PERGAMON

Neural Networks 14 (2001) 107–113

Neural Networks

www.elsevier.com/locate/neunet

A synthesis procedure for associative memories based on space-varying cellular neural networks

J. Park[a,*], H.-Y. Kim[b], Y. Park[c], S.-W. Lee[d]

[a]*Department of Control and Instrumentation Engineering, Korea University, Chochiwon, Chungnam, 339-800, South Korea*
[b]*Department of Visual Image Processing, Korea University, Seoul, 136-701, South Korea*
[c]*Division of Aerospace Engineering, Department of Mechanical Engineering, KAIST, Taejon, 305-701, South Korea*
[d]*Department of Computer Science and Engineering, Korea University, Seoul, 136-701, South Korea*

Received 9 March 2000; revised 5 October 2000; accepted 5 October 2000

Abstract

In this paper, we consider the problem of realizing associative memories via space-varying CNNs (cellular neural networks). Based on some known results and a newly derived theorem for the CNN model, we propose a synthesis procedure for obtaining a space-varying CNN that can store given bipolar vectors with certain desirable properties. The major part of our synthesis procedure consists of solving generalized eigenvalue problems and/or linear matrix inequality problems, which can be efficiently solved by recently developed interior point methods. The validity of the proposed approach is illustrated by a design example.

Keywords: Associative memory; Cellular neural network; Generalized eigenvalue problem; Linear matrix inequality problem

1. Introduction

CNNs (cellular neural networks) as introduced by Chua and Yang (1988) are a special class of continuous-time feedback neural networks which consist of cells connected only to their neighborhood cells. CNNs are well suited for analog very-large-scale integration implementation due to this local interconnection property, and have found many applications in a variety of areas (see, e.g. Roska & Vandewalle, 1995, and references therein). In this paper, we consider the problem of realizing associative memories via space-varying CNNs, in which the cells have local interconnections with identical neighborhood size, but with cell dependent connection weights. Since Hopfield (1982) showed that fully interconnected feedback neural networks trained by the Hebbian learning rule can function as a new concept of associative memories, the synthesis of associative memories using neural network models has attracted a great deal of interest among researchers (see, e.g. Hassoun, 1993). Recently, associative memories based on CNNs also have been studied. Generally, the goal of the CNN memories is to store desired bipolar vectors as memory vectors of the network so that when a vector sufficiently close to a stored bipolar vector is applied as an initial condition of the network, the stored bipolar vector may be retrieved as the final output of the network. In order to design space-varying CNNs that can achieve this goal, several methods have been proposed. Liu and Michel (1993) proposed a singular value decomposition-based synthesis procedure. This procedure is a modification of the synthesis method for neural associative memories (Li, Michel & Porod, 1989), and is often called 'the eigenstructure method'. Seiler, Schuler, and Nossek (1993) developed an algorithm to design a space-varying CNN with prescribed stable and unstable output patterns while maximizing its robustness with respect to changes of its parameters. The algorithm consists of formulating and solving a set of linear inequalities using linear programming. Liu and Lu (1997) showed that the perceptron training algorithm can be applied to design space-varying CNNs as well as fully connected feedback neural networks. Chan and Zak (1998) proposed 'designer' neural network for the synthesis of associative memories based on a class of discrete cellular neural networks. Perfetti (1999) presented a local learning algorithm which can efficiently be implemented on chip by exploiting the parallel analog computation of CNNs. In Park and Park (2000) we proposed a GEVP (generalized eigenvalue problem)-based synthesis procedure for space-varying CNNs. The procedure yields the CNN memories whose connection weight matrices are symmetric with their diagonal elements fixed at unity. Thus, it is not applicable when

* Corresponding author. Tel.: +82-41-860-1444; fax: +82-41-865-1820.
E-mail address: jpark@tiger.korea.ac.kr (J. Park).

0893-6080/01/$ - see front matter
PII: S0893-6080(00)00086-1

desired vectors cannot be stored with connection weight matrices constrained as such.

In this paper, we are concerned with developing a procedure to find the parameters (i.e. the connection weight matrix and the bias vector) of a nonsymmetric space-varying CNN[1] that can achieve the forenamed design goal without constraints on the diagonals of the connection weight matrix. We will first present a theorem which addresses conditions for storing each desired bipolar vector with attractivity in a certain sense and with the absence of spurious patterns (i.e. undesirable memory vectors) near it. Next, based on some known results and the presented theorem, we will propose a synthesis procedure for finding the parameters of the CNN memory. The major part of the proposed synthesis procedure consists of solving GEVPs and/or LMIPs (linear matrix inequality problems), which can be efficiently solved by recently developed interior point algorithms (see, e.g. Vandenberghe & Balakrishnan, 1997) implemented in readily available software (e.g. MATLAB LMI Control Toolbox (Gahinet, Nemirovski, Laub, & Chilali, 1995). The synthesis method of this paper is closely related to the strategy of our previous paper (Park and Park, 2000), and may be viewed as a continuing effort to widen applicability and achieve better performance.

Throughout this paper we use the following definitions and notation, in which R^n denotes the normed linear space of real n-vectors with the Euclidean norm $\|\cdot\|$. A symmetric matrix $\mathbf{A} \in R^{n\times n}$ is positive definite if $\vec{x}^T\mathbf{A}\vec{x} > 0$ for any $\vec{x} \neq 0$, and $\mathbf{A} > 0$ denotes this. Also, $\mathbf{A} > \mathbf{B}$ denotes that $\mathbf{A} - \mathbf{B}$ is positive definite. H^n denotes the hypercube $[-1, +1]^n$. The set of all the bipolar vectors in H^n is denoted by B^n. The usual Hamming distance between two bipolar vectors $\vec{\alpha}$ and $\vec{\alpha}^*$ is denoted by $H(\vec{\alpha}, \vec{\alpha}^*)$. The following typographical conventions are used throughout this paper for expressing scalars, vectors and matrices: italics (e.g. α_i, b_i, T_{ij}) for scalars, italics with $\rightarrow$ attached upon the letter (e.g. $\vec{\alpha}, \vec{x}_e$) for vectors, bold capitals (e.g. **T**, **W**) for matrices.

The remaining part of this paper is organized as follows: in Section 2, we briefly introduce some fundamentals on CNNs and present a theorem, which will play a critical role in our synthesis method. In Section 3, utilizing the material in Section 2, we establish a GEVP/LMIP-bsed synthesis procedure for nonsymmetric space-varying CNN memories. In Section 4, we present a design example to illustrate the validity of the proposed approach. Finally, in Section 5, concluding remarks are given.

2. Background results and main theorem

In this paper, we consider the problem of realizing associative memories via two-dimensional CNNs described by the following equations (Chua & Yang, 1988; Liu & Michel, 1993):

$$\begin{cases} \dot{v}_{xij} = -v_{xij} + \Sigma_{C(k,l)\in N_r(i,j)} W_{ij,kl} v_{ykl} + d_{ij} \\ v_{yij} = \mathrm{sat}(v_{xij}) \end{cases} \tag{1}$$

where $1 \le i \le M, 1 \le j \le N$. This CNN consists of $n = MN$ cells, each of which is connected to its r-neighborhood in a two-dimensional $M \times N$ array. The meaning of each element appearing in (1) is as follows: v_{xij} and v_{yij} represent the state and the output of the (i, j)th cell, $C(i, j)$, respectively. $N_r(i, j)$ is the r-neighborhood of cell $C(i, j)$, which is defined by

$$N_r(i,j) \triangleq \{C(k,l) : \max\{|k-i|, |l-j|\} \le r, 1 \le k \le M, 1 \le l \le N\}.$$

$W_{ij,kl}$ denotes the connection weight from cell $C(k, l)$ to cell $C(i, j)$, and d_{ij} denotes the bias term for cell $C(i, j)$. sat : $R \rightarrow [-1, 1]$ is the linear saturating function defined by

$$\mathrm{sat}(\bar{v}) \triangleq \frac{1}{2}(|\bar{v}+1| - |\bar{v}-1|).$$

With $\mathbf{W} = [W_{ij,kl}] \in R^{MN\times NM}$ expressed as an $n \times n$ matrix $\mathbf{T} = [T_{ij}] = \mathbf{W}$ and with $\vec{d} = [d_{ij}] \in R^{MN}$ expressed as an n-dimensional vector $\vec{b} = [b_i] = \vec{d}$, system (1) can be repacked as the following matrix-vector form

$$\begin{cases} \dot{\vec{x}} = -\vec{x} + \mathbf{T}\mathrm{sat}(\vec{x}) + \vec{b} \\ \vec{y} = \mathrm{sat}(\vec{x}) \end{cases} \tag{2}$$

where $\vec{x} = [v_{x11}, v_{x12}, \ldots, v_{x21}, \ldots, v_{xMN}]^T \in R^n$ is the state vector, $\vec{y} = [v_{y11}, v_{y12}, \ldots, v_{y21}, \ldots, v_{yMN}]^T \in H^n$ is the output vector, $\mathbf{T} \in R^{n\times n}$ is the connection weight matrix, $\vec{b} \in R^n$ is the bias vector, and $\mathrm{sat}(\vec{x}) \triangleq [\mathrm{sat}(x_1), \ldots, \mathrm{sat}(x_n)]^T$. The above type of CNNs are usually called space-varying CNNs, because their cells have local interconnections with identical neighborhood radius r, but with cell-dependent connection weights. Note that **T** in (2) is a sparse matrix whose structure is dependent on the neighborhood radius. To express the fashion of sparsity explicitly, we make use of the index matrix $\mathbf{S} = [S_{ij}] \in \{0, 1\}^{n\times n}$. Each element of the index matrix is 1 or 0, which represent 'connected' or 'disconnected', respectively. $\mathbf{T} = \mathbf{T}|_{\mathbf{S}}$ represents that the interconnection between cells is indicated by the index matrix **S**, i.e. the connection weight T_{ij} can have nonzero value only when $S_{ij} = 1$. In the discussion on the qualitative properties of the CNN model (2), the following definitions are used:

- A point $\vec{x}_e \in R^n$ is an equilibrium point of system (2) if $\vec{x}(0) = \vec{x}_e$ implies $\vec{x}(t) = \vec{x}_e, \forall t > 0$.
- An equilibrium point $\vec{x}_e \in R^n$ of system (2) is stable if for any $\varepsilon > 0$, there exists $\delta > 0$ such that

  $$\|\vec{x}(0) - \vec{x}_e\| < \delta \text{ implies } \|\vec{x}(t) - \vec{x}_e\| < \epsilon, \forall t > 0.$$

- An equilibrium point $\overrightarrow{x_e}$ of system (2) is asymptotically

[1] By a nonsymmetric space-varying CNN, we mean a space-varying CNN whose connection weight matrix is not necessarily symmetric.

stable if it is stable and there exists $\delta > 0$ such that

$\vec{x}(t) \rightarrow \vec{x}_e$ as $t \rightarrow \infty$ if $\|\vec{x}(0) - \vec{x}_e\| < \delta$.

- $\vec{\alpha} \in H^n$ is a memory vector of system (2) if there exists an asymptotically stable equilibrium point $\vec{\beta} \in R^n$ of system (2) such that $\vec{\alpha} = sat(\vec{\beta})$.

In order to work as satisfactory associative memories, the CNN memories should have the following properties:

- The desired bipolar vectors, which are called the prototype patterns in this paper, are stored as memory vectors of the network.
- The number of memory vectors which do not correspond to the prototype patterns (i.e. spurious patterns) is small. Especially, the number of spurious patterns outside B^n (i.e. nonvertex spurious patterns) should be minimal.

The following, which are some of the known results useful for checking the above, will act as important foundations for our synthesis method:

- A bipolar vector $\vec{\alpha} \in \beta^n$ is a memory vector of system (2) if and only if

$$\alpha_i \left(\sum_{j=1}^{n} T_{ij}\alpha_j + b_i \right) > 1, i = 1, \ldots, n \tag{3}$$

(Liu and Michel, 1993).

- If

$$T_{ii} \geq 1, i = 1, \ldots, n, \tag{4}$$

then memory vectors of system (2) can exist only in B^n (Liu & Lu, 1997).

In the following, we present a new theorem, which will play a central role in our synthesis method for the CNN memories. We make use of the following notation for clear and convenient presentation of the theorem:

$$Z_i \triangleq \{j \big| S_{ij} = 1, j \neq i\}.$$

Theorem. Given the index matrix $\mathbf{S} \in \{0, 1\}^{n \times n}$ and a prototype pattern $\vec{\alpha}^* \in B^n$, suppose that for an integer $i \in \{1, \ldots, n\}$, the CNN parameters $\mathbf{T} = \mathbf{T}|_{\mathbf{S}} \in R^{n \times n}$ and $\vec{b} \in R^n$ satisfy

$$\alpha_i^* \left(\sum_{j \in Z_i} T_{ij}\alpha_j^* + b_i \right) > \kappa_i \max_{j \in Z_i} \left| T_{ij} \right| + (T_{ii} - 1) \tag{5}$$

with $\kappa_i \geqslant 0$. Then any bipolar vector $\vec{\alpha} \in B^n$ such that $\alpha_i \neq \alpha_i^*$ and $\sum_{j \in Z_i} |\alpha_j - \alpha_j^*| < \kappa_i$ has the following properties:

(a). If $\vec{x}(0) = \vec{\alpha}$, then $x_i(t)$ moves toward α_i^* at $t = 0$.
(b). $\vec{\alpha}$ is not a memory vector.

Proof. Let $\vec{\alpha} \in B^n$ be any bipolar vector satisfying $\alpha_i \neq \alpha_i^*$ (i.e.$\alpha_i = -\alpha_i^*$) and $\sum_{j \in Z_i} |\alpha_j - \alpha_j^*| < \kappa_i$. Since $\vec{\delta} \triangleq \vec{\alpha} - \vec{\alpha}^*$ satisfies

$$\left| \sum_{j \in Z_i} T_{ij}\delta_j \right| = \left| \sum_{j \in Z_i} T_{ij}(\alpha_j - \alpha_j^*) \right|$$

$$\leq (\max_{j \in Z_i} |T_{ij}|) \left(\sum_{j \in Z_i} |\alpha_j - \alpha_j^*| \right) \leq \kappa_i \max_{j \in Z_i} |T_{ij}|$$

we have

$$\alpha_i \left(\sum_{j=1}^{n} T_{ij}\alpha_j + b_i \right) = \alpha_i \left(T_{ii}\alpha_i + \sum_{j \in Z_i} T_{ij}\alpha_j + b_i \right)$$

$$= T_{ii} + \alpha_i \left(\sum_{j \in Z_i} T_{ij}\alpha_j + b_i \right)$$

$$= T_{ii} - \alpha_i^* \left(\sum_{j \in Z_i} T_{ij}\alpha_j^* + b_i + \sum_{j \in Z_i} T_{ij}\delta_j \right)$$

$$\leq T_{ii} - \alpha_i^* \left(\sum_{j \in Z_i} T_{ij}\alpha_j^* + b_i \right) + \left| \sum_{j \in Z_i} T_{ij}\delta_j \right|$$

$$\leq T_{ii} - \alpha_i^* \left(\sum_{j \in Z_i} T_{ij}\alpha_j^* + b_i \right) + \kappa_i \max_{j \in Z_i} \left| T_{ij} \right|$$

$$= -\left[\alpha_i^* \left(\sum_{j \in Z_i} T_{ij}\alpha_j^* + b_i \right) - \kappa_i \max_{j \in Z_i} |T_{ij}| - (T_{ii} - 1) \right] + 1.$$

This, together with (5), imply

$$\alpha_i \left(\sum_{j=1}^{n} T_{ij}\alpha_j + b_i \right) < 1. \tag{6}$$

Now, let system (2) start from $\vec{x}(0) = \vec{\alpha}$. Since the ith component of the system's state vector satisfies

$$\dot{x}_i(0) = -x_i(0) + \sum_{j=1}^{n} T_{ij}\text{sat}(x_j(0)) + b_i$$

$$= -\alpha_i + \sum_{j=1}^{n} T_{ij}\alpha_j + b_i = \alpha_i^* + \sum_{j=1}^{n} T_{ij}\alpha_j + b_i,$$

we have

$$\alpha_i^* \dot{x}_i(0) = \alpha_i^* \left(\alpha_i^* + \sum_{j=1}^{n} T_{ij}\alpha_j + b_i \right) = 1 - \alpha_i \left(\sum_{j=1}^{n} T_{ij}\alpha_j + b_i \right) > 0.$$

This inequality shows that α_i^* and $\dot{x}_i(0)$ have the same sign, which proves (a). Finally, since (6) violates condition (3), we can conclude that (b) is true. This completes the proof.

Remark. Since inequality (5) in the theorem is equivalent to:

$$\alpha_i^*\left(\sum_{j=1}^{n} T_{ij}\alpha_j^* + b_i\right) = T_{ii} + \alpha_i^*\left(\sum_{j\in Z_i} T_{ij}\alpha_j^* + b_i\right)$$

$$> \kappa_i \max_{j\in Z_i}\left|T_{ij}\right| + (2T_{ii} - 1),$$

the existence of $\kappa_i \geq 0$ satisfying (5) under the condition of (4) implies

$$\alpha_i^*\left(\sum_{j=1}^{n} T_{ij}\alpha_j^* + b_i\right) > 1.$$

Thus, it is clear from criterion (3) that ensuring (5) with $\kappa_i \geq 0$ for each $i \in \{1,\ldots,n\}$, together with (4), guarantee the prototype pattern $\vec{\alpha}^* \in B^n$ to be stored as a memory vector of the network.

3. A new synthesis procedure for cellular neural networks

In this section, we will establish a new synthesis procedure for space-varying CNNs based on the material of the previous section. The major part of our synthesis procedure consists of solving GEVPs and/or LMIPs, both of which are LMI (linear matrix inequality)-based problems. (For details on LMI-based problems, see, e.g. Boyd et al., 1994.) An LMI is any constraint of the form

$$\mathbf{A}(\vec{z}) \triangleq \mathbf{A}_0 + z\mathbf{A}_1 + \cdots + z_N\mathbf{A}_N > 0, \tag{7}$$

where $\vec{z} \triangleq [z_1 \cdots z_N]^T$ is the variable, and $\mathbf{A}_0, \ldots, \mathbf{A}_N$ are given symmetric matrices. Note that LMIs include simple linear inequalities like (3). The LMI problem (LMIP) corresponding to LMI (7) is to find a solution $\vec{z}^o \in R^N$ such that $\mathbf{A}(\vec{z}^o) > 0$ or determine that the LMI is infeasible. The GEVP is to minimize the maximum generalized eigenvalue of a pair of matrices that depend affinely on a vector variable, subject to LMI constraints. The general form of a GEVP is given in the following:

$$\min \lambda$$

$$s.t.\ \lambda\mathbf{B}(\vec{z}) - \mathbf{A}(\vec{z}) > 0$$

$$\mathbf{B}(\vec{z}) > 0$$

$$\mathbf{C}(\vec{z}) > 0$$

where $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are symmetric matrices that are affine functions of the variable $\vec{z}$. Both LMIPs and GEVPs can be efficiently solved by recently developed interior point algorithms.

From the aforementioned known results and our theorem, we can get guidelines for our synthesis. First, because storing each prototype pattern as a memory vector and preventing the occurrence of nonvertex memory vectors are two of the most fundamental requirements for the CNN memories, conditions (3) and (4) are enforced with top priority in our synthesis. Also, we require each parameter of the CNN memories bounded appropriately (i.e. $|T_{ij}| < U$ and $|b_i| < U$ and $i = 1, \ldots, n$, where $U > 0$ is an upper bound chosen by a designer).

Next, we observe from our theorem that with each $\kappa_i\ (\geqslant 0)$ getting larger, the neighboring region of the prototype pattern $\vec{\alpha}^*$, where spurious patterns do not exist and trajectories starting from which are very likely to converge to $\vec{\alpha}^*$, become larger. Thus, in our synthesis, we take the strategy to maximize κ_i when (5) is feasible with a nonnegative κ_i. Note that maximizing κ_i in (5) entails minimizing T_{ii}, the solution of which is obviously $T_{ii} = 1$ under the condition of (4). Also, note that when $T_{ii} = 1$, the feasibility of (5) with a nonnegative κ_i is equivalent to

$$\alpha_i^*\left(\sum_{j\in Z_i} T_{ij}\alpha_j^* + b_i\right) > 0. \tag{8}$$

Thus, our strategy when (8) is feasible can be summed up as follows: Set $T_{ii} = 1$, and find $T_{ij} \in (-U, U), j \in Z_i$, and $b_i \in (-U, U)$ maximizing $\kappa_i \geqslant 0$ such that

$$\alpha_i^*\left(\sum_{j\in Z_i} T_{ij}\alpha_j^* + b_i\right) > \kappa_i \max_{j\in Z_i}\left|T_{ij}\right|. \tag{9}$$

Note that this optimization problem has nonlinear constraints (9), which prevent us from applying linear programming techniques. In this paper we convert this problem into a GEVP. For the conversion, we restrict our attention to the nontrivial case such that $\max_{j\in Z_i}|T_{ij}| \neq 0$ and $\kappa_i > 0$, and for the case, we introduce a positive variable $q_i \in (L, U)$. With $\kappa_i q_i$ inserted in between, (9) can be rewritten as follows:

$$\alpha_i^*\left(\sum_{j\in Z_i} T_{ij}\alpha_j^* + b_i\right) > \kappa_i q_i > \kappa_i \max_{j\in Z_i}\left|T_{ij}\right| \tag{10}$$

Note that (10) can be separated into

$$\alpha_i^*\left(\sum_{j\in Z_i} T_{ij}\alpha_j^* + b_i\right) - \kappa_i q_i > 0$$

and

$$\left|T_{ij}\right| < q_i, j \in Z_i.$$

As a result of the above conversion process, our synthesis strategy when (8) is feasible can be restated as follows: Set $T_{ii} = 1$ and find T_{ij}, $j \in Z_i$ and b_i by solving

$$\min\ (-\kappa_i)$$

$$\text{s.t. } (-\kappa_i)q_i + \alpha_i^*\left(\sum_{j\in Z_i} T_{ij}\alpha_j^* + b_i\right) > 0$$

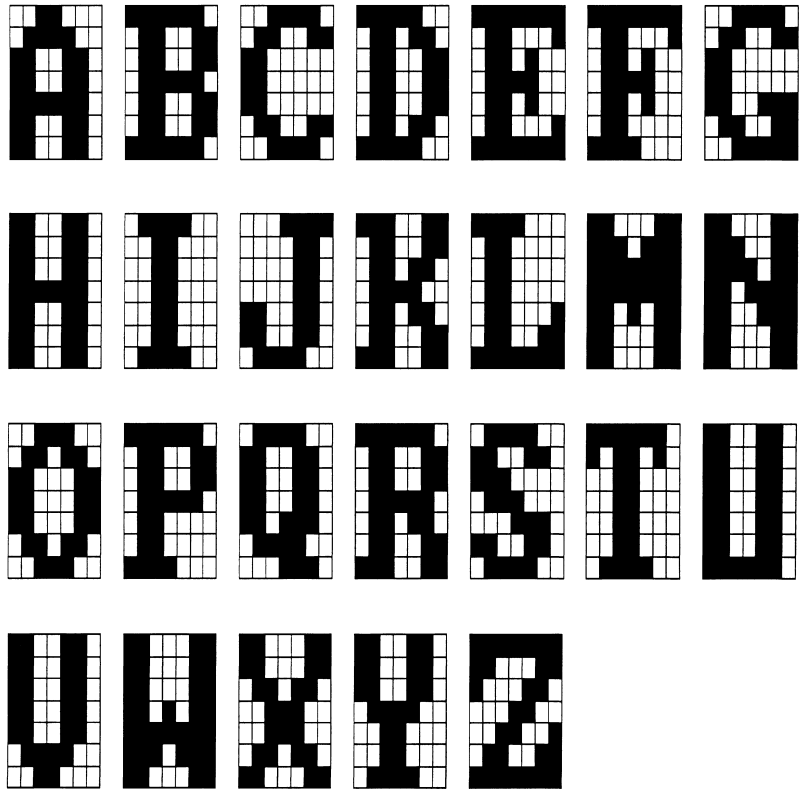

Fig. 1. Prototype patterns.

$$\left|T_{ij}\right| < q_i, j \in Z_i$$

$$|b_i| < U$$

$$L < q_i < U.$$

When condition (8) cannot be satisfied for any $T_{ij} \in (-U, U)$ $j \in Z_i$ and $b_i \in (-U, U)$, the above strategy is not applicable. In this case, we will resort to fundamental requirements (3) and (4) only. More precisely, when $\alpha_i^*\left(\sum_{j\in Z_i} T_{ij}\alpha_j^* + b_i\right) > 0$ is not feasible, our synthesis will be based on finding $T_{ij} \in (-U, U)$, $j \in Z_i \cup \{i\}$ and $b_i \in (-U, U)$ such that

$$\begin{cases} \alpha_i^*\left(\displaystyle\sum_{j=1}^{n} T_{ij}\alpha_j^* + b_i\right) = T_{ii} + \alpha_i^*\left(\displaystyle\sum_{j\in Z_i} T_{ij}\alpha_j^* + b_i\right) > 1 \\ T_{ii} \geq 1. \end{cases} \tag{11}$$

Note that when T_{ii} is sufficiently large, (11) can be trivially satisfied. However, large T_{ii} tends to increase the number of spurious patterns of the network. Especially, when $T_{ii} \geq \sum_{j\in Z_i}\left|T_{ij}\right| + \left|b_i\right| + 1$ for each $i \in \{1, \ldots, n\}$, every vertex of H^n will be stored as a memory vector of the network, which certainly is not desirable in associative memories. To prevent this side effect, we will fix $T_{ii} = 1 + \varepsilon$ in which ε is a small positive number. Thus, our synthesis strategy when (8) is not feasible can be summarized as follows: Set $T_{ii} = 1 + \varepsilon$, and find $T_{ij} \in (-U, U)$, $j \in Z_i$ and $b_i \in (-U, U)$ such that

$$\alpha_i^*\left(\sum_{j\in Z_i} T_{ij}\alpha_j^* + b_i\right) > -\epsilon$$

Note that this inequality always has solutions.

With the arguments up to this point, we can establish a new synthesis procedure for the CNN memories. Given the index matrix $\mathbf{S} \in \{0,1\}^{n\times n}$ for the considered space-varying CNN model, and given a set of prototype patterns $\vec{\alpha}^{(1)}, \cdots \vec{\alpha}^{(n)} \in B^n$, the parameters (i.e. $\mathbf{T} = \mathbf{T}|_{\mathbf{S}} \in R^{n\times n}$ and $\vec{b} \in R^n$) of the CNN memory with desirable properties (as stated in our theorem) can be found by the following procedure. For each $i \in \{1, \ldots, n\}$:

- if

$$\alpha_i^{(k)}\left(\sum_{j\in Z_i} T_{ij}\alpha_j^{(k)} + b_i\right) > 0, k = 1, \ldots, m \tag{12}$$

  is feasible:
  - set $T_{ii} = 1$, and

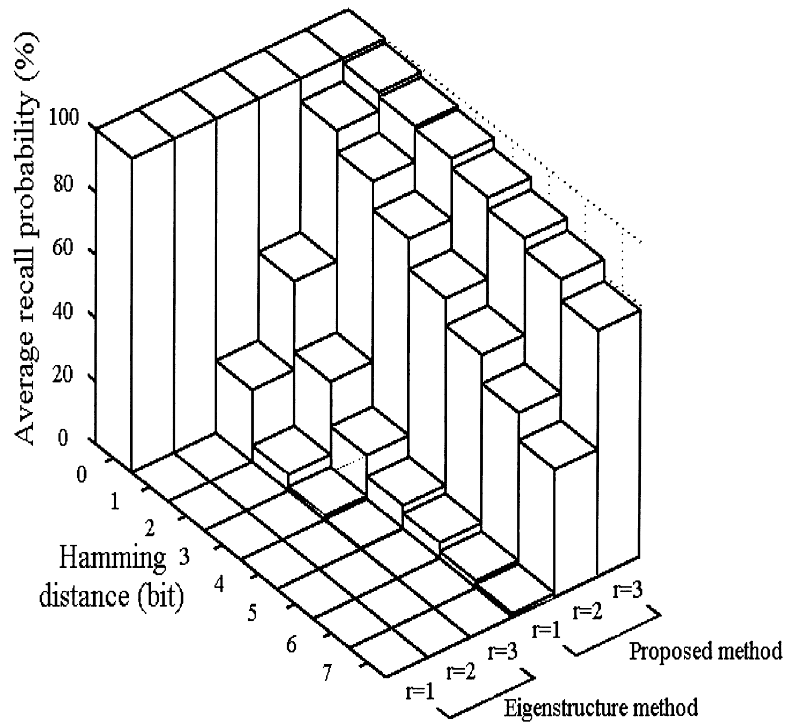

Fig. 2. Average recall probabilities of the cellular neural networks with neighborhood radius $r \in \{1, 2, 3\}$.

- find $T_{ij}, j \in Z_j$ and b_i by solving

$$\begin{cases} \min(-\kappa_i) \\ \text{s.t.}(-\kappa_i)q_i + \alpha_i^{(k)}\left(\sum_{j\in Z_i} T_{ij}\alpha_j^{(k)} + b_i\right) > 0, k = 1,\dots,m \\ -q_i < T_{ij} < q_i, j \in Z_i \\ -U < b_i < U \\ L < q_i < U \end{cases} \tag{13}$$

- else:
  - set $T_{ii} = 1 + \varepsilon$, and
  - find $T_{ij}, j \in Z_i$, and b_i such that

$$\begin{cases} \alpha_i^{(k)}\left(\sum_{j\in Z_i} T_{ij}\alpha_j^{(k)} + b_i\right) > -\epsilon, k = 1,\dots,m \\ -U < T_{ij} < U, j \in Z_i \\ -U < b_i < U. \end{cases} \tag{14}$$

Note that in the above procedure, problem (13) is a GEVP, while problems (12) and (14) are LMIPs.

4. A design example

To demonstrate the applicability of the synthesis procedure proposed in this paper, a design example is considered. In the example, we wish to store the 26 CGA (Color Graphics Adapter) fonts for the English capital letters, which are shown in Fig. 1, in the CNN memories comprised of 7×7 cells with neighborhood size r. In the figure, each prototype pattern is represented by 7×7 boxes, in which black and white represent '1' and '$-$ 1', respectively. Three neighborhood sizes, $r = 1$, 2, and 3, are considered in this example. For each $r \in \{1, 2, 3\}$ and the corresponding index matrix $\mathbf{S}$, we obtained a pair of the connection weight matrix $\mathbf{T} = \mathbf{T}|_{\mathbf{S}}$ and the bias vector $\vec{b}$ by following the proposed synthesis procedure with $L = 1$, $U = 10$, and $\varepsilon = 0.1$. The GEVPs and LMIPs arising in the procedure were solved using the functions 'gevp' and 'feasp' of the MATLAB LMI Control Toolbox, respectively. For performance evaluation of each designed CNN, we used average recall probabilities $Prob(l)$'s, which are defined as follows: Given the pair of a prototype patterns $\vec{\alpha}^{(k)}$ and an integer $l \in \{0, 1, \dots\}$, we define the recall probability $P(\vec{\alpha}^{(k)}, l)$ as the probability that the trajectory starting from a bipolar initial condition vector on the set $\delta B(\vec{\alpha}^{(k)}, l) \triangleq \{\vec{\alpha} \in B^n | HD(\vec{\alpha}, \vec{\alpha}^{(k)}) = l\}$ successfully yields $\vec{\alpha}^{(k)}$ as its final output. The average recall probability $Prob(l)$ is the average of the $P(\vec{\alpha}^{(k)}, l)$ over all the prototype patterns $\vec{\alpha}^{(1)}, \dots, \vec{\alpha}^{(k)}$ (i.e. $Prob(l) \triangleq \{\sum_{k=1}^{m} P(\vec{\alpha}^{(k)}, l)\}/m$). The estimates for the $Prob(l)$ were obtained via simulations, in which each test vector[2] in $\cup_{k\in\{1,\ldots,m\}}\delta B(\vec{\alpha}^{(k)}, l)$ was applied as an initial condition of the system, and its final output was observed after a sufficiently long time. Shown in the background of Fig. 2 are the estimates for the average recall probabilities obtained by the above method.

For comparison purposes, we also considered other synthesis methods to solve the same example. As is well-known, the constraint[3] on the diagonal elements of the connection weight matrix $\mathbf{T}$ given by $T_{ii} = 1$ for $i = 1, \ldots, n$ enables the CNN memories to have certain desirable properties with respect to the prevention of spurious patterns and the attractivity of prototype patterns. Some of the existing synthesis methods (e.g. the GEVP method of Park and Park (2000), and the sparse design algorithm of Liu and Lu (1997), with the optimal constraint) utilize this constraint. However, once the diagonal elements of $\mathbf{T}$ are fixed at unity, the problem of storing the alphabets in Fig. 1 as the memory vectors of the network becomes infeasible for $r \in \{1, 2, 3\}$; thus, these synthesis methods are not applicable in our example. On the other hand, the eigenstructure method of Liu and Michel (1993) does not depend on the optimal constraint. Following their procedure with $k = 15$ and $\mathbf{W}_{\mathrm{i}} = -7 \times [1, \ldots, 1] \times \mathbf{U}_{\mathrm{i2}}$,[4] we designed a CNN memory for each neighborhood radius $r \in \{1, 2, 3\}$. For these memories, we conducted simulations with the same test vector set, and obtained the estimates for their average recall probabilities shown in the foreground of Fig. 2. Comparing the experimental results in the figure, we can see that the procedure of this paper can be a promising choice for CNN memory synthesis.

5. Concluding remarks

In this paper, we proposed a synthesis procedure for CNN memories. The procedure, which was developed based on some known results and a newly derived theorem, consists of GEVPs and/or LMIPs. Since efficient interior point methods that can solve GEVPs/LMIPs arising in the procedure within a given tolerance are readily available, the procedure is very useful in practice. A design example was presented to illustrate the proposed method, and the CNN memories designed according to the proposed procedure showed encouraging simulation results.

References

Boyd, S., ElGhaoui, L., Feron, E., & Balakrishnan, V. (1994). *Linear matrix inequalities in systems and control theory*, Philadelphia, PA: SIAM.

Chan, H. Y., & Zak, S. H. (1998). Real-time synthesis of sparsely interconnected neural associative memories. *Neural Networks*, *11*, 749–759.

Chua, L. O., & Yang, L. (1988). Cellular neural networks: theory and applications. *IEEE Transactions on Circuits and Systems*, *35*, 1257–1290.

Gahinet, P., Nemirovski, A., Laub, A. J., & Chilali, M. (1995). MATLAB LMI Control Toolbox. Natick, MA: Math Works Inc.

Hassoun, M. H. (1993). *Associative neural memories: theory and implementation*, New York: Oxford University Press.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of National Academy of Science*, *79*, 2554–2558.

Li, J. H., Michel, A. N., & Porod, W. (1989). Analysis and synthesis of a class of neural networks: linear systems operating on a closed hypercube. *IEEE Transactions on Circuits and Systems*, *36*, 1405–1422.

Liu, D., & Michel, A. N. (1993). Cellular neural networks for associative memories. *IEEE Transactions on Circuits and Systems-II*, *40*, 119–121.

Liu, D., & Lu, Z. (1997). A new synthesis approach for feedback neural networks based on the perceptron training algorithm. *IEEE Transactions on Neural Networks*, *8*, 1468–1482.

Park, J., & Park, Y., (2000). An optimization approach to design of cellular neural networks. To appear in International Journal of Systems Science.

Perfetti, R. (1999). Dual-mode space-varying self-designing cellular neural networks for associative memory. *IEEE Transactions on Circuits and Systems-I*, *46*, 1281–1285.

Roska, T. & Vandewalle, J. (1995). *Cellular neural networks*, Chichester: John Wiley & Sons Ltd.

Seiler, G., Schuler, A. J., & Nossek, J. A. (1993). Design of robust cellular neural networks. *IEEE Transactions on Circuits and Systems-I*, *40*, 358–364.

Vandenberghe, L., & Balakrishnan, V. (1997). Algorithms and software for LMI problems in control. *IEEE Control Systems Magazine*, *17*, 89–95.

[2] When l is 0 or 1 we used all the possible bipolar vectors for the test vector set. When integer l is greater than 1, the test vector set for computing $Prob(l)$ was obtained by collecting the 100 bipolar vectors randomly chosen on $\delta B(\vec{\alpha}^{(k)},l)$ per each prototype pattern $\vec{\alpha}^{(k)}$.

[3] This constraint is called 'the optimal constraint' by Liu and Lu (1997).

[4] For exact meanings of k, $\mathbf{W}_{\mathrm{i}}$, and $\mathbf{U}_{\mathrm{i2}}$, see the CNN design procedure (D) of Liu and Michel (1993).