ELSEVIER

Neurocomputing 12 (1996) 59–76

NEUROCOMPUTING

Robustness analysis and design of a class of neural networks with sparse interconnecting structure ☆

Derong Liu [a,*], Anthony N. Michel [b]

[a] *Electrical and Electronics Department, General Motors R&D Center, 30500 Mound Road, Box 9055, Warren, MI 48090-9055, USA*

[b] *Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA*

Received 12 October 1994; accepted 21 March 1995

Abstract

We first conduct an analysis of the robustness properties of a class of neural networks with applications to associative memories. Specifically, for a network with nominal parameters which stores a set of desired bipolar memories, we establish sufficient conditions under which the same set of bipolar memories is also stored in the network with perturbed parameters. This result enables us to establish a synthesis procedure for neural networks whose stored memories are invariant under perturbations. Our synthesis procedure is capable of generating artificial neural networks with prespecified sparsity constraints (on the interconnecting structure) and with nonsymmetric and symmetric interconnection matrices. To demonstrate the applicability of the present results, we consider several specific examples.

Keywords: Sparsely interconnected neural networks; Parameter perturbations; Robustness analysis; Associative memories; VLSI implementation of neural networks

1. Introduction

We consider neural networks described by equations of the form

$$\begin{cases} \dot{x} = -Ax + T\,\mathrm{sat}(x) + I \\ y = \mathrm{sat}(x) \end{cases} \tag{1}$$

where $x \in R^n$ is the state vector, $\dot{x}$ denotes the derivative of x with respect to time t, $y \in D^n \triangleq \{x \in R^n: -1 \le x_i \le 1,\ i = 1, \cdots, n\}$ is the output vector, $A =$

☆ This work was supported by the National Science Foundation under grant ECS 91-07728.

* Corresponding author. Email: dliu@rcsuna.gmr.com

0925-2312/96/$15.00
SSDI 0925-2312(95)00040-2

$\text{diag}[a_1,\cdots,a_n]$ with $a_i > 0$ for $i = 1,\cdots,n$, $T = [T_{ij}] \in R^{n\times n}$ is the coefficient (or connection) matrix, $I = [I_1,\cdots,I_n]^T \in R^n$ is a bias vector, and $\text{sat}(x) = [\text{sat}(x_1),\cdots,\text{sat}(x_n)]^T$ represents the activation function, where

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

We assume that the initial states of (1) satisfy $|x_i(0)| \le 1$ for $i = 1,\cdots,n$.

In [7], we established *analysis* results for system (1) which enable us to locate all equilibrium points and ascertain their qualitative properties. Also, in [6] and [7], we presented a *synthesis* procedure which guarantees to store a *desired* set of bipolar patterns as memories and which results in a *predetermined* interconnecting structure for system (1). Thus, this synthesis procedure enables us to synthesize neural networks which are either *fully* interconnected, or *partially* (or *sparsely*) interconnected.

In the present paper, we address the implementation of *associative memories* via neural networks modeled by (1). In practice the desired memory patterns are usually represented by bipolar vectors (or binary vectors). We will call a vector α a *memory vector* (or simply, a *memory*) of system (1), if $\alpha = \text{sat}(\beta)$ and if β is an asymptotically stable equilibrium point of system (1). We present in Section 2 robustness analysis of the stability properties of bipolar type memory vectors for neural network (1). Specifically, we will assume that $\alpha^1,\cdots,\alpha^m \in B^n \triangleq \{x \in R^n: x_i = 1 \text{ or } -1, i = 1,\cdots,n\}$ are the desired memory vectors of system (1) and we will investigate under what conditions $\alpha^1,\cdots,\alpha^m$ are *also* memory vectors of the perturbed system described by

$$\begin{cases} \dot{x} = -(A+\Delta A)x + (T+\Delta T)\text{sat}(x) + (I+\Delta I) \\ y = \text{sat}(x) \end{cases} \tag{2}$$

where $\Delta A = \text{diag}[\Delta a_1,\cdots,\Delta a_n]$ with $a_i + \Delta a_i > 0$ for $i = 1,\cdots,n$, $\Delta T \in R^{n\times n}$, and $\Delta I \in R^n$. This problem is of great interest from a practical point of view, especially in VLSI implementations of system (1), since one cannot realize *precisely* synthesized parameters $\{A, T, I\}$. We will establish an upper bound for the permissible perturbations ΔA, ΔT, and ΔI in terms of the expression $\|A^{-1}\Delta A\|_\infty + \|A^{-1}\Delta T\|_\infty + \|A^{-1}\Delta I\|_\infty$, where $\|\cdot\|_\infty$ denotes the matrix norm induced by the l_∞ vector norm (see Theorem 1).

Utilizing the above results, we will solve in Section 3 the following problem: Given $\alpha^1,\cdots,\alpha^m \in B^n$ as desired patterns and given a prespecified interconnecting structure, find A, T, and I such that $\alpha^1,\cdots,\alpha^m$ become memory vectors of system (1), such that the system satisfies the prespecified interconnecting structure (in terms of sparsity), and such that the connection matrix is *symmetric*.

In Section 4, we consider several examples to demonstrate the applicability and versatility of the present results, and in Section 5, we conclude with several pertinent remarks.

2. Robustness analysis

In the sequel, we will make use of the notation

$$\delta(x) = \min_{1 \le i \le n} \{|x_i|\} \quad \text{for} \quad x \in R^n$$

and

$$C(\alpha) = \{x \in R^n \colon x_i \alpha_i > 1\}$$

for $\alpha \in B^n = \{x \in R^n \colon x_i = 1 \text{ or } -1,\ i = 1, \cdots, n\}$. Recall that for $x \in R^n$, the l_∞ vector norm is defined by

$$\| x \|_\infty = \max_{1 \le i \le n} \{|x_i|\}.$$

Also recall that the matrix norm induced by the l_∞ vector norm for a matrix $F = [f_{ij}] \in R^{m \times n}$ is defined by

$$\| F \|_\infty = \max_{1 \le i \le m} \left\{ \sum_{j=1}^{n} |f_{ij}| \right\}.$$

We will make use of the following result which has been proven in [7] (cf. Corollary 4 in [7]).

Lemma 1. *Let* $\alpha \in B^n$. *If*

$$\beta = A^{-1}(T\alpha + I) \in C(\alpha)$$

then (α, β) *is a pair of memory vector and asymptotically stable equilibrium point of system* (1).

We are now in a position to prove the next result.

Theorem 1. *Suppose that* $\alpha^1, \cdots, \alpha^m \in B^n$ *are desired memory vectors of system* (1), *and suppose that* $\beta^1, \cdots, \beta^m$ *are asymptotically stable equilibrium points of system* (1) *corresponding to* $\alpha^1, \cdots, \alpha^m$, *respectively. Let*

$$\nu = \min_{1 \le l \le m} \{\delta(\beta^l)\}. \tag{3}$$

Then $\alpha^1, \cdots, \alpha^m$ *are also memory vectors of system* (2) *provided that*

$$\| A^{-1}\Delta A \|_\infty + \| A^{-1}\Delta T \|_\infty + \| A^{-1}\Delta I \|_\infty < \nu - 1. \tag{4}$$

Proof. From Lemma 1, we see that, for $l = 1, \cdots, m$, $\beta^l = A^{-1}(T\alpha^l + I)$, or equivalently, $a_i^{-1}(T_i\alpha^l + I_i) = \beta_i^l$ for $i = 1, \cdots, n$, where a_i is the ith diagonal element of matrix A, T_i represents the i^{th} row of matrix T, and I_i and β_i^l are the ith element of I and β^l, respectively. In the rest of the proof, we assume that

$$\| A^{-1}\Delta T \|_\infty + \| A^{-1}\Delta I \|_\infty \le \eta \tag{5}$$

and

$$\| A^{-1}\Delta A \|_\infty < \nu - \eta - 1 \tag{6}$$

i.e. (4) is satisfied. We will show that $\alpha^1,\cdots,\alpha^m$ are also memory vectors of (2).

For $l=1,\cdots,m$, compute $\Delta T\alpha^l + \Delta I$ and apply (5) to obtain

$$\begin{aligned} | a_i^{-1}\Delta T_i\alpha^l + a_i^{-1}\Delta I_i | &\le \sum_{j=1}^{n} | a_i^{-1}\Delta T_{ij} | + \max_{i\le i\le n} | a_i^{-1}\Delta I_i | \\ &\le \| A^{-1}\Delta T \|_\infty + \| A^{-1}\Delta I \|_\infty \le \eta \end{aligned} \tag{7}$$

where $\Delta T_i = [\Delta T_{i1},\cdots,\Delta T_{in}]$ represents the ith row of ΔT and ΔI_i is the ith component of ΔI.

We now compute

$$\begin{aligned} \bar{\beta}_i^l &\triangleq (a_i + \Delta a_i)^{-1}\left[(T_i + \Delta T_i)\alpha^l + I_i + \Delta I_i\right] \\ &= \frac{a_i}{a_i + \Delta a_i}\left[a_i^{-1}(T_i\alpha^l + I_i) + a_i^{-1}\Delta T_i\alpha^l + a_i^{-1}\Delta I_i\right] \\ &= \frac{a_i}{a_i + \Delta a_i}\left(\beta_i^l + a_i^{-1}\Delta T_i\alpha^l + a_i^{-1}\Delta I_i\right). \end{aligned}$$

From (3) and (7), when $\beta_i^l > 1$ ($\alpha_i^l = 1$), we have

$$\bar{\beta}_i^l \ge \frac{a_i}{a_i + \Delta a_i}\left(\beta_i^l - | a_i^{-1}\Delta T_i\alpha^l + a_i^{-1}\Delta I_i |\right) \ge \frac{a_i}{a_i + \Delta a_i}(\nu - \eta) > 1. \tag{8}$$

Also, when $\beta_i^l < -1$ ($\alpha_i^l = -1$), we have

$$\bar{\beta}_i^l \le \frac{a_i}{a_i + \Delta a_i}\left(\beta_i^l + | a_i^{-1}\Delta T_i\alpha^l + a_i^{-1}\Delta I_i |\right) \le \frac{a_i}{a_i + \Delta a_i}(-\nu + \eta) < -1. \tag{9}$$

Relations (8) and (9) are true since (6) implies that

$$1 + \frac{\Delta a_i}{a_i} \le 1 + \left| \frac{\Delta a_i}{a_i} \right| < \nu - \eta$$

which is equivalent to

$$\frac{a_i}{a_i + \Delta a_i}(\nu - \eta) > 1.$$

(8) and (9) in turn imply that

$$\bar{\beta}^l = (A + \Delta A)^{-1}\left[(T + \Delta T)\alpha^l + (I + \Delta I)\right] \in C(\alpha^l)$$

for $l = 1,\cdots,m$. From Lemma 1, we now see that $\alpha^1,\cdots,\alpha^m$ are also memory vectors for system (2). □

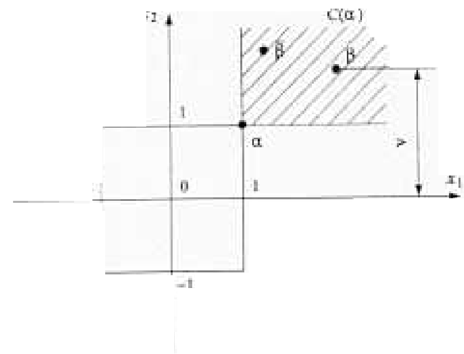

Fig. 1. A geometric interpretation of Theorem 1.

In the following, we give a geometric interpretation of Theorem 1 in R^2. Suppose that $\alpha \in R^2$ is a (desired) memory of system (1) and its corresponding asymptotically stable equilibrium point is β. Then, $\beta = A^{-1}(T\alpha + I)$ *must* be in the region $C(\alpha)$ (cf. the crosshatched region in Fig. 1), since we have $\nu > 1$ in Theorem 1.

When we have perturbations ΔA, ΔT, and ΔI as in system (2), the vector β will be displaced from its original location to, say, $\bar{\beta}$. In order for α to remain as a memory vector for system (1) after perturbation (i.e. for α to be a memory vector for system (2)), we require that $\bar{\beta}$ also be in $C(\alpha)$. It is clear from Lemma 1 that as long as $\bar{\beta}$ is in $C(\alpha)$, α will be a memory vector of the perturbed system (2). Theorem 1 gives one of the *possible* upper bounds for the perturbations, specified by $\| A^{-1}\Delta A \|_\infty + \| A^{-1}\Delta T \|_\infty + \| A^{-1}\Delta I \|_\infty$, which will ensure that the perturbed vector $\bar{\beta}$ and the original vector β are within the same region given by $C(\alpha)$. This upper bound is given by $\nu - 1$ (if ν satisfies condition (3)).

Remark 1. In system (2), we have to require that $a_i + \Delta a_i > 0$ for each i. From Lemma 1, we see that a perturbation ΔA with $\Delta a_i < 0$ for $i = 1, \cdots, n$ will not change the desired memory vectors $\alpha^1, \cdots, \alpha^m \in B^n$ of system (1).

Remark 2. When considering perturbations due to an implementation process, the focus is usually on the interconnection matrix T and not on the parameters A and I. Assuming $\Delta A = 0$ and $\Delta I = 0$, system (2) takes the form

$$\begin{cases} \dot{x} = -Ax + (T + \Delta T)\text{sat}(x) + I \\ y = \text{sat}(x) \end{cases} \tag{10}$$

and condition (4) assumes the form

$$A^{-1}\Delta T \|_\infty \le \nu - 1. \tag{11}$$

3. Synthesis procedure

In this section, we first present a summary of the synthesis procedure developed in [6] and [7], and utilize Theorem 1 to analyze this synthesis procedure. We then present a synthesis procedure for neural network (1) with sparsity and symmetry constraints.

3.1. Summary of the synthesis procedure with sparsity constraints (non-symmetric interconnection matrix)

Suppose we are given a set of desired patterns $\alpha^1,\cdots,\alpha^m \in B^n$. We wish to design a system of form (1) which stores $\alpha^1,\cdots,\alpha^m$ as memories. Without loss of generality, we choose A as the $n \times n$ identity matrix. We denote $Y=[\alpha^1-\alpha^m,\cdots,\alpha^{m-1}-\alpha^m]$, choose $\mu>1$, and set $\beta^l=\mu\alpha^l$ for $l=1,\cdots,m$ (hence, $\beta^l \in C(\alpha^l)$). In view of Lemma 1, it can be verified that in order for system (1) to store the desired patterns $\alpha^1,\cdots,\alpha^m$ as memories and to store $\beta^1,\cdots,\beta^m$ as corresponding asymptotically stable equilibrium points, matrix T must be a solution of the matrix equation,

$$TY=\mu Y. \tag{12}$$

In our implementations of associative memories via neural networks (1), we have *sparsity constraints* on the interconnecting structure. Specifically, we will consider constraints which require that predetermined elements of T be zero. For example, when $n=4$, the constraints on matrix T might be given by

$$T=\begin{bmatrix} T_{11} & 0 & T_{13} & 0 \\ 0 & T_{22} & 0 & T_{24} \\ T_{31} & 0 & T_{33} & 0 \\ 0 & T_{42} & 0 & T_{44} \end{bmatrix}, \tag{13}$$

where the T_{ij}'s are to be determined. The question to be answered is for a given set of vectors $\alpha^1,\cdots,\alpha^m \in B^n$ and $Y=[\alpha^1-\alpha^m,\cdots,\alpha^{m-1}-\alpha^m]$, whether it is possible to find (non-trivial) solutions of T with structure (13) from the matrix Eq. (12). We have shown in [7] that (non-trivial) solutions for such T always exist as long as all the diagonal elements of matrix T are assumed to be non-prespecified elements (e.g. as given in (13)) and $p<n(p=\text{rank}[Y])$. One class of sparsely interconnected neural networks which satisfies the above structural condition are *cellular neural networks*, first introduced by Chua and Yang in 1988 [1,2]. Cellular neural networks (which are also described by Eq. (1)), require that the matrix T have a special *sparse* structure in which all the diagonal elements are required to be non-zero (non-prespecified).

In Section 3.3, we develop a synthesis procedure for associative memories which results in *sparse and symmetric* interconnection matrices T for system (1). To accomplish this, we will make use of the synthesis procedure summarized in the following, originally developed in [6] and [7].

We call a matrix $S=[S_{ij}]\in R^{n\times n}$ an *index matrix* if $S_{ij}=0$ or 1. The *restriction* of a matrix $Q=[Q_{ij}]\in R^{n\times n}$ to index S, denoted by $Q\mid S$, is defined by $Q\mid S=[h_{ij}]$, where

$$h_{ij}=\begin{cases}Q_{ij} & \text{if } S_{ij}=1\\ 0 & \text{otherwise}\end{cases}.$$

3.1.1. Sparse design problem

Given an $n\times n$ index matrix $S=[S_{ij}]$ with $S_{ii}\neq 0$ for $i=1,\cdots,n$, and m vectors $\alpha^1,\cdots,\alpha^m$ in B^n, choose $\{A, T, I\}$ with $T=T\mid S$ in such a manner that $\alpha^1,\cdots,\alpha^m$ are memory vectors of system (1).

3.1.2. Summary of the sparse design procedure

(1) Choose A as the identity matrix.
(2) Choose $\mu>1$ and compute $\beta^i=\mu\alpha^i$ for $i=1,\cdots,m$.
(3) T is solved from $TY=\mu Y$ with the constraints of $T=T\mid S$, where $Y=[\alpha^1-\alpha^m,\cdots,\alpha^{m-1}-\alpha^m]$.
(4) $I=[I_1,\cdots,I_n]^T$ is computed by $I=\mu\alpha^m-T\alpha^m$.

Then, $\alpha^1,\cdots,\alpha^m$ will be stored as memory vectors for system (1) with A, T, and I determined above. The states β^i corresponding to α^i, $i=1,\cdots,m$, will be asymptotically stable equilibrium points of the synthesized system.

3.2. Robustness analysis of the synthesis procedure with sparsity constraints (nonsymmetric interconnection matrix)

In this subsection, we will utilize the robustness analysis result developed in Section 2 to analyze the Sparse Design Procedure summarized above.

The following result can readily be proved.

Corollary 1. *The Sparse Design Procedure (given above) guarantees that $\alpha^1,\cdots,\alpha^m$ are also memory vectors of system (2) provided that*

$$\begin{aligned}&\|A^{-1}\Delta A\|_\infty+\|A^{-1}\Delta T\|_\infty+\|A^{-1}\Delta I\|_\infty\\&\quad=\|\Delta A\|_\infty+\|\Delta T\|_\infty+\|\Delta I\|_\infty<\mu-1.\end{aligned}\tag{14}$$

The above enables us to specify *an upper bound* for the parameter inaccuracies encountered in the implementation of a given design to store a desired set of bipolar patterns in system (1). This bound is chosen by the designer during the initial phase of our design procedure. This type of flexibility does not appear to have been achieved in existing synthesis procedures (e.g. [3–5,8,9]). Specifically, the synthesis procedure advocated above incorporates two features which are very important in the VLSI implementation of artificial neural networks:

(i) it allows the designer to choose a suitable interconnecting structure for the neural network; and
(ii) it takes into account inaccuracies which arise in the realization of the neural network by hardware.

3.3. Synthesis procedure for neural network (1) with sparsity and symmetry constraints on the interconnection matrix

For the A, T, and I determined by the Sparse Design Procedure with $\mu > 1$, let us choose

$$\Delta T = (T^T - T)/2. \tag{15}$$

Then, $T_s \triangleq T + \Delta T = (T + T^T)/2$ is a symmetric matrix. From Theorem 1 (see Remark 2), we note that if

$$A^{-1}\Delta T\|_\infty = \|T^T - T\|_\infty/2 < \mu - 1, \tag{16}$$

the neural network (10) will also store all the desired patterns as memories, with a symmetric connection matrix $T + \Delta T = T_s$.

The above observation gives rise to the possibility of designing a neural network (1) with *prespecified interconnecting structure* and *with a symmetric interconnection matrix*. (Note that in this case, we require that $S = S^T$.) Such capability is of *great* interest since neural network (1) will be *globally stable* when T is symmetric [1]. (Global stability means that for every initial state, the network will converge to some asymptotically stable equilibrium point and periodic and chaotic solutions do not exist.) It appears that (16) might be satisfied by choosing μ sufficiently large. However, it can easily be shown that large μ will usually result in large absolute values of the components of T which in turn may result in a large $\|T^T - T\|_\infty$. Therefore, it is not always possible for (16) to be satisfied by choosing μ large. From (16), we see that if our synthesized matrix T is sufficiently close to its symmetric part $(T + T^T)/2$, or equivalently, if $\|T^T - T\|_\infty$ is sufficiently small, then (16) is satisfied and we are able to design a neural network of form (1) with the following properties: (i) the network stores $\alpha^1, \cdots, \alpha^m$ as memory vectors; (ii) the network has a predetermined (full or sparse) interconnecting structure; and (iii) the connection matrix T of the network is symmetric.

The synthesis procedure summarized in the previous subsection will usually result in a nonsymmetric coefficient matrix T. In the following, we develop an *iterative algorithm* (design procedure) which in most cases will result in a neural network (1) with *symmetric* and *sparse* interconnection. In doing so, we apply Lemma 1 and Theorem 1 (Remark 2) *iteratively*. Let ΔT be defined as in (15). For the given μ (from the Sparse Design Procedure), suppose that $\|\Delta T\|_\infty \geq \mu - 1$. We can find a λ, $0 < \lambda < 1$, such that $\lambda \|\Delta T\|_\infty < \mu - 1$, and we let $T_1 = T + \lambda \Delta T$. We use this T_1 as the *new* connection matrix for our neural network (1). According to Lemma 1 and Remark 2 (Theorem 1), we see that $\alpha^1, \cdots, \alpha^m$ are still memory vectors of system (1) with coefficient matrix T_1, and we can compute the corresponding asymptotically stable equilibrium points as $\bar{\beta}^l = A^{-1}(T_1\alpha^l + I)$ for $l = 1,\cdots,m$. Clearly $\bar{\beta}^l \in C(\alpha^l)$. Using Theorem 1 (Remark 2), we can determine the upper bound ν for the permissible perturbation ΔT as in (3), where we use $\bar{\beta}^l$ instead of β^l. We *repeat* the above procedure, until we determine a symmetric coefficient matrix T or until we arrive at $\nu \leq 1 + \eta$ (where η is a small positive number, e.g. $\eta = 0.001$).

Because of its importance and for the sake of completeness, we summarize in the following our symmetric design procedure.

3.3.1. Symmetric design problem

Suppose we are given an index matrix $S = S^T = [S_{ij}] \in R^{n \times n}$ with $S_{ii} \neq 0$ for $i = 1,\cdots,n$, and m vectors $\alpha^1,\cdots,\alpha m \in B^n$. Choose $\{A, T, I\}$ with $T = T \mid S$ and $T = T^T$ in such a manner that $\alpha^1,\cdots,\alpha^m$ are memory vectors of neural network (1).

3.3.2. Symmetric design procedure

(1) According to the Sparse Design Procedure summarized in Section 3.1, we first choose A as the identity matrix and we determine T and I for neural network (1) with a $\mu > 1 + \eta$ (e.g. $\mu = 10$, $\eta = 0.001$).
(2) If $T = T^T$ or $\mu \leq 1 + \eta$, stop. Otherwise go to step 3.
(3) Compute $\Delta T = (T^T - T)/2$. If $\| \Delta T \|_\infty < \mu - 1$, choose $\lambda = 1$. Otherwise, choose

$$\lambda = \frac{\mu - 1}{\| \Delta T \|_\infty} - \varepsilon$$

where ε is a small positive number (e.g. $\varepsilon = 0.01$). Compute $T_1 = T + \lambda \Delta T$.
(4) Compute $\bar{\beta}^l = A^{-1}(T_1\alpha^l + I)$ for $l = 1,\cdots,m$, and compute $\nu = \min_{1 \leq l \leq m}\{\delta(\bar{\beta}^l)\} > 1$.
(5) Replacing μ by ν and replacing T by T_1, go to step 2.

If we end up with $T = T^T$, we have found a solution for our symmetric design problem. If we end up with $\mu \leq 1 + \eta$ and $T \neq T^T$, our design procedure is not successful in solving a symmetric T for the given problem.

The above design procedure yields a set of parameters $\{A, T, I\}$. For VLSI implementations, these parameters have to be appropriately *scaled*. The theoretical basis for doing this is provided by the following result which was proved in [7].

Corollary 2. *Suppose that β is an asymptotically stable equilibrium point and $\alpha = sat(\beta)$ is a memory vector of system* (1) *with parameters $\{A, T, I\}$. Then, α and*

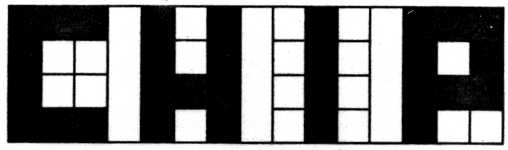

Fig. 2. The four desired memory patterns used in Example 1.

β will also be a pair of memory vector and asymptotically stable equilibrium point of system (1) *with parameters* $\{kA, kT, kI\}$ *for every real number* $k > 0$.

4. Examples

To demonstrate the applicability and versatility of the analysis and synthesis procedures presented in the preceding sections, we consider two specific examples.

Example 1. We consider a neural network with 12 neurons ($n = 12$) with the objective of storing the four patterns shown in Fig. 2 as memories. As indicated in this figure, twelve boxes are used to represent each pattern (in R^{12}), 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, as

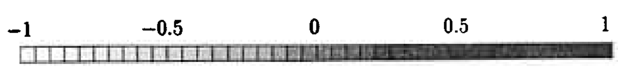

Fig. 3. Grey levels.

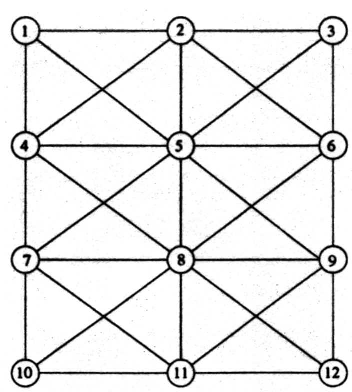

Fig. 4. Interconnecting structure of a cellular neural network.

shown in Fig. 3. The four desired patterns given in Fig. 2 correspond to the following four bipolar vectors:

$$\alpha^1 = [1, 1, 1, 1, -1, -1, 1, -1, -1, 1, 1, 1]^T,$$

$$\alpha^2 = [1, -1, 1, 1, -1, 1, 1, 1, 1, 1, -1, 1]^T,$$

$$\alpha^3 = [-1, 1, -1, -1, 1, -1, -1, 1, -1, -1, 1, -1]^T,$$

and

$$\alpha^4 = [1, 1, 1, 1, -1, 1, 1, 1, 1, 1, -1, -1]^T.$$

In all cases, we seek to design a cellular neural network with the configuration given in Fig. 4 (for details concerning cellular neural networks, see [1] and [2]). The index matrix for this interconnecting structure is as follows, where '0' represents no connection and '1' represents a connection.

$$S = S^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \tag{17}$$

Case I: Nonsymmetric design. We utilize the Sparse Design Procedure summarized in Section 3.1 to design a non-symmetric cellular neural network with the index matrix given in (17). We obtain A as the identity matrix, and we obtain

$$T = \left[\begin{array}{cccccc} 0.333 & -0.000 & 0 & 0.333 & -14.333 & 0 \\ -3.500 & 15.000 & -3.500 & -3.500 & -10.500 & 0 \\ 0 & 0 & 0.500 & 0 & -14.500 & 0.000 \\ 0.250 & 0 & 0 & 0.250 & -14.250 & 0 \\ -5.941 & -0.000 & -5.941 & -5.941 & -8.059 & 0.353 \\ 0 & 0.000 & -5.125 & 0 & -8.875 & 5.625 \\ 0 & 0 & 0 & -2.111 & -11.889 & 0 \\ 0 & 0 & 0 & -7.158 & -6.842 & 0.211 \\ 0 & 0 & 0 & 0 & -3.143 & -0.714 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right.$$

0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0.250	0	0	0	0	0
−5.941	−0.706	0.353	0	0	0
0	3.750	5.625	0	0	0
−2.111	−9.444	0	−2.111	−9.444	0
−7.158	0.368	0.211	−7.158	−14.211	−0.000
0	3.143	−0.714	0	−13.286	0.000
1.800	−11.400	0	1.800	−11.400	0
−5.375	−10.750	−9.125	−5.375	−4.875	−0.000
0	0.000	−7.000	0	−7.000	15.000

and

$$I = [0, 0.000, -0.000, -0.000, 0.706, -3.750, 9.444, 14.632, -3.143, 11.400, 10.750, 0.000]^T.$$

In the above computations, we chose $\mu = 15$ for the Sparse Design Procedure. From Theorem 1 (Remark 2), we see that the upper bound for the admissible perturbation $\| \Delta T \|_\infty$ is $\mu - 1 = 14$. (For simplicity, in all of our examples, we considered $\Delta A = 0$ and $\Delta I = 0$. For the case when they are not zero, we can make similar conclusions and give similar examples.)

The performance of this network is illustrated by means of a typical simulation run of Eq. (1), shown in Fig. 5. In this figure, the desired memory pattern is depicted in the lower right corner. The initial state, shown in the upper left corner, is generated by adding to the desired pattern zero-mean Gaussian noise with a standard deviation SD = 1. The iteration of the simulation evolves from left to

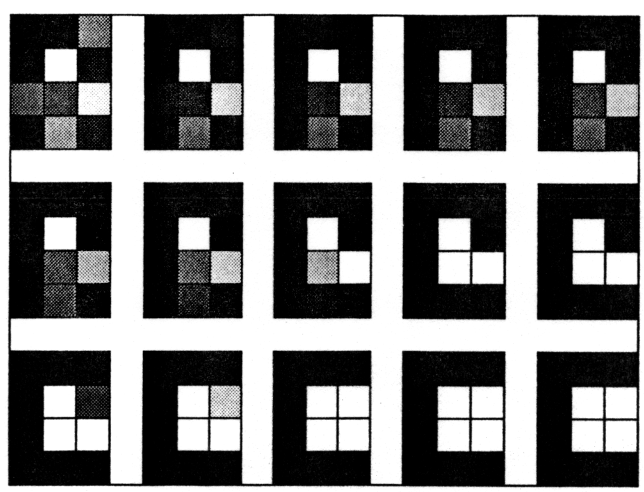

Fig. 5. A typical evolution of pattern No. 1 of Fig. 2.

right in each row and from the top row to the bottom row. The desired pattern is recovered in 13 steps with a step size $h = 0.06$ in the simulation of Eq. (1). All simulations for the present paper were performed on a Sun SPARC Station using MATLAB.

Case II: Nonsymmetric T with perturbations. We generate randomly a matrix $\Delta T = \Delta T \mid S$ as

$$\Delta T = \begin{bmatrix} -1.079 & -1.175 & 0 & 0.875 & 0.303 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1.160 & 0.671 & -0.824 & -1.649 & -0.127 & 1.194 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.363 & -0.064 & 0 & 1.739 & 1.162 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.131 & -1.671 & 0 & 1.839 & -0.105 & 0 & -1.223 & 0.184 & 0 & 0 & 0 & 0 \\ 1.673 & -0.983 & -0.995 & -1.454 & 0.914 & 1.847 & -1.439 & -0.502 & -1.516 & 0 & 0 & 0 \\ 0 & 0.021 & 0.310 & 0 & -1.890 & 1.601 & 0 & 1.305 & 1.150 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.668 & 1.098 & 0 & -0.844 & -1.299 & 0 & -1.232 & -0.947 & 0 \\ 0 & 0 & 0 & 1.031 & -0.988 & -0.007 & -0.178 & -1.643 & 1.635 & -0.588 & 0.107 & -1.951 \\ 0 & 0 & 0 & 0 & 1.027 & 0.026 & 0 & -1.708 & -0.825 & 0 & 1.179 & -0.737 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1.320 & -1.264 & 0 & 1.124 & -0.412 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.911 & 0.032 & -0.720 & -0.843 & 0.358 & -1.862 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.253 & 0.599 & 0 & 1.602 & 0.821 \end{bmatrix}$$

which satisfies the condition that $\| \Delta T \|_\infty < \mu - 1$. We use $T_2 \triangleq T + \Delta T$ in system (10).

Since $\| \Delta T \|_\infty < \mu - 1$, we see from Theorem 1 (Remark 2) that $\alpha^1, \cdots, \alpha^4$ are also memories for system (10). A typical simulation run of Eq. (10) with ΔT given above is depicted in Fig. 6. In this figure, the noisy pattern is generated by adding to the desired pattern uniformly distributed noise defined on $[-1, 1]$. Convergence occurs in 9 steps with $h = 0.06$.

Case III: Symmetric design. Using the Symmetric Design Procedure outlined in Section 3.3, we can easily determine a symmetric matrix T for the present design. Starting with matrix $T_2 = T + \Delta T$ (where T is obtained in Case I and ΔT is obtained in Case II), we determine from Theorem 1 that $\nu = 9.5512$. Choosing

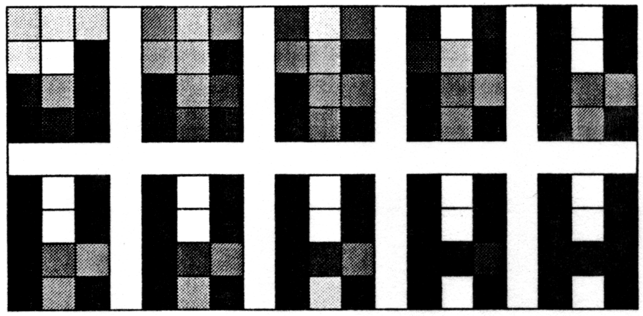

Fig. 6. A typical evolution of pattern No. 2 of Fig. 2.

$\varepsilon = 0.01$ and $\eta = 0.001$ in our Symmetric Design Procedure, we find a symmetric matrix T_3 in four iterations (step 2 to step 5 of the Symmetric Design Procedure) as

$$T_3 = \left[\begin{array}{cccccc} -0.746 & -2.918 & 0 & 0.795 & -9.150 & 0 \\ -2.918 & 15.671 & -1.980 & -3.410 & -5.805 & 0.607 \\ 0 & -1.980 & 0.436 & 0 & -9.849 & -1.827 \\ 0.795 & -3.410 & 0 & 2.089 & -10.875 & 0 \\ -9.150 & -5.805 & -9.849 & -10.875 & -7.145 & -4.282 \\ 0 & 0.607 & -1.827 & 0 & -4.282 & 7.226 \\ 0 & 0 & 0 & -0.708 & -9.086 & 0 \\ 0 & 0 & 0 & -2.971 & -4.519 & 2.629 \\ 0 & 0 & 0 & 0 & -1.640 & 3.043 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right.$$

$$\left.\begin{array}{cccccc} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.708 & -2.971 & 0 & 0 & 0 & 0 \\ -9.086 & -4.519 & -1.640 & 0 & 0 & 0 \\ 0 & 2.629 & 3.043 & 0 & 0 & 0 \\ -2.955 & -9.039 & 0 & -1.431 & -8.339 & 0 \\ -9.039 & -1.274 & 1.641 & -10.205 & -12.411 & -1.602 \\ 0 & 1.641 & -1.539 & 0 & -10.976 & -3.569 \\ -1.431 & -10.205 & 0 & 2.924 & -9.015 & 0 \\ -8.339 & -12.411 & -10.976 & -9.015 & -4.517 & -3.630 \\ 0 & -1.602 & -3.569 & 0 & -3.630 & 15.821 \end{array}\right]$$

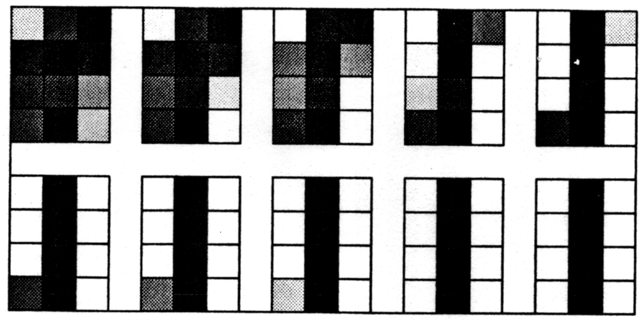

Fig. 7. A typical evolution of pattern No. 3 of Fig. 2.

It can be verified by Lemma 1 that α^1, α^2, α^3, and α^4 are also memories for system (1) with the symmetric matrix T_3 given above. From Theorem 1 (Remark 2), we can verify that the allowable upper bound for the perturbation for system (1) with the above symmetric matrix T_3 is given by $\| \Delta T \|_\infty < 6.2807 - 1 = 5.2807$.

A typical simulation run for system (1) with T_3 given above is shown in Fig. 7. In this case, the noisy pattern is generated by adding Gaussian noise $N(0, 1)$ to the desired pattern. Convergence occurs in 8 steps with $h = 0.06$.

Case IV: Rounded matrix T. We round every component of matrix T_3 obtained in Case III to its closest integer and obtain a matrix T_4 given by

$$T_4 = \begin{bmatrix} -1 & -3 & 0 & 1 & -9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3 & 16 & -2 & -3 & -6 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & -10 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -3 & 0 & 2 & -11 & 0 & -1 & -3 & 0 & 0 & 0 & 0 \\ -9 & -6 & -10 & -11 & -7 & -4 & -9 & -5 & -2 & 0 & 0 & 0 \\ 0 & 1 & -2 & 0 & -4 & 7 & 0 & 3 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & -9 & 0 & -3 & -9 & 0 & -1 & -8 & 0 \\ 0 & 0 & 0 & -3 & -5 & 3 & -9 & -1 & 2 & -10 & -12 & -2 \\ 0 & 0 & 0 & 0 & -2 & 3 & 0 & 2 & -2 & 0 & -11 & -4 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & -10 & 0 & 3 & -9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -8 & -12 & -11 & -9 & -5 & -4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & -4 & 0 & -4 & 16 \end{bmatrix} \tag{18}$$

Using Theorem 1 (Remark 2), we can see that with the matrix T_4, which consists of integers and which is also symmetric, the desired patterns α^l, $l = 1, 2, 3, 4$, are also memories of system (1) with T_4 computed above, since the perturbation which we used to obtain the above $T_4 = T_3 + \Delta T$ satisfies $\| \Delta T \|_\infty < 5.2807$. We can determine the permissible upper bound for the perturbation ΔT to the matrix T_4 as $\| \Delta T \|_\infty < 5$ (by Theorem 1).

A typical simulation run for the present case is depicted in Fig. 8. In this figure, the noisy pattern is generated by adding Gaussian noise $N(0.1, 1)$ to the desired pattern. Convergence occurs in 14 steps with $h = 0.06$.

Example 2. In order to test our Symmetric Design Procedure and to see how typical the results of Case III in Example 1 are, we repeat these examples 200 times using different sets of desired patterns to be stored as memory vectors. Each set contains $m = 4$ vectors in B^{12} which are generated randomly. For each given set of vectors, we synthesize system (1) using the Symmetric Design Procedure.

In these 200 tests, we chose $\mu = 10$ in the Symmetric Design Procedure. There were only eleven tests out of 200 in which we did not succeed in finding a symmetric matrix T for the generated desired patterns and using the above specification for μ. Furthermore, for these eleven failed tests, when we increased μ from 10 to 15, we were able to determine symmetric matrices T again.

Results in Example 2 suggests that our Symmetric Design Procedure will frequently succeed in determining a symmetric matrix T. It also suggests that choosing a larger μ makes it easier to find a symmetric T. However, a large μ will usually result in a matrix T having components with large absolute values. In VLSI implementations of neural networks, we usually want to avoid large values for the parameters (since they correspond to amplifications).

5. Concluding remarks

The results of the present paper complement our earlier results on sparsely interconnected neural networks [6,7]:

(i) We provide upper bounds for the perturbations of parameters under which desired memories stored in a neural network (1) are preserved. This type of

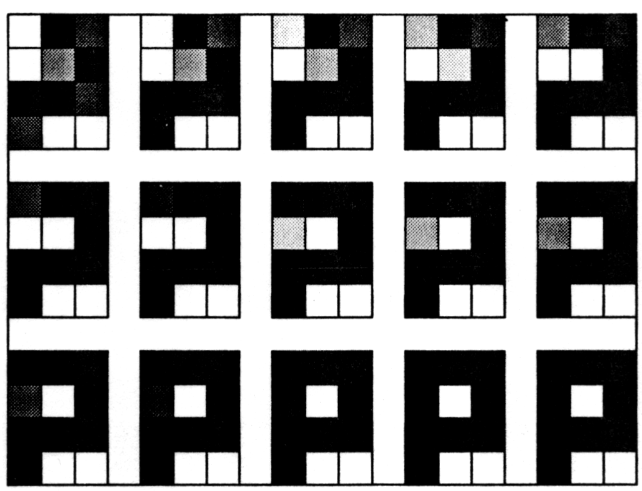

Fig. 8. A typical evolution of pattern No. 4 of Fig. 2.

information is of great practical interest during the implementation process of such networks.

(ii) The Symmetric Design Procedure presented herein enables us to design artificial neural networks with prespecified interconnecting structure and with symmetric interconnection matrix which store a given set of desired bipolar patterns as memories.

References

[1] L.O. Chua and L. Yang, Cellular neural networks: Theory, *IEEE Trans. Circuits Syst.* 35 (10) (1988) 1257–1272.

[2] L.O. Chua and L. Yang, Cellular neural networks: Applications, *IEEE Trans. Circuits Syst.* 35 (10) (1988) 1273–1290.

[3] S.R. Das, On the synthesis of nonlinear continuous neural networks, *IEEE Trans. Syst. Man Cybernet.* 21 (2) (1988) 413–418.

[4] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. Nat. Acad. Sci. USA* 79 (4) (1982) 2554–2558.

[5] J.-H. Li, A.N. Michel, and W. Porod, Analysis and synthesis of a class of neural networks: Linear systems operating on a closed hypercube, *IEEE Trans. Circuits Syst.* 36 (11) (1989) 1405–1422.

[6] D. Liu and A.N. Michel, Cellular neural networks for associative memories, *IEEE Trans. Circuits Syst.-II: Analog and Digital Signal Processing* 40 (2) (1993) 119–121.

[7] D. Liu and A.N. Michel, Sparsely interconnected neural networks for associative memories with applications to cellular neural networks, *IEEE Trans. Circuits Syst.-II: Analog and Digital Signal Processing* 41 (4) (1994) 295–307.

[8] A.N. Michel, J. Si and G. Yen, Analysis and synthesis of a class of discrete-time neural networks described on hypercubes, *IEEE Trans. Neural Networks* 2 (1) (1991) 32–46.

[9] F.M.A. Salam, Y. Wang and M.-R. Choi, On the analysis of dynamic feedback neural nets, *IEEE Trans. Circuits Syst.* 38 (2) (1991) 196–201.

Derong Liu received the BS degree in mechanical engineering from East China Institute of Technology in 1982, the MS degree in electrical engineering from Institute of Automation, Chinese Academy of Sciences in 1987, and the PhD degree in electrical engineering from University of Notre Dame in 1993. During his first year of graduate study at Notre Dame (1990–1991), he received the Michael J. Birck Fellowship.

From 1982 to 1984, he worked at China North Industries Corp., Jilin, China. From 1987 to 1990, he was a faculty member in the Department of Electrical Engineering, Graduate School of Chinese Academy of Sciences, Beijing, China. Currently, he is with the Electrical and Electronics Department, General Motors R & D Center, Warren, MI. He is coauthor of the book *Dynamical Systems with Saturation Nonlinearities: Analysis and Design* (with A.N. Michel) (London, Springer-Verlag, 1994). His current research interests include control theory, systems theory, neural networks, signal processing, and fault diagnosis. He is a member of the Conference Editorial Board (CEB) of the IEEE Control Systems Society. He is a member of Eta Kappa Nu.

Anthony N. Michel received the PhD degree in electrical engineering from Marquette University and the DSc degree in applied mathematics from the Technical University of Graz, Austria.

He has seven years of industrial experience. From 1968 to 1984, he was at Iowa State University, Ames, IA. From 1984 to 1988, he was Frank M. Freimann Professor and Chairman of the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN. Currently, he is Frank M. Freimann Professor and Matthew H. McCloskey Dean of the College of Engineering at the University of Notre Dame. He is author and coauthor of five texts and several other publications.

Dr. Michel received the 1978 Best Transactions Paper Award of the IEEE Control Systems Society, the 1984 Guillemin-Cauer Award from the IEEE Circuits and Systems Society, the 1993 Myril B. Reed Outstanding Paper

Award of the IEEE Circuits and Systems Society, the 1995 Technical Achievement Award of the IEEE Circuits and Systems Society, and an IEEE Centennial Medal. He is a former Associate Editor and a former Editor of the IEEE Transactions on Circuits and Systems, a former Associate Editor of the IEEE Transactions on Automatic Control, and a former Associate Editor of the IEEE Transactions on Neural Networks. He was the Program Chairman of the 1985 IEEE Conference on Decision and Control. He was President of the IEEE Circuits and Systems Society in 1989. He was Co-Chairman of the 1990 IEEE International Symposium on Circuits and Systems. He was a Fulbright Scholar in 1992 (at the Technical University of Vienna, Austria). He is an Associate Editor at Large for the IEEE Transactions on Automatic Control, he is a member of the IEEE Neural Networks Council, and he is an elected member of the Board of Governors of the IEEE Control Systems Society. He is a Foreign Member of the Academy of Engineering of the Russian Federation. He is the Vice President of Technical Affairs of the IEEE Control Systems Society.