represented by a polynomial in powers of 2, viz., $\sum_{i=0}^{s-1} a_i 2^s$. Based on this property, a s-bit gray image can be decomposed into s number of bit plane images [5]. In the proposed method, the gray scale images to be stored in the network are, first of all, decomposed into the corresponding set of bit plane or binary images. Each of these binary images were then stored separately in networks.

The above decomposition method has an inherent disadvantage. Any correlation between neighboring pixels in the image is destroyed due to the decomposition process. To retain this information, a modification to the aforementioned technique can be introduced, whereby a s-bit gray code $g_{s-1} \ldots g_1 g_0$ is generated from the bit plane coefficients, a_i, as follows :

$$g_i = a_i \oplus a_{i+1},\ 0 \le i \le s-2 \text{ and } g_{s-1} = a_{s-1}$$

where, $\oplus$ is the XOR operation. This modification was also used as a possible refinement of the proposed method.

3 Results and Discussion

Computer simulation of the proposed method was carried out with 64 × 64, 8-bit gray level images. After storing sample images in the network through the learning process, their corrupted or occluded versions were used as input for recall. Fig. 1(a) and (b) shows two such input images (noisy and occluded, respectively) and the corresponding recovered image (Fig. 1(c)).

The noise was introduced with an uniform distribution in the unit interval [0,1]. For a given fraction of bits to be corrupted (F_{corr}, say) the prescription was as follows:

- Choose a random number n_j from the above distribution.
- If $n_j < (1 - F_{corr})$, then $x_j(0) = \xi_j^\mu$.
- If $n_j > (1 - F_{corr})$, then $x_j(0) = 0$ if $\xi_j^\mu = 1$ and vice versa.
- Repeat the above procedure for all $i = (1, \ldots, N)$

For Fig. 1(a), $F_{corr} = 0.1$.

The occlusion in the input image (Fig. 1(b)) was generated by setting all the bits of the bottom half of a stored pattern equal to zero. Some of the sample images stored in the network are shown in Fig. 2(a-c). In all cases, the network converges to the desired image within a few iterations. Investigation is now being carried out to ascertain the dependence of network performance (in terms of frequency of correct recall and recall time) on the loading fraction and the degree of corruption/occlusion of the input image.

Note that due to the bit-plane representation of the image, recall process of a single bit-plane does not depend on the other planes. This makes the proposed scheme suitable for parallel implementation. It is, therefore, superior to a Hopfield network with multivalued nodes (Potts spin model) for storing gray scale images in terms of time required for convergence. The proposed method also differs from pattern recognition networks [6] designed for gray images where the output is the pattern class instead of the image itself.

Acknowledgements

The authors would like to thank Prof. Sankar K. Pal (MIU, ISI) for his constant encouragement.

References

[1] B. Muller and J. Reinhardt, *Neural Networks: An Introduction*, New York: Springer Verlag, 1990.

[2] R. P. Lippmann, "An introduction to computing with neural nets",*IEEE ASSP Mag.*, vol. 4, pp. 4-22, 1987.

[3] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", *Proc. Natl. Acad. Sci. USA*, vol. 79, pp. 2554-2558, 1982.

[4] D. J. Amit, H. Gutfreund and H. Sompolinsky, "Statistical mechanics of neural networks near saturation", *Annals of Physics*, vol. 173, pp. 30-67, 1987.

[5] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Reading, Mass.: Addison-Wesley, 1992.

[6] F. J. Pineda, "Dynamics and architecture in neural computation", *Journal of Complexity*, vol. 4, pp. 216-245, 1988.

Fig. 1(a)

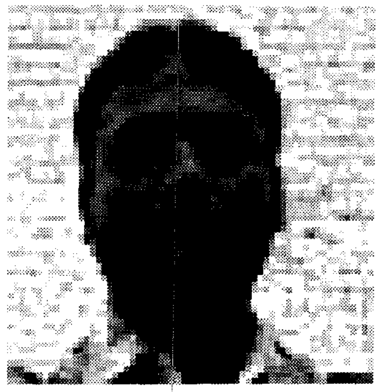

Fig. 2(a)

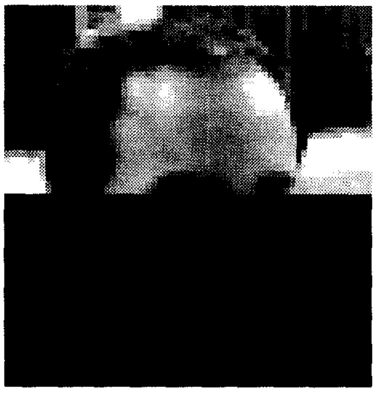

Fig. 1(b)

Fig. 2(b)

Fig. 1(c)

Fig. 2(c)