Proceedings of 1993 International Joint Conference on Neural Networks

ON AN OPTIMAL LEARNING SCHEME FOR BIDIRECTIONAL ASSOCIATIVE MEMORIES

K.Shanmukh and Y.V.Venkatesh

Computer Vision and Artificial Intelligence Laboratory
Department of Electrical Engineering
Indian Institute of Science
Bangalore 560012, India

Abstract

An optimal learning scheme is proposed for a class of Bidirectional Associative Memories (BAM's). This scheme, based on the Perceptron Learning Algorithm, is motivated by the inadequacies / incompleteness of the weighted learning by global optimization, as derived by Wang et al [1]. It is shown that the new scheme has superior properties : (1) Convergence to the correct solution, when it exists ; and (2) A larger basin of attraction for the given set of patterns.

1 Introduction

How to achieve versatile processing power by inter-connecting a large number of neurons (which are widely acknowledged to be simple and primitive) to create the so-called Artificial Neural Networks (ANN's) has assumed considerable significance in recent years. The inputs to the neurons consist of weighted sums of neuron outputs. And, as is well known, the input / output characteristic of a neuron can be modelled by a sigmoidal function.

In practice, the ANN's range from feed-through systems with no feedback, e.g., Perceptrons, to systems endowed with local feedback interconnections (e.g., cellular networks), to fully interconnected feedback systems. For instance, the Hopfield network exemplifies the last class.

As processing units for information analysis, neural networks are quite interesting, because, dynamically, they exhibit, at times, some stable states which act as basins of attraction. That is, when the neural network, starting from an arbitrary point in state-space, is allowed to evolve in time, it reaches a stable equilibrium point which, in effect, 'attracts' the neighboring states. Therefore, while referring to the behaviour of a neural network as a Pattern Analyzer, the convergence of its dynamics toward an equilibrium point is interpreted as the recognition (or labelling) of an imperfect pattern in terms of the correct (or stored) pattern. Conceptually, this is very similar (see Kohonen [2]) to the storage of information in an Associative Memory (AM).

2 Synthesis of an AM

Literature abounds in the study of neural networks to implement AM's. As indicated above, an AM can 'recall' the correct pattern when fed with either a part or corrupted version of it. In particular, the Hopfield neural network models have been shown [3] to implement AM's to some extent.

If these AM's are to be useful in practice as Pattern Recognizers, we need to design them in such a way that the memory patterns to be stored are its stable equilibrium points. Implementation of this requirement leads to designing an AM in which the patterns are input as vectors, which, in turn, are transformed, by an appropriate Learning Strategy, to interconnection strengths in the AM. After the Learning Stage, when the AM is used to recognize patterns, the input pattern vector initiates the dynamics of the AM which evolves, in time, in such a way that an energy function of the AM assumes a minimum. If the input pattern is similar to a pattern stored (as interconnection strengths, during the learning phase) in the AM, the dynamical evolution will result in convergence to the nearest stored pattern. This is because the AM design implies that all of its trajectories seek the local minima of the energy function.

Mathematically, if the vector $\mathbf{x}$ represents a given pattern stored in an AM, then if the input vector is $\mathbf{x}+\delta\mathbf{x}$, the AM will dynamically evolve to $\mathbf{x}$, provided that an appropriate norm of $\delta\mathbf{x}$ (say $\mid \delta x \mid$, $\parallel \delta x \parallel$) is sufficiently small.

2.1 Bidirectional Associative Memory

BAM is a two layer feedback neural network, which can be used to store associations between patterns. See Fig.1. The neurons in one layer are connected to the neurons in the other layer, but there are no connections within a layer. Let N and P be the number of neurons in the first and second layers respectively. Let W_{ij} be the connection weight between neuron i in the first layer and neuron j in the second layer. Let $\{(X_k, Y_k)\}$, $k = 1, ..., M$ be the set of pattern

pairs to be stored, where X_k's and Y_k's are N- and P-dimensional vectors, respectively. In the standard recall procedure, each neuron state $X(i)$ and $Y(j)$ is updated as follows:

$$X(i)' = \phi[\Sigma_{j=1}^{P} W_{ij} Y(j)] \qquad i = 1, ..., N$$
$$Y(j)' = \phi[\Sigma_{i=1}^{N} W_{ij} X(i)] \qquad j = 1, ..., P \quad (1)$$

where $X(i)'$ and $Y(j)'$ are the new states of the i^{th} unit in the first layer and j^{th} unit in the second layer respectively. The function $\phi(x)$ is +1 for $x > 0$ and -1 for $x < 0$.

In the learning phase, a weight matrix to store the given pattern pairs is obtained. Two aspects are to be considered in learning:

1. The desired patterns are to be stored as stable states.
2. The stored patterns are to be made *optimally* stable.

From (1), we see that a pattern pair (X, Y) is a stable state, if and only if the following equations hold.

$$X(i)\ \phi[\Sigma_{j=1}^{P} W_{ij} Y(j)] > 0 \qquad i = 1, ..., N$$
$$Y(j)\ \phi[\Sigma_{i=1}^{N} W_{ij} X(i)] > 0 \qquad j = 1, ..., P \quad (2)$$

The first aspect of the learning phase is to obtain the weight matrix satisfying the above equations for all pattern pairs to be stored. The second aspect requires a W which gives wider basins of attraction around each stored pattern pair. In view of the difficulty of solving this problem, we consider a simpler version of this, namely, finding a W which satisfies the following equations :

$$X_k(i)\ \phi[\Sigma_{j=1}^{P} W_{ij} Y_k(j)] > \Delta,$$
$$Y_k(j)\ \phi[\Sigma_{i=1}^{N} W_{ij} X_k(i)] > \Delta, \quad (3)$$

where $i = 1, ..., N; j = 1, ..., P; k = 1, ...M$; and Δ is a positive number. An optimal learning algorithm should maximise the Δ in the above equations with respect to a normalized weight matrix.

2.2 Learning in a BAM

Kosko[4] has proposed a correlation learning rule which obtains the connection weights using the formula,

$$W_{ij} = \Sigma_{k=1}^{M} X_k(i)\ Y_k(j) \quad (4)$$

When the patterns are not orthogonal, this rule does not work well. Even when there exists a matrix W such that the given patterns are stable, this learning rule may fail in storing the patterns.

To overcome this problem, and to obtain an optimal weight matrix, a modified learning algorithm based on multiple training has been proposed in [1]. In this method, the connection weight is of the form :

$$W_{ij} = \Sigma_{k=1}^{M} C_k\ X_k(i)\ Y_k(j) \quad (5)$$

The algorithm proposed in [1] obtains C_k's which make the patterns stable. It is proved therein that this algorithm converges whenever there exists a set of C_k's such that the given patterns are stable. A condition for the existence of C_k's is derived, and it is shown to be equivalent to the linear separability of a set of patterns which are constructed from the given pattern set.

However a drawback of this method is that it may not be able to store the patterns which can, theoretically, be stored. This problem arises because, the algorithm obtains the solution only when it is of the form given in (5). For some pattern sets which can be stored, there may not exist a set of C_k's such that W in (5) makes them stable.

For instance, consider the set of patterns in Fig.2. It can be shown, by using the Ho-Kashyap algorithm [6], that there *do not* exist C_k's which make the patterns stable, inspite of the fact that these patterns can be stored.

```
X1 = [ -1 -1 -1  1 -1 -1  1 ]
Y1 = [ -1 -1 -1 -1  1  1 -1 ] ;
X2 = [ -1 -1  1  1 -1  1 -1 ]
Y2 = [  1  1  1 -1  1 -1 -1 ] ;
X3 = [  1  1  1 -1  1 -1  1 ]
Y3 = [ -1  1  1 -1 -1  1 -1 ] ;
X4 = [ -1  1 -1 -1  1 -1  1 ]
Y4 = [  1 -1  1 -1 -1 -1 -1 ]
```

Fig.2

3 Main Results

In the present paper, we give an algorithm which takes care of both the aspects of learning. First, we show the equivalence of the BAM to a Perceptron, and then translate the optimal learning algorithms of Perceptrons into the structure of the BAM. To this end, consider the BAM of Fig.3 with $N = P = 3$.

Let (X_k, Y_k) be one pair of patterns to be stored. The conditions on W_{ij}'s for (X_k, Y_k) to be stable are given in (2).

Now consider the Perceptron of Fig.4. The conditions (2) on W_{ij}'s are equivalent to the following pattern set for this Perceptron with the corresponding outputs.

$$\begin{pmatrix} Y_k(1) \\ Y_k(2) \\ Y_k(3) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} -> X_k(1); \quad \begin{pmatrix} 0 \\ 0 \\ 0 \\ Y_k(1) \\ Y_k(2) \\ Y_k(3) \\ 0 \\ 0 \\ 0 \end{pmatrix} -> X_k(2)....$$

Each pattern pair in the BAM structure gives rise to $(N+P)$ sub-patterns in the eqivalent perceptron, corresponding to the $(N + P)$ neurons

in the BAM. Now, storing M pattern pairs in BAM is equivalent to learning in a Perceptron with these $M * (N + P)$ sub-patterns.

3.1 Optimal Learning

As BAM is equivalent to a Perceptron, we can apply the optimal learning algorithms of the Perceptron to BAM. One such algorithm is described below. For further details, see [6].

Let $(Z_1, ..., Z_L)$ be the pattern set for a Perceptron, and let $(y_1, ..., y_L)$ be the set of corresponding outputs. Let $W = (w_1, ..., w_n)$ denote the weight vector.

Then, adopt the following learning rule : When the pattern Z_p is presented, the change in the weight w_i is given by

$$\delta w_i = \varepsilon^p \; Z_p(i) \; y_p \qquad (6)$$

where ε^p is defined as

$$\varepsilon^p = \theta\left[\Delta \left(\Sigma_{j=1}^n w_j^2\right)^{1/2} - \Sigma_{j=1}^n w_j Z_p(j) y_p\right] \qquad (7)$$

$\theta[t] = 1$ if $t > 0$ and $= 0$ otherwise ; and $\Delta >= 0$.

It can be shown [6] that this algorithm converges whenever there exists a W such that

$$\left(W^T \; Z_i\right) \; y_i > \Delta \left(\Sigma_{j=1}^n w_j^2\right)^{1/2} \qquad (8)$$

for all i. Using this algorithm, we can obtain the largest Δ for which (8) holds.

Here we note that (see [7]) even if the weights are modified only after presenting a set of patterns instead of a single pattern, the algorithm still converges. The convergence can be proved on the same lines as found in [6].

Now we use these results to obtain a learning algorithm for BAM. When a pattern is presented in the equivalent Perceptron, it is equivalent to (i) presenting a pattern to the BAM ; (ii) checking the output of the corresponding neuron ; and (iii) modifying the corresponding weights. If all the $(N + P)$ sub-patterns corresponding to a pattern pair (X_k, Y_k) are presented to the Perceptron, and the weights are modified after all these are presented, then the change in W_{ij} is given by

$$\delta W_{ij} = \left(\varepsilon_{1i}^k + \varepsilon_{2j}^k\right) X_k(i) \; Y_k(j) \qquad (9)$$

where

$$\varepsilon_{1i}^k = \theta[u]$$

$$\varepsilon_{2j}^k = \theta[v],$$

and

$$u = \Delta \left(\Sigma_{i=1}^N \Sigma_{j=1}^P W_{ij}^2\right)^{1/2} - \left(\Sigma_{j=1}^P W_{ij} Y_k(j)\right) X_k(i)$$

$$v = \Delta \left(\Sigma_{i=1}^N \Sigma_{j=1}^P W_{ij}^2\right)^{1/2} - \left(\Sigma_{i=1}^N W_{ij} X_k(i)\right) Y_k(j).$$

4 Experimental Results

We compare the proposed method with the weighted learning method of [1] using a set of test patterns on a simulated neural BAM, with N = P = 15, and M assuming values from 2 to 7. For each value of M, 100 experiments are performed. In each experiment, patterns are generated randomly, and the two algorithms are applied. We call an experiment successful if the algorithm is able to store all the given patterns. Results of the comparison are summarized in Table 1 and Table 2 : Table 1 gives the percentage of succesful experiments ; and Table 2 contains $(\Delta / \| W \|_i)$, averaged over all neurons and over all succesful experiments, where $\| W \|_i$ is the Euclidian norm of the vector of weights connecting neuron i to other neurons ; and Δ, as obtained by the algorithm, is the largest positive number that satisfies (3) . The superiority of the proposed method is evident from the results given.

5 Conclusions

For a class of Bidirectional Associative Memories (BAM's), an optimal learning scheme has been proposed, inspired by the Perceptron Learning Algorithm, and motivated by the inadequacies / incompleteness of the weighted learning by global optimization, as derived by Wang et al [1]. It is shown that the new scheme has superior convergence and stability properties.

6 References

1. T.Wang, X.Zhuang and X.Xing, Weighted learning of bidirectional associative memories by global minimization, IEEE Trans. on Neural Networks, Vol.3, No.6, pp 1010-1018, 1993.

2. T.Kohonen, Self-Organizing and Associative Memories, Springer-Verlag, New York, 1987

3. J.J.Hopfield, Neural Networks and Physical Systems with Emergent Collective Computational Abilities, Proc. Natl. Acad. Sci. Vol. 79 pp. 2554-2558, 1982.

4. B.Kosko, Adaptive bidirectional associative memories, Applied Optics, Vol.26, No.23, pp 4947-4960, 1987.

5. J.T.Tou and R.C.Gonzalez, Pattern Recognition Principles, Addison-Wesley, 1974.

6. E.Gardner, The space of interactions in neural network models, Journal of Physics A: Math. and General, Vol. 21, pp. 257-270, 1988.

7. K.Shanmukh and Y.V.Venkatesh, On an Optimal Learning Scheme for Bidirectional Associative Memories, Technical Report April 1993, Department of Electrical Engineering, Indian Institute of Science, Bangalore, India.

M	2	3	4	5	6	7
Weighted Learning	94	87	78	51	10	5
Proposed Method	100	100	99	100	100	98

Table 1

M	2	3	4	5	6	7
Weighted Learning	2.51	1.59	1.14	0.81	0.69	0.52
Proposed Method	2.72	1.88	1.5	1.26	1.13	1.01

Table 2

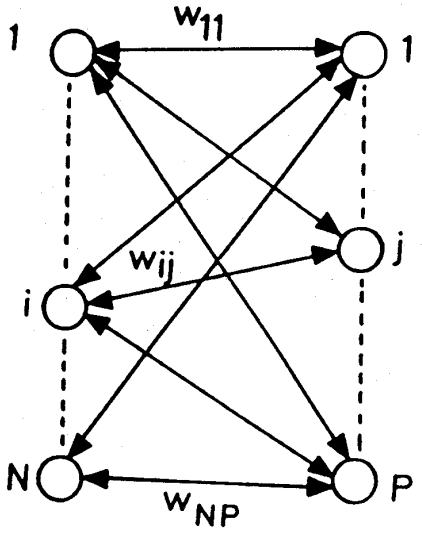

Fig. 1

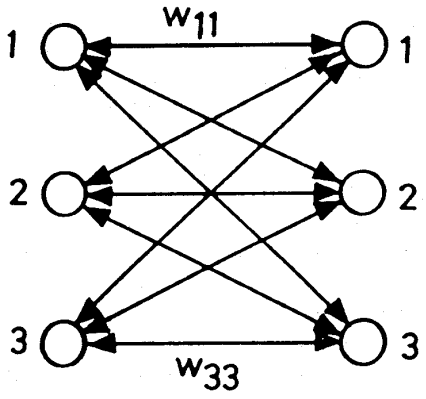

Fig. 3

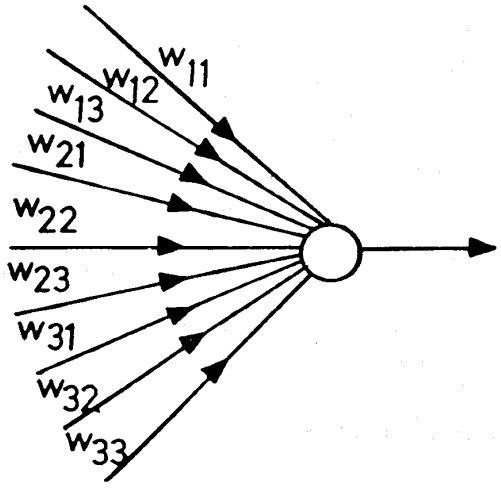

Fig.4