ELSEVIER

Neurocomputing 35 (2000) 137–148

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Synthesis approach for bidirectional associative memories based on the perceptron training algorithm

Ismail Salih[a], Stanley H. Smith[a], Derong Liu[b,*,1]

[a]Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030-5991, USA

[b]Department of Electrical Engineering and Computer Science, University of Illinois, 1120 Science and Engineering Offices, 851 South Morgan Street, Chicago, IL 60607-7053, USA

Received 6 June 1999; accepted 13 April 2000

Abstract

Bidirectional associative memories are being used extensively for solving a variety of problems related to pattern recognition. In the present paper, a new synthesis approach is developed for bidirectional associative memories using feedback neural networks. The synthesis problem of bidirectional associative memories is formulated as a set of linear inequalities which can be solved using the perceptron training algorithm. To demonstrate the applicability of the present results, a specific example is considered. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Associative memory; Bidirectional associative memory; Perceptron training algorithm

1. Introduction

One of the important features of artificial neural networks is the information storage and retrieval implemented as associative memories. Feedback neural networks are a special class of nonlinear dynamical systems which are endowed with many asymptotically stable equilibrium points (stable memories) as well as unstable equilibria. The study of such systems has been of great interest to many researchers in recent years (see, e.g., [1–15,17,18,20,23–27]).

* Corresponding author. Tel.: + 1-312-355-4475; fax: + 1-312-413-0024.
E-mail address: derong@eecs.uic.edu (D. Liu).

[1] Supported by the National Science Foundation under Grant ECS-9732785.

PII: S0925-2312(00)00313-1

A bidirectional associative memory (BAM) stores a set of pattern associations and each pattern is represented by a pair of vectors [4,5]. The architecture of the network consists of two layers of neurons, connected by directional weighted connection paths. The network iterates, sending signals back and forth between the two layers, until all neurons reach equilibrium states. BAM neural networks can respond to inputs to either one of the layers. We shall refer to the layers as the X-layer and the Y-layer. BAM neural networks behave as heteroassociative pattern matchers, storing and recalling pattern pairs, which also allow the retrieval of stored data associations from incomplete or noisy patterns. The design of a BAM neural network involves the computation of all the connection weights in the neural network, such that the network will have a given set of vector pairs as its memories.

Since Kosko's work on the design of BAM using correlation encoding algorithm [4,5], several authors have presented new design methods for BAM neural networks. One problem existing in the design method of [4,5] is that it does not guarantee to store all the desired patterns as equilibrium points of the network. In [25], a design algorithm based on an optimization technique is developed which guarantees to store all the desired patterns as stable equilibrium points of the network. In [28], a design algorithm based on the so-called best approximation projection is provided for the design of asymmetric BAMs; however, the design algorithm can only be applied to linearly independent desired memory patterns. In [29], the eigenstructure method is employed for the design of BAM neural networks; it does not have any constraints on the desired memory patterns and it guarantees to store all the desired patterns as stable equilibrium points of the network.

The model that we consider is described by the equations

$$\dot{x} = -Ax + U\,\mathrm{sat}(y) + I,$$
$$\dot{y} = -By + V\,\mathrm{sat}(x) + J, \tag{1}$$

where $x \in \Re^m$ and $y \in \Re^n$ are the state vectors, $\dot{x}$ and $\dot{y}$ denote the derivatives of x and y with respect to time t, $A = \mathrm{diag}[a_1, \ldots, a_m]$ with $a_i > 0$ for $i = 1, \ldots, m$, $B = \mathrm{diag}[b_1, \ldots, b_n]$ with $b_i > 0$ for $i = 1, \ldots, n$, $U = [U_{ij}] \in \Re^{m \times n}$ is the coefficient matrix in the X-layer, $V = [V_{ij}] \in \Re^{n \times m}$ is the coefficient matrix in the Y-layer, $I = [I_1, \ldots, I_m]^{\mathrm{T}} \in \Re^m$, and $J = [J_1, \ldots, J_n]^{\mathrm{T}} \in \Re^n$. I and J are the bias vectors for the X-layer and the Y-layer, respectively. The function $\mathrm{sat}(x) = [\mathrm{sat}(x_1), \ldots, \mathrm{sat}(x_m)]^{\mathrm{T}}$ represents the activation function, where

$$\mathrm{sat}(x_i) = \begin{cases} 1, & x_i > 1, \\ x_i, & -1 \le x_i \le 1, \\ -1, & x_i < -1. \end{cases}$$

The outputs for fields X and Y are $\mathrm{sat}(x)$ and $\mathrm{sat}(y)$, respectively.

This paper makes contributions to feedback neural networks for bidirectional associative memories. In particular, a new synthesis approach for BAMs using feedback neural networks will be developed based on the perceptron training algorithm. The design (synthesis) problem of neural networks (1) for bidirectional associative memories is developed by formulating and solving a set of linear *inequalities*. The inequalities are solved by training a set of perceptrons to obtain connection matrices, and the perceptron training is guaranteed to converge if a solution of the design problem exists. Similar technique has been employed in [10] where the design problem of neural networks for associative memories is considered. In [10], the design of neural networks for associative memories based on the perceptron training algorithm is compared to the eigenstructure method [9] and the optimization method [21] through extensive simulation. The simulation in [10] concluded that the perceptron training approach in general has less spurious memories than the eigenstructure method; the perceptron training approach is also simpler to implement and takes less computational time than the optimization method.

Preliminaries will be first introduced in Section 2 concerning the perceptron training algorithm and its convergence theorem. In Section 3, a new synthesis algorithm for bidirectional associative memories will be developed based on the perceptron training algorithm. In Section 4, a specific example will be considered to demonstrate the applicability of the present results. In Section 5, the paper will be concluded with several pertinent remarks.

2. Preliminaries

This section introduces necessary preliminaries including the perceptron training algorithm and its convergence theorem.

A number of different types of perceptrons are described in [16,19]. The one which will be utilized in the present paper is described by

$$z = \mathrm{sgn}(Wu),$$

where $u = [u_1, u_2, \ldots, u_n, 1]^{\mathrm{T}}$, $W = [w_1, w_2, \ldots, w_n, \theta]$, and

$$\mathrm{sgn}(\xi) = \begin{cases} 1, & \xi \geq 0, \\ -1, & \xi < 0. \end{cases}$$

This simple perceptron can perform pattern classification (between two classes denoted by C_1 and C_2). The weight vector W can be obtained by the following perceptron training algorithm (cf. [16,19]).

Perceptron Training Algorithm: Given p training patterns α^k, $k = 1, 2, \ldots, p$, which are known to belong to class C_1 (corresponding to $z = 1$) or C_2 (corresponding to $z = -1$), the weight vector W can be obtained by the following algorithm:

(1) Initialize the weight vector $W(l)$ for $l = 0$.
(2) For $l = 0, 1, 2, \ldots,$
  - if $W(l)u(l) \geq 0$ and $u(l) \in C_2$, then update $W(l+1) = W(l) - \eta u(l)$,
  - if $W(l)u(l) < 0$ and $u(l) \in C_1$, then update $W(l+1) = W(l) + \eta u(l)$,
  - otherwise, $W(l+1) = W(l)$,

  where $u(l) = \alpha^k$ for some k, $1 \leq k \leq p$, and $\eta > 0$ is the perceptron learning rate.

(3) Stop the training when no more updates for the weight vector W are needed; i.e., stop the training when all the training patterns can be correctly classified by W.

The following result is well-known [16,19].

Perceptron Training Convergence Theorem. *The Perceptron Training Algorithm will be convergent if and only if the two classes C_1 and C_2 are linearly separable.*

Remark 1. It is noted that one can always continue the training of a perceptron until a weight vector W is obtained such that

$$W\alpha^k \begin{cases} > 0 & \text{if } \alpha^k \in C_1, \\ < 0 & \text{if } \alpha^k \in C_2 \end{cases}$$

for $k = 1, 2, \ldots, p$.

In the sequel, the Perceptron Training Algorithm will be used to develop a new synthesis approach for bidirectional associative memories realized by neural networks (1).

3. Synthesis algorithm

We use B^n to denote the set of n-dimensional *bipolar vectors*, i.e, $B^n \triangleq \{x \in \Re^n \colon x_i = 1 \text{ or } -1,\ i = 1, \ldots, n\}$. For $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]^{\mathrm{T}} \in B^n$, we define $E(\alpha) = \{x \in \Re^n \colon \alpha_i x_i > 1,\ i = 1, \ldots, n\}$. The pair (α, β), with $\alpha \in B^m$ and $\beta \in B^n$, will be called a memory vector if $\alpha = \mathrm{sat}(x_e)$ and $\beta = \mathrm{sat}(y_e)$, where (x_e, y_e) is an asymptotically stable equilibrium point of the system (1).

The following result is obtained for bipolar memory vectors.

Lemma 2. *If $\alpha \in B^m$ and $\beta \in B^n$ with*

$$A^{-1}(U\beta + I) \in E(\alpha),$$

$$B^{-1}(V\alpha + J) \in E(\beta) \tag{2}$$

then (α, β) is a memory vector of system (1).

Proof. Assume that $x \in E(\alpha)$ and $y \in E(\beta)$, where $\alpha \in B^m$ and $\beta \in B^n$. System (1) becomes

$$\dot{x} = -Ax + U\beta + I,$$

$$\dot{y} = -By + V\alpha + J \tag{3}$$

since $\mathrm{sat}(x) = \alpha$ and $\mathrm{sat}(y) = \beta$. System (3) has a unique equilibrium point at

$$x_e = A^{-1}(U\beta + I),$$

$$y_e = B^{-1}(V\alpha + J).$$

Clearly, the equilibrium point (x_e, y_e) is asymptotically stable since the eigenvalues of the system matrices $-A$ and $-B$ in (3) are all negative and $x_e \in E(\alpha)$ and $y_e \in E(\beta)$ by assumption. □

The following synthesis problem concerns the design of neural networks (1) for BAM.

Synthesis problem. Given p pairs of vectors in B^{m+n}, say $(\alpha^1, \beta^1), \ldots, (\alpha^p, \beta^p)$, we wish to design a system (1) such that $(\alpha^1, \beta^1), \ldots, (\alpha^p, \beta^p)$ are memory vectors of the system.

To solve the synthesis problem, one needs to determine A, B, U, V, I and J from Eq. (2) with $\alpha = \alpha^k$ and $\beta = \beta^k$ for $k = 1, 2, \ldots, p$, i.e., one needs to determine A, B, U, V, I and J such that

$$A^{-1}(U\beta^k + I) \in E(\alpha^k),$$

$$B^{-1}(V\alpha^k + J) \in E(\beta^k)$$

for $k = 1, 2, \ldots, p$. This condition can be equivalently written as, for $k = 1, 2, \ldots, p$

$$U_i\beta^k + I_i \begin{cases} > a_i & \text{if } \alpha_i^k = 1, \\ < -a_i & \text{if } \alpha_i^k = -1 \end{cases} \quad i = 1, 2, \ldots, m \tag{4}$$

and

$$V_j\alpha^k + J_j \begin{cases} > b_j & \text{if } \beta_j^k = 1, \\ < -b_j & \text{if } \beta_j^k = -1 \end{cases} \quad j = 1, 2, \ldots, n, \tag{5}$$

where U_i (V_j) represents the ith (jth) row of U (V), I_i (J_j) denotes the ith (jth) element of I (J), and α_i^k (β_j^k) is the ith (jth) entry of α^k (β^k).

Remark 3. From Eqs. (4) and (5), one can see that the design problem of BAM neural networks is essentially equivalent to designing $m + n$ threshold logic functions with each containing $m + n + 2$ variables. This observation motivates the choice of perceptron training algorithm over other techniques for solving the BAM neural network design problem in the present work and in [10] for the design of neural networks for associative memories.

From the preceding, the following synthesis algorithm (design method) based on the Perceptron Training Algorithm can now be obtained.

Synthesis algorithm 3.1. Using the Perceptron Training Algorithm, one obtains $m + n$ perceptrons

$$P^i = [p_1^i, p_2^i, \ldots, p_{n+1}^i], \quad i = 1, 2, \ldots, m$$

and

$$Q^j = [q_1^j, q_2^j, \ldots, q_{m+1}^j], \quad j = 1, 2, \ldots, n$$

such that

$$P^i\bar{\beta}^k \begin{cases} > 0 & \text{if } \alpha_i^k = 1, \\ < 0 & \text{if } \alpha_i^k = -1 \end{cases} \tag{6}$$

and

$$Q^j\bar{\alpha}^k \begin{cases} > 0 & \text{if } \beta_j^k = 1, \\ < 0 & \text{if } \beta_j^k = -1 \end{cases} \tag{7}$$

for $k = 1, 2, \ldots, p$, where

$$\bar{\alpha}^k = \begin{bmatrix} \alpha^k \\ \cdots \\ 1 \end{bmatrix} \quad \text{and} \quad \bar{\beta}^k = \begin{bmatrix} \beta^k \\ \cdots \\ 1 \end{bmatrix}.$$

Choose $A = \mathrm{diag}[a_1, \ldots, a_m]$ with

$$0 < a_i < \min_{1 \le i \le m, 1 \le k \le p} |P^i\bar{\beta}^k|$$

and choose $B = \mathrm{diag}[b_1, \ldots, b_n]$ with

$$0 < b_j < \min_{1 \le j \le n, 1 \le k \le p} |Q^j\bar{\alpha}^k|.$$

For $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, choose $U_{ij} = P_j^i$, $V_{ji} = Q_i^j$, $I_i = p_{n+1}^i$ and $J_j = q_{m+1}^j$. □

The next result addresses the validity of the above synthesis algorithm.

Theorem 4. (1) *The perceptron training* P^i *converges if and only if the two classes given by*

$$\begin{aligned} C_1 &= \{\bar{\beta}^k : \alpha_i^k = 1\}, \\ C_2 &= \{\bar{\beta}^k : \alpha_i^k = -1\} \end{aligned} \tag{8}$$

are linearly separable, where $1 \le i \le n$. *Conditions for the training of* Q^j *are similar.*

(2) *The* A, B, U, V, I *and* J *obtained in the Synthesis Algorithm* 3.1 *guarantee that* $(\alpha^1, \beta^1), \ldots, (\alpha^p, \beta^p)$ *are memory vectors of the system* (1).

Proof. (1) The proof is straightforward by considering (6) and (7).

(2) From the choice of a_i, $U_i = [U_{i1}, U_{i2}, \ldots, U_{in}]$ and I_i in the Synthesis Algorithm 3.1,

$$[U_i \;\; I_i]\,\bar{\beta}^k \begin{cases} = P^i\bar{\beta}^k > a_i & \text{if } \alpha_i^k = 1, \\ = P^i\bar{\beta}^k < -a_i & \text{if } \alpha_i^k = -1 \end{cases} \tag{9}$$

which implies

$$a_i^{-1}[U_i\beta^k + I_i]\begin{cases} > 1 & \text{if } \alpha_i^k = 1, \\ < -1 & \text{if } \alpha_i^k = -1 \end{cases} \tag{10}$$

or equivalently, $A^{-1}(U\beta^k + I) \in E(\alpha^k)$. The proof of $B^{-1}(V\alpha^k + J) \in E(\beta^k)$ is similar. Therefore, $(\alpha^1, \beta^1), \dots, (\alpha^p, \beta^p)$ are memory vectors of the system (1) according to Lemma 2. □

Corollary 5. *If the vectors* $\bar{\beta}^k$ $(\bar{\alpha}^k)$, $k = 1, \dots, p$, *are linearly independent, then the perceptron training* P^i (Q^j) *converges.*

Proof. We show that if $\bar{\beta}^k$, $k = 1, \dots, p$, are linearly independent, C_1 and C_2 given in Eq. (8) are linearly separable. Assume that $C_1 = \{\bar{\beta}^1, \dots, \bar{\beta}^q\}$ and $C_2 = \{\bar{\beta}^{q+1}, \dots, \bar{\beta}^p\}$, without loss of generality. Let $co\,C$ denote the convex hull of a set $C \subset R^n$. Assume that $co\,C_1 \cap co\,C_2 \neq \emptyset$. Then, there exists $\bar{\beta} \in \{co\,C_1 \cap co\,C_2\}$ which could be written as a convex combination of the elements of class C_1 on one hand, and on the other hand as a convex combination of the elements of class C_2; i.e, $\bar{\beta} = \lambda_1\bar{\beta}^1 + \cdots + \lambda_q\bar{\beta}^q = \lambda_{q+1}\bar{\beta}^{q+1} + \cdots + \lambda_p\bar{\beta}^p$, for some $\lambda_1, \dots, \lambda_q$, $\lambda_{q+1}, \dots, \lambda_p \geq 0$ such that $\lambda_1 + \lambda_2 + \cdots + \lambda_q = 1$ and $\lambda_{q+1} + \cdots + \lambda_p = 1$ (cf. [22]). Then, $\lambda_1\bar{\beta}^1 + \cdots + \lambda_q\bar{\beta}^q - \lambda_{q+1}\bar{\beta}^{q+1} - \cdots - \lambda_p\bar{\beta}^p = 0$ which contradicts the assumption that the vectors $\bar{\beta}^k$, $k = 1, \dots, p$, are linearly independent. This shows that $co\,C_1 \cap co\,C_2 = \emptyset$ which implies that C_1 and C_2 are linearly separable. □

Remark 6. The learning rate η in the perceptron training algorithm can be any positive real number [16,19]. If $\eta = 1$ or any other positive integer and if one chooses the initial P^i and Q^j to be the zero vector or any vector with integer and zero components, then, U, V, I, J will have integer components. It is noted that in VLSI implementations of neural networks, certain weights (e.g., integers) can be implemented *more* accurately than others (e.g, numbers with many decimal digits). The learning rate η specifies the step size of every update for the weight vector during the perceptron training. A large η gives large step size which implies a coarse searching in the solution space of P^i and Q^j. In most cases, choosing η to be $0 < \eta < 1$ is desirable for the perceptron training to converge quickly.

Remark 7. If one wishes that the above Synthesis Algorithm 3.1 results in a system of form (1) with $I = 0$ and $J = 0$, one can modify (6) and (7) in the Synthesis Algorithm as follows:

$$P^i\beta^k\begin{cases} > 0 & \text{if } \alpha_i^k = 1, \\ < 0 & \text{if } \alpha_i^k = -1 \end{cases} \tag{11}$$

and

$$Q^j\alpha^k\begin{cases} > 0 & \text{if } \beta_j^k = 1, \\ < 0 & \text{if } \beta_j^k = -1. \end{cases} \tag{12}$$

A, B, U, and V are chosen in the same way as in the Synthesis Algorithm 3.1.

4. An example

To demonstrate the applicability of the results in the present paper, a specific example will be considered.

The neural network considered is a two layer network each of which consists of 49 neurons ($m = 49$, $n = 49$) with the objective of storing the $p = 4$ pairs of patterns $(\alpha^1, \beta^1), \ldots, (\alpha^4, \beta^4)$ shown in Fig. 1 as memories. As indicated in this figure, 49 boxes are used to represent each pattern (in R^{49}), with each box corresponding to a vector component which is allowed to assume values between -1 and 1. For purpose of visualization, -1 will represent white, 1 will represent black, and the intermediate values will correspond to appropriate grey levels.

We use the Synthesis Algorithm 3.1 developed in Section 3 to design a bidirectional associative memory as in system (1). The performance of the designed network is illustrated by means of typical simulation results shown in Figs. 2–5. In Fig. 2, both initial patterns are generated by randomly reversing 18–20 bits of each original patterns. In Fig. 3, the initial pattern of field X is chosen as the zero vector and the initial pattern of field Y is generated by randomly reversing 13 bits of the desired pattern. In Fig. 4, the initial pattern of field X is chosen as the zero vector and the initial pattern of field Y is chosen as the desired pattern ("c" recall "C"). In Fig. 5, the initial pattern of field X is chosen as the desired pattern and the initial pattern of

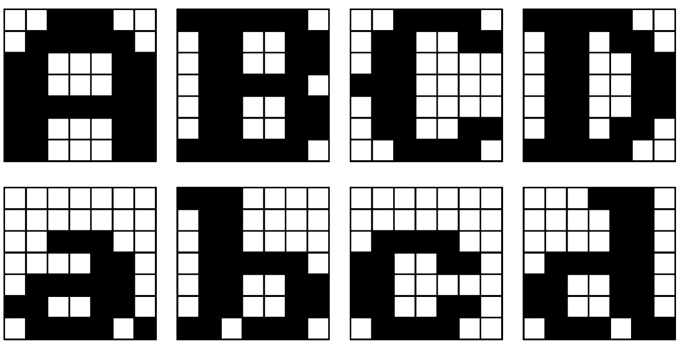


Fig. 1. The four desired pairs of memory patterns.

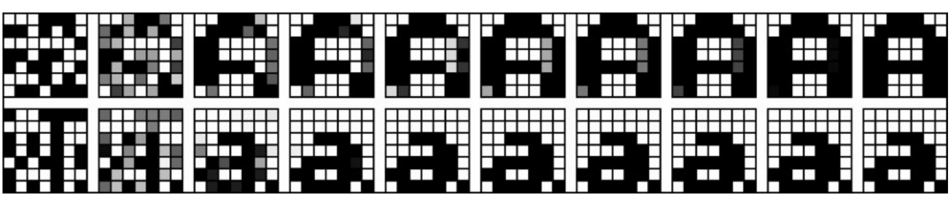


Fig. 2. The simulation result for the first pair.

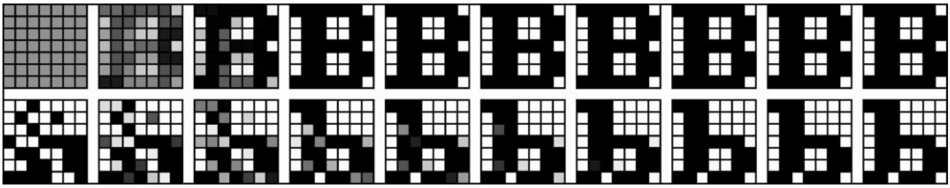


Fig. 3. The simulation result for the second pair.

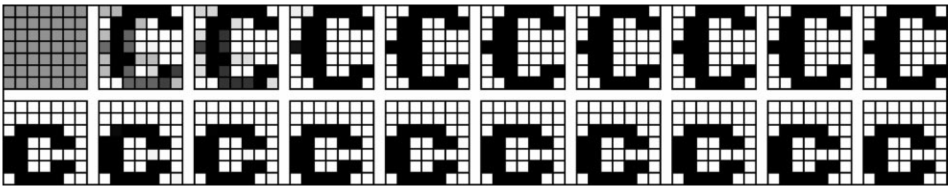


Fig. 4. The simulation result for the third pair.

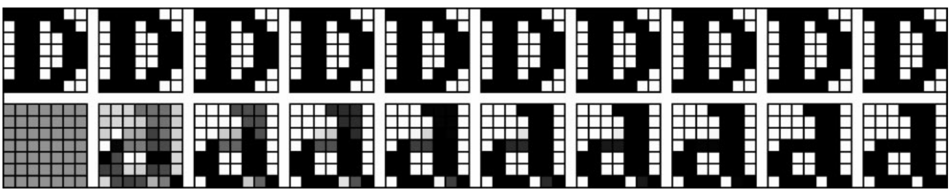


Fig. 5. The simulation result for the fourth pair.

field Y is chosen as the zero vector ("D" recalls "d"). Convergence in all cases occurs within 9 steps.

Using the approach of [29], a BAM neural network is designed which stores the four desired patterns in the present example as memory vectors. Simulation has been conducted for this design using the same initial conditions as in Figs. 2–5. Only the last two pairs showed convergence to the right targets (as in Figs. 4 and 5) and the first two pairs converged to spurious memories. This indicates that the eigenstructure method may have smaller basins of attraction for desired memory patterns than the present approach for this particular example.

5. Concluding remarks

In the present paper, the design problem of bidirectional associative memories (BAM) realized by feedback neural networks is considered. A new synthesis approach for BAM with bipolar memory vectors is developed. The well-known perceptron training algorithm is utilized to solve a set of linear inequalities which is formulated from the BAM neural network design problem. The validity of the present synthesis approach is shown (Theorem 4) and a condition under which the perceptron training will converge is given (Corollary 5). The use of perceptron training in the present synthesis approach is motivated by the way the design problem is formulated. The present formulation is equivalent to designing a set of threshold logic functions which justifies the use of perceptron training algorithm. An example is given to demonstrate the applicability of the present results. A comparison study is also conducted to compare the present approach with one of the existing results.

Acknowledgements

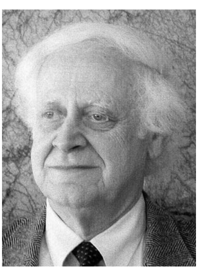
We would like to thank the reviewers for helpful suggestions which improved the presentation of the present paper.

References

[1] S.M.R. Hasan, N.K. Siong, A parallel processing VLSI BAM engine, IEEE Trans. Neural Networks 8 (1997) 424–436.
[2] M.H. Hassoun, Associative Neural Memories: Theory and Implementation, Oxford University Press, New York, NY, 1993.
[3] S. Hui, S.H. Żak, Dynamical analysis of the brain-state-in-a-box (BSB) neural models, IEEE Trans. Neural Networks 3 (1992) 86–94.
[4] B. Kosko, Bidirectional associative memories, IEEE Trans. Syst. Man Cybern. 18 (1988) 49–60.
[5] B. Kosko, Neural Networks and Fuzzy Systems, Prentice-Hall, Englewood Cliffs, NJ, 1992.
[6] D.-L. Lee, W.-J. Wang, Improvement of bidirectional associative memory by using correlation significance, Electron. Lett. 29 (1993) 688–690.
[7] D.-L. Lee, W.-J. Wang, Neighbor-layer updating in MBDS for the recall of pure bipolar patterns in gray-scale noise, IEEE Trans. Neural Networks 6 (1995) 1478–1489.
[8] C.-S. Leung, L.-W. Chan, E.M.K. Lai, Stability and statistical properties of second-order bidirectional associative memory, IEEE Trans. Neural Networks 8 (1997) 267–277.
[9] J.-H. Li, A.N. Michel, W. Porod, Analysis and synthesis of a class of neural networks: linear systems operating on a closed hypercube, IEEE Trans. Circuits Syst. 36 (1989) 1405–1422.
[10] D. Liu, Z. Lu, A new synthesis approach for feedback neural networks based on the perceptron training algorithm, IEEE Trans. Neural Networks 8 (1997) 1468–1482.
[11] D. Liu, A.N. Michel, Sparsely interconnected neural networks for associative memories with applications to cellular neural networks, IEEE Trans. Circuits Systems-II: Analog Digital Signal Process. 41 (1994) 295–307.
[12] D. Liu, A.N. Michel, Robustness analysis and design of a class of neural networks with sparse interconnecting structure, Neurocomputing 12 (1996) 59–76.
[13] A.N. Michel, J.A. Farrell, Associative memories via artificial neural networks, IEEE Control Systems Mag. 10 (1990) 6–17.
[14] A.N. Michel, J. Si, G. Yen, Analysis and synthesis of a class of discrete-time neural networks described on hypercubes, IEEE Trans. Neural Networks 2 (1991) 32–46.
[15] A.N. Michel, K. Wang, D. Liu, H. Ye, Qualitative limitations incurred in implementations of recurrent neural networks, IEEE Control Systems Mag. 15 (1995) 52–65.

[16] M.L. Minsky, S.A. Papert, Perceptrons: an Introduction to Computational Geometry, MIT Press, Cambridge, MA, 1988.
[17] R. Perfetti, A synthesis procedure for brain-state-in-a-box neural networks, IEEE Trans. Neural Networks 6 (1995) 1071–1080.
[18] L. Personnaz, I. Guyon, G. Dreyfus, Information storage and retrieval in spin-glass like neural networks, J. Phys. Lett. 46 (1985) L359–365.
[19] F. Rosenblatt, Principles of Neurodynamics, Spartan, New York, NY, 1962.
[20] F.M.A. Salam, Y. Wang, M.-R. Choi, On the analysis of dynamic feedback neural nets, IEEE Trans. Circuits Syst. 38 (1991) 196–201.
[21] G. Seiler, A.J. Schuler, J.A. Nossek, Design of robust cellular neural networks, IEEE Trans. Circuits Syst.-I: Fund. Theory Appl. 40 (1993) 358–364.
[22] F. A. Valentine, Convex Sets, McGraw-Hill, New York, NY, 1964.
[23] C.-C. Wang, H.-S. Don, An analysis of high-capacity discrete exponential BAM, IEEE Trans. Neural Networks 6 (1995) 492–496.
[24] W.J. Wang, D.L. Lee, A modified bidirectional decoding strategy based on the BAM structure, IEEE Trans. Neural Networks 4 (1993) 710–717.
[25] C.-C. Wang, J.-P. Lee, The decision-making properties of discrete multiple exponential bidirectional associative memories, IEEE Trans. Neural Networks 6 (1995) 993–999.
[26] T. Wang, X. Zhuang, X. Xing, Weighted learning of bidirectional associative memories by global minimization, IEEE Trans. Neural Networks 3 (1992) 1010–1018.
[27] T. Wang, X. Zhuang, X. Xing, Designing bidirectional associative memories with optimal stability, IEEE Trans. Syst. Man Cybern. 24 (1994) 778–790.
[28] Z.-B. Xu, Y. Leung, X.-W. He, Asymmetric bidirectional associative memories, IEEE Trans. Syst. Man Cybern. 24 (1994) 1558–1564.
[29] G.G. Yen, Eigenstructure bidirectional associative memory: an effective synthesis procedure, IEEE Trans. Neural Networks 6 (1995) 1293–1297.

Ismail Salih received the B.S. degree from Reims University France in 1991, the M.S. degree from New York Institute of Technology Old Westbury NY in 1995, and the Ph.D. degree from Stevens Institute of Technology Hoboken NJ in 1998, all in Electrical Engineering. From 1995 to 1997, he was a researcher at the Advanced Telecommunications Institute Hoboken, NJ. From 1997–1999 he served as faculty member of the Electrical and Computer Engineering Department at Stevens Institute of Technology. He is currently a Member of the Technical Staff with Bell Labs, Lucent Technologies. His research interests include neural networks, image and video compression, voice/data communication and network traffic management. He is a member of Eta Kappa Nu and IEEE.



Dr. Stanley H. Smith received the B.A.E., M.A.E., Ph.D. (Engineering Science), and M.E. (honorary) degrees from New York University, Cornell University, New York University and Stevens Institute of Technology, respectively. In 1966, he joined the faculty of the Stevens Institute of Technology in Hoboken, N.J., where he is currently a Professor (Emeritus) of Electrical and Computer Engineering. From 1992 to 1996, he was the Head of the Department of Electrical Engineering and Computer Science.

Dr. Smith has served on technical evaluation and program committees and expert panels for SDIO (now BMDO), BTI, DARPA and Department of the Army programs. The areas that he was involved with were sensors/seekers, electronics, computer architectures, algorithm development and communications. He is a member of the Board of Advisers of the Northeast Region of the Federal Laboratory Consortium (FLC) for Technology Transfer, encompassing all U.S. Government laboratories in New Jersey, New York and New England. He was also a member of the National Advisory Group of the FLC.

His present research interests are concerned with multimedia, interactive video and wireless communications, sensors/seekers, multispectral image processing and pattern recognition (autonomous target recognition), application specific computer architectures (ASCA) for high-speed VLSI and VHSIC realizations including neural networks and embedded computer design. Dr. Smith is the author of more than ninety technical reports and papers. He was a contributing author in books on digital signal processing and on product defects for the legal profession. He is a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE), the IEEE Computer Society, the IEEE Communications Society and the Society for Photographic and Instrumentation Engineers (SPIE).

Derong Liu received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, Indiana, in 1994. From 1993 to 1995, he was with General Motors Research and Development Center, Warren, Michigan, where he worked in automotive engine diagnostics. From 1995 to 1999, he was Assistant Professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, New Jersey. Currently, he is Assistant Professor in the Department of Electrical Engineering and Computer Science, University of Illinois, Chicago, Illinois. He is coauthor of the book Dynamical Systems with Saturation Nonlinearities: Analysis and Design (with A.N. Michel) (New York: Springer–Verlag, 1994). Dr. Liu was a member of the Program Committee of the 11th IEEE International Symposium on Intelligent Control (1996), he was the Local Arrangements Chair for the 6th IEEE Conference on Control Applications (1997), he served as Associate Editor for IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications (1997–1999), and he was a member of the Program Committee of the 14th IEEE International Symposium on Intelligent Control (1999). He is a member of the Conference Editorial Board of the IEEE Control Systems Society, and he is a member of the Program Committee of the 39th IEEE Conference on Decision and Control (2000).