Proceedings of the 10th WSEAS International Conference on CIRCUITS, Vouliagmeni, Athens, Greece, July 10-12, 2006 (pp376-379)

Design of Associative Memory for Gray-Scale Images by Multilayer Hopfield Neural Networks

GIOVANNI COSTANTINI
Department of Electronic Engineering
University of Rome "Tor Vergata"
Via del Politecnico 1, 00133 - Rome
ITALY

Abstract: - In this paper a new design procedure for Hopfield associative memories storing grey-scale images is presented. The proposed architecture, with both intra-layer and inter-layer connections, is an evolution of a previous work based on the decomposition of the image with 2L gray levels into L binary patterns, stored in L uncoupled neural networks: that allows to store images with the commonly used number of 256 gray levels. The learning algorithm, used to store the binary images, guarantees asymptotic stability of the stored patterns, has a low computational cost, and allows to control the precision of the connection weights.

Key-Words: - Hopfield neural networks, associative memories, grey-scale images, finite precision weights.

1 Introduction

Reliable binary associative memories can be designed using neural networks with two-state activation function, or networks with saturation nonlinearities whose asymptotically stable states are binary valued [1,2]. The design of neural associative memories storing gray-scale images is a challenging problem investigated by few authors. The main problem to be faced is the robust recall of a stored image which can differ in any pixel from the noisy version. An image with n pixels and 2^L gray levels can be represented using L bits for each pixel. It can be stored using a binary neural network with nL neurons. This approach retains the main advantages of binary neural networks. However, it is not practical for either hardware implementation or software simulation, because of the quadratic growth with both the number of pixels and the number of bits (n^2L^2).

Some methods have been proposed to reduce the computational complexity. A first approach is based on neural networks with *multilevel threshold neurons*. The activation function is a quantization nonlinearity with 2^L *plateaus* in place of two, as in the usual sigmoidal function. The resulting stable equilibrium points have multivalued components, corresponding to the gray levels. The number of neurons is n and the number of interconnections is n^2. Some design methods have been proposed for networks with this type of nonlinearity, with interesting experimental results [3,4].

A second approach is based on *complex-valued neural networks* [5-7]. The neuron state can assume one of 2^L complex values, equally spaced on the unit circle. Each phase angle corresponds to a gray level. The number of neurons is n; the number of interconnections is n^2. For complex-valued neural networks a generalized Hebb rule was proposed in [5,6]. As shown in [6], this generalized rule has the same limitations on storage capacity of the usual Hebbian learning for binary-valued networks. Recently, a sophisticated method for weigth matrix design has been proposed in [7], based on the solution of a set of linear inequalities.

A third approach consists in the decomposition of the image into L binary patterns. Each pattern represents one bit in a suitable digital coding of the gray levels, and it is indipendently stored using a conventional *binary* neural network with n neurons [8]. There are L uncoupled sub-networks, each with n^2 interconnections. The main advantage is that the L sub-networks can be implemented via parallel hardware, with considerable saving in time, both for learning and recall. Moreover, each sub-network can be designed using one of the several available synthesis methods. However this approach presents two drawbacks. First, if a binary pattern cannot be stored in a sub-network, the corresponding image cannot be stored at all. In quantitative terms, the storage probability of a random set of images is the product of the storage probabilities in each sub-network. Hence, the storage capacity is quite lower than that of each sub-network with n neurons.

In the same way, the recall probability of an image, starting from a noisy version, equals the product of the recall probabilities in each sub-network. Therefore, the noise removal capability is modest.

The same problems arise if a large scale image is decomposed into sub-images stored in indipendent neural networks as explained in [9], where some overlapping between sub-images is suggested to alleviate these effects.
As the number of gray levels increases, both problems become worse since the number of indipendent networks increases. As a consequence, the method suggested in [8] is applicable only with resolution up to 16 gray levels. To overcome this limitation, here we present an evolution of our previous approach based on the introduction of connections between layers. Building inter-layer connections introduces interactions among all the neurons, even if not direct; due to the presence of these interactions both the capacity and the recall performance are improved with respect to the uncoupled case.

2 The Multilayer Hopfield Net Architecture

To implement the neural network architecture, we used a multilayer Hopfield neural net.
The Hopfield net [1] has only one layer of unit, that play a triple role as input, output, and processing unit. The units are globally interconnected and the state equation of a 2-dimensional network is the following:

$$\frac{dU_{ij}}{dt} = -U_{ij} + \sum_{kl} W_{ij,kl} V_{kl} + I_{ij}$$

$$i,k = 0, 1, 2 \ldots.., M \qquad j,l = 0, 1, 2 \ldots.., N$$

where $M \times N$ is the number of units, U_{ij} is the input of the unit at row i and column j, V_{kl} the output of the unit at row i and column j, $\mathbf{W}=[W_{ij,kl}]$ is the connection matrix, and I_{ij} the input bias to the unit at row i and column j. The output V_{ij} of is given as $V_{ij}=f(U_{ij})$, where f(.) is the piecewise-linear saturation nonlinearity:

$$f(x) = 0.5 \left(|x+1| - |x-1|\right).$$

In this work, we considered $I_{ij} = 0$ for every (i,j) and we assume a multilayer Hopfield net defined on a 3-dimensional ($M \times N \times L$) grid of units, where $C(i,j,p)$ denotes the unit at the intersection of row i and column j in layer p. The net connection is local, i.e. each unit in layer p is only connected to its neighborhood, defined by

$$N_{r,s}(i,j,p) = \{ (k, \ell, q) : |k-i| \le r \vee |k-i| \ge M-r,\ |\ell-j| \le r \vee |\ell-j| \ge N-r,\ |q-p| \le s \vee |q-p| \ge L-s \}$$

Each unit is therefore connected to $(2r+1)^2$ units in each of the layers p-s, ..., p-1, p, p+1, ..., p+s. We assume $r \ge s$.
To simplify the statements and the notation, *wraparound* connections will be assumed. This is equivalent to consider the net arranged on a hyper-torus. The number of connections per unit is:

$$\mu = (2s+1)(2r+1)^2.$$

The state equation of the multilayer Hopfield net is the following:

$$\frac{dU_{ijp}}{dt} = -U_{ijp} + \sum_{C(k,\ell,q) \in N_{r,s}(i,j,p)} W_{ijp,klp} V_{klp} + I_{ijp}$$

where U_{ijp} is the state of unit $C(i,j,p)$. $W_{ijp,k\ell q}$ denotes the connection weight from unit $C(k,\ell,q)$ to unit $C(i,j,p)$.
The image with n pixels and 2^L gray levels is decomposed into L binary patterns with n pixels. Each pattern corresponds to a *layer* of the multilayer net. Hence, the proposed architecture consists of L layers, each with n units; the total number of units is nL and the total number of interconnections is $\mu n L$. The number of interconnections grows only linearly with the number of pixels and logarithmically with the number of gray levels.
The decomposition of the image into binary patterns can be made using different coding strategies. The usual binary-weighted coding entails an amplification of the effect of additive Gaussian noise with zero mean. This common type of noise gives a high probability of “jumping” from a quantization level to an adjacent one, if the standard deviation σ is low. For example, if σ is equal to the quantization interval the probability that a gray level is transformed into an adjacent one is almost 50% (0.4834). This level jump results in a moderate degradation of the image from a perceptive viewpoint; however it could correspond to the reversing of several bits. For example, moving from gray-level 3 to gray-level 4, all the bits change ($011 \rightarrow 100$), hence there is a bit inversion in each layer of the proposed network. If there are too many binary errors, they cannot be corrected by the associative memory due to the excessive Hamming distance from a stored pattern.
To alleviate this problem we used the reflected-binary or Gray code which has the property that, moving from a quantization level to an adjacent one, only one bit changes. Now, moving from gray-level 3 to gray-level 4, only one bit changes ($010 \rightarrow 110$). Using the Gray code, additive zero-mean Gaussian noise results in a reduced Hamming distance between the stored binary pattern and that corresponding to the noisy image. As a

consequence, the recall of a stored image is simplified.

3 Design of associative memories with finite precision

To design the multilayer Hopfield net, we adapted the method proposed in [13], summarized in the following.

Let $\underline{N}_{r,s}(i,j,p) = N_{r,s}(i,j,p) - \mathrm{C}(i,j,p)$, and $y_{ijp} = f(x_{ijp})$. Assume $W_{ijp,ijp} = 1$ for every i, j and p. The design of the associative memory is as follows. First, we decompose the i-th Gray coded image to be stored, into L binary patterns from which we construct the i-th bipolar pattern $\mathbf{y}^{(i)} \in \{-1,+1\}^{M\times N\times L}$. Let $\mathbf{y}^{(1)} \ldots \mathbf{y}^{(Q)}$ be the Q bipolar patterns corresponding to the images to be stored. Then, we find the connection weights $W_{ijp,k\ell q}$ (i, k =1,…, M; j, ℓ =1,…, N; p, q =1,…, L, $k, \ell, q \neq i, j, p$), satisfying the following set of constraints:

$$\sum_{\mathrm{C}(k,\ell,q)\in \underline{N}_r(i,j,p)} W_{ijp,klq} V_{ijp}^{(m)} V_{klq}^{(m)} \geq \delta > 0 \qquad (1)$$

$i = 1,\ldots, M, j = 1,\ldots, N, p = 1,\ldots, L, m = 1,\ldots, Q.$

where δ represents a margin of stability for the stored patterns.

To compute the weights we use the following algorithm.

Let

$W_{ijp,k\ell q}(0) = 0$ for every $(i, j, p) \neq (k, \ell, q)$.

For every $t > 0$ compute:

$$\mathrm{P}(\Delta_{ijp}^{(m)}(t))$$

$i = 1,\ldots, M, j = 1,\ldots, N, p = 1,\ldots, L, m = 1,\ldots, Q,$

where

$$\Delta_{ijp}^{(m)}(t) = \sum_{\mathrm{C}(k,\ell,q)\in \underline{N}_r(i,j,p)} W_{ijp,klq} V_{ijp}^{(m)} V_{klq}^{(m)} - \delta \quad (2)$$

and

$$\mathrm{P}(x) = 0, \text{ for } x \geq 0, \ \mathrm{P}(x) = 1, \text{ for } x < 0.$$

Then, update the weights as follows:

$$W_{ijp,klq}(t+1) = W_{ijp,klq}(t) + \eta \sum_{m=1}^{Q} V_{ijp}^{(m)} V_{klq}^{(m)} \mathrm{P}(\Delta_{ijp}^{(m)}(t)) \qquad (3)$$

$$\mathrm{C}(k,\ell,q) \in \underline{N}_r(i,j,p).$$

P(x) is a penalty function of the constraint violation. $\eta > 0$ is a learning rate. A digital implementation of the algorithm is suggested in [13]. The **W** matrix so computed is not symmetric in general. To ensure the symmetry of **W** matrix, we consider the following updating rule, instead of eq. (3):

$$W_{ijp,klq}(t+1) = W_{ijp,klq}(t) + \eta \sum_{m=1}^{M} V_{ijp}^{(m)} V_{klq}^{(m)} P(\Delta_{ijp}^{(m)}(t), \Delta_{klq}^{(m)}(t))$$

where $\mathrm{P}(x,y)=1$ if $x<0$ or $y<0$, $\mathrm{P}(x,y)=0$ if $x\geq 0$ and $y\geq 0$.

Asymptotic convergence of this type of algorithm to a solution of (1) is not guaranteed, since it can approach a limit cycle in the solution space. However, by choosing η sufficiently small, it is possible to force the sequence $W_{ijp,klq}(t)$, $W_{ijp,klq}(t+1)$, ……. to stay arbitrarily close to a correct solution of (1). The presence of a margin δ in (2) guarantees the storage of the desired patterns.

Let examine some properties of the proposed algorithm:

1. Only additions are required for its implementation.
2. The weights can be represented as $W_{ijp,k\ell q} = \pm \eta N_{ijp,k\ell q}$, where $N_{ijp,k\ell q}$ is a positive integer. Hence, all the weights (at each iteration) have finite precision; the required number of bits is $\log_2(N_{max})+1$, where N_{max} is the maximum value of $N_{ijp,k\ell q}$.
3. The algorithm can be implemented or simulated on a digital hardware, without numerical errors, provided that a sufficient number of bits is used. In fact, no rounding or truncation is required to represent the weights.
4. The **W** matrix is symmetric, so the complete stability of the system is guaranteed.

4 Experimental results

A design example is presented to show the effectiveness of the proposed method, and we assume, for this example, $\eta = 0.1$.

The design objective is to store two real-life images with 200×200 pixels and 256 gray levels. The images are *lenna* and *stefan,* shown in Fig. 1. Due to computer memory limitations we partition each image into 16 parts, each with 50×50 pixels. This way we obtain 32 images which can be stored using a multilayer Hopfield net with 50×50×8=20,000 units and $r = s = 3$. Each part is stored as an indipendent stable equilibrium point of the net. The used value for δ is 500. Then, we recall the 32 stored images starting from a corrupted version. Noisy versions of the stored images are generated by adding zero mean gaussian noise, with standard deviation $\sigma = 16$. Noisy images are partitioned into 16 parts, which are used as initial states of the net. With $\delta = 500$ and $\sigma = 16$, both the images were correctly recalled without errors.

(a)

(b)

Fig. 1 – Images stored in the example: "Lenna" image (a), "Stefan" image (b).

We repeated this experiment using the uncoupled neural network proposed in [11], with the same images Lenna and Stefan shown in Fig. 4 and σ = 16. The best recall results, obtained with δ = 800, are shown in Fig. 2.

Fig. 2 – Recall results of the uncoupled neural network.

References:

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

[2] M. H. Hassoun (ed.), Associative Neural Memories: Theory and Implementation, Oxford Univ. Press, Oxford, 1993

[3] S. Mertens, H.M. Koehler, S. Bos, "Learning grey-toned patterns in neural networks", J. Phys. A, Math. Gen., vol. 24, pp. 4941-4952, 1991

[4] J. Si, A.N. Michel, "Analysis and syntehsis of a class of discrete-time neural networks with multilevel threshold neurons", IEEE Trans. Neural Networks, vol. 6, no. 5, pp. 105-116, January 1995

[5] N.N. Aizenberg, I. N. Aizenberg, "CNN based on multi-valued neuron as a model of associative memory for grey-scale images", Proc. of CNNA 92, Munich, Germany, pp. 36-41, 1992

[6] S. Jankowski, A. Lozowski, J.M. Zurada, "Complex-valued multistate neural associative memory", IEEE Trans. Neural Networks, vol. 7, no. 6, pp. 1491-1496, 1996

[7] M.K. Muezzinoglu, C. Guzelis, M. Zurada, "A new design method for the complex-valued multistate Hopfield associative memory", IEEE Trans. Neural Networks, vol. 14, no. 4, pp. 891-899, July 2003

[8] G. Costantini, D. Casali, R. Perfetti, "Neural associative memory storing Gray-coded gray-scale images", IEEE Trans. Neural Networks, vol. 14, no. 3, pp. 703-707, May 2003

[9] C. Oh, S.H. Zak, "Associative memory design using overlapping decompositions and generalized Brain-State-in a-Box neural networks", Int. J. Neural Syst., vol. 13, no. 3, pp. 139-153, 2003

[10] R. Perfetti, G. Costantini, "Multiplierless digital learning algorithm for cellular neural networks", IEEE Trans. Circuits and Systems – Part I, vol. CAS-48, No. 5, pp. 630-635, 2001