Brief Paper

Associative memory design using overlapping decompositions☆

Mehmet Akar[a],*, M. Erol Sezer[b]

[a]*Department of Electrical Engineering, Center for Systems Science, Yale University, P.O. Box 208267, New Haven, CT 06520-8267, USA*
[b]*Department of Electrical Engineering, Bilkent University, 06533 Bilkent, Ankara, Turkey*

Abstract

This paper discusses the use of decomposition techniques in the design of associative memories via artificial neural networks. In particular, a disjoint decomposition which allows an independent design of lower-dimensional subnetworks and an overlapping decomposition which allows subnetworks to share common parts, are analyzed. It is shown by a simple example that overlapping decompositions may help in certain cases where design by disjoint decompositions fails. With this motivation, an algorithm is provided to synthesize neural networks using the concept of overlapping decompositions. Applications of the proposed design procedure to a benchmark example from the literature and to a pattern recognition problem indicate that it may improve the effectiveness of the existing methods. © 2001 Published by Elsevier Science Ltd.

Keywords: Overlapping decompositions; Associative memory design; Neural networks

1. Introduction

Design of neural networks for the purpose of associative memories has been one of the major research topics since Hopfield proposed his model (Hopfield, 1982). With a proper choice of the connection weights, the neural network can store some desired vectors as asymptotically stable equilibria of the network. This problem is called the associative memory design problem, and has been analyzed by various researchers using both discrete-time and continuous-time neural network models (Hopfield, 1982; Personnaz, Guyon, & Dreyfus, 1985; Li, Michel, & Porod, 1989; Michel, Si, & Yen, 1991; Lillo, Miller, Hui, & Zak, 1994; Perfetti, 1995; Xu & Kwong, 1998).

The objective of this paper is to investigate the use of overlapping decompositions in the design of associative memories. The idea of obtaining a solution to a large problem by first decomposing the problem into a number of smaller subproblems, and then combining their solutions in a suitable way is well known, and has been successfully applied to analysis and design of large-scale systems (see, for example, Himmelblau, 1973; Courtois, 1977; Šiljak, 1990). The concept of overlapping decompositions is a generalization of this idea by allowing the subsystems to share some common parts, and thus providing greater flexibility in the choice of the subsystems (Ikeda & Šiljak, 1980). The theory behind the overlapping decompositions is the *Inclusion Principle* (Ikeda, Šiljak, & White, 1984), which justifies the embedding of a finite-dimensional system into a higher-dimensional system called an expansion of the original system. An overlapping decomposition of the original system corresponds to a disjoint decomposition of the expanded system. A decentralized solution for the (disjoint) pieces of the expanded system is then contracted to obtain a solution for the original system.

The concept of overlapping decompositions has been successively applied to various large-scale problems including decentralized optimal control (Ikeda & Šiljak, 1980; Ikeda, Šiljak, & White, 1981), parallel distributed compensation for fuzzy systems (Akar and Özgüner, 1999), and solution of linear equations (Calvet & Titli, 1989). The objective of this paper is to point out that the same approach can also be used in the design of associative memories; a problem concerned with the choice of

some design parameters to satisfy a set of linear inequalities, which is briefly stated in Section 2. The use of decomposition techniques in the context of associative memory problem is further discussed in Section 3, and a decomposition algorithm is suggested in Section 4 to break down the problem into smaller pieces for which available design methods can be used more effectively. Finally, the proposed design procedure is illustrated with two examples in Section 5, and concluding remarks are given in Section 6.

2. Associative memory problem

Consider a discrete-time neural network described by Hopfield's model (Hopfield, 1982) as

$$\mathscr{S}: \quad x(k+1) = F(Wx(k) + b), \quad x(0) = x_0, \tag{1}$$

where $x(k) \in \mathbb{B}^n$, with $\mathbb{B} = \{0,1\}$, is the binary state vector at instant k, $W \in \mathbb{R}^{n \times n}$ is the interconnection matrix, $b \in \mathbb{R}^n$ is the bias term, and $F(x) = [f(x_1), f(x_2), \ldots, f(x_n)]^T$ with the activation function f being a step function defined as

$$f(\sigma) = \begin{cases} 1, & \sigma \geq 0, \\ 0, & \sigma < 0. \end{cases} \tag{2}$$

A vector $y \in \mathbb{B}^n$ is an equilibrium of $\mathscr{S}$ if $x(k, y) = y$ for all $k \geq 0$, where $x(k, x_0)$ denotes the solution of (1) starting with $x(0) = x_0$. The basin of attraction of an equilibrium y is defined to be the set $\mathbb{A}_y = \{x_0 \in \mathbb{B}^n \mid x(K, x_0) = y$ for some $K \geq 0\}$.

The associative memory problem is to store a desired set of patterns $y^l \in \mathbb{B}^n$, $l = 1, 2, \ldots, m$, as equilibria of the neural network in (1). The problem corresponds to solving for W and b for the above model such that the number of spurious states (undesired equilibria) in the resulting network is minimum, and the basins of attraction of the desired patterns are as large as possible. Other requirements include the absence of limit cycles, high storage and retrieval efficiency, learning and forgetting capabilities, and a non-symmetric W to avoid difficulties in implementation.

At this stage, it is appropriate to point out that the network $\mathscr{S}$ in (1) is equivalent to a network with bipolar states $x(k) \in \{-1, 1\}^n$ and compatible signum-type nonlinearities in that one can establish a one-to-one correspondence between the equilibria of the two models with an invertible transformation of the weight and bias terms. In this sense, the model in (1) is also equivalent to the brain-state-in-a-box neural network model (Lillo et al., 1994) with $x(k) \in [-1, 1]^n$ and compatible saturation type non-linearities as far as the vertex equilibria of the latter are concerned. These observations allow for a fair comparison of the design procedure proposed in this paper with the existing approaches.

3. Disjoint and overlapping decompositions

To motivate the concept of overlapping decompositions we start with an example. Consider a stick of three pieces which are colored as black (B) or white (W). Among the $2^3 = 8$ possible colorings of the stick, let us choose the desired patterns to be stored as {BBB, BBW}. Since the desired colorings of the pieces are independent of each other, we can design a one-neuron neural network for each piece independently of the others, and then combine these individual solutions. If we add WWW to be a third pattern to be stored, then, since all three pieces can assume both colors, a disjoint decomposition as above is not possible. The best we can do is to consider the first two pieces together, and the third piece separately, in which case the desired patterns are {BB, WW} for the first subsystem, and {B, W} for the second. Then a combination of solutions results in four stored patterns {BBB, BBW, WWB, WWW}, WWB being a spurious state.

Now, suppose that we duplicate the middle piece, and view the stick as consisting of four pieces instead of three, with the restriction that the two middle pieces should be of the same color. For this expanded stick, the patterns to be stored are {BBBB, BBBW, WWWW}. A disjoint decomposition of the expanded stick into two parts consisting of the first two and the last two pieces, respectively, results in two subsystems each with two states. The desired patterns to be stored are {BB, WW} for the first subsystem, and {BB, BW, WW} for the second subsystem. Among $2 \times 3 = 6$ possible combinations of the solutions, only three satisfy the restriction concerning the middle pieces, and those are exactly the desired patterns to be stored.

The actual design can be summarized as follows: Coding black and white with 1 and 0, the desired memory matrix is written and expanded as

$$Y = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \qquad \tilde{Y} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ \cdots & \cdots & \cdots \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} \tilde{Y}_1 \\ \tilde{Y}_2 \end{bmatrix}.$$

The memory matrices $\tilde{Y}_1$ and $\tilde{Y}_2$ can be realized by two independent neural networks with

$$\tilde{W}_{11} = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}, \qquad \tilde{b}_1 = \begin{bmatrix} -0.5 \\ -0.5 \end{bmatrix},$$

$$\tilde{W}_{22} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \qquad \tilde{b}_2 = \begin{bmatrix} -0.5 \\ -1.5 \end{bmatrix}.$$

A contraction of the expanded network results in the required original network, whose parameters are

obtained as

$$W = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \qquad b = \begin{bmatrix} -0.5 \\ -0.5 \\ -1.5 \end{bmatrix}.$$

As illustrated in this example, disjoint decompositions do not help much other than trivial cases, while we may be able to solve the problem using overlapping decompositions. In the following, we will further look into this concept.

Suppose that the state vector of the neural network $\mathscr{S}$ in (1) is partitioned into three components as $x = [x_1^T\ x_2^T\ x_3^T]^T$ with $x_q \in \mathbb{B}^{n_q}$, $q = 1,2,3$, and $n = n_1 + n_2 + n_3$. This induces a decomposition of $\mathscr{S}$ in (1) into an interconnection of three subsystems as

$$\mathscr{S}_q: \quad x_q(k+1) = F_q\left(W_{qq} x_q(k) + b_q + \sum_{r=1, r \neq q}^{3} W_{qr} x_r(k) \right),$$
$$q = 1,2,3. \tag{3}$$

This is a disjoint decomposition of the system, where the state spaces of the subsystems are linearly independent. Suppose that the subsystems are decoupled, that is, $W_{qr} = 0$ for $r \neq q$. In this case, if y_q^l, $q = 1,2,3$, $l = 1,2,\ldots,m_q$, are the equilibria (stored patterns) for the qth subnetwork, then each of the $m = m_1 \times m_2 \times m_3$ possible combinations of the form $y^l = [y_1^{l_1\ T}\ y_2^{l_2\ T}\ y_3^{l_3\ T}]^T$ is a stored pattern of the original network. Conversely, if the desired memory vectors can be expressed as such, then the overall network can be obtained by synthesizing the decoupled subsystems in (3) independently of each other.

Alternatively, we may decompose the state vector into two parts $\tilde{x}_1 = [x_1^T\ x_2^T]^T$ and $\tilde{x}_2 = [x_2^T\ x_3^T]^T$, which overlap on x_2. Clearly, this is not a disjoint decomposition of the original state vector x, but of an expanded vector $\tilde{x} = [x_1^T\ x_2^T\ x_2^T\ x_3^T]^T = Vx$, where

$$V = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix},$$

with I_1, I_2, and I_3 being identity matrices of order n_1, n_2, and n_3, respectively. This transformation defines an expanded network $\tilde{\mathscr{S}}$ described by

$$\tilde{\mathscr{S}}: \quad \tilde{x}(k+1) = \tilde{F}(\tilde{W}\tilde{x}(k) + \tilde{b}), \quad \tilde{x}(0) = \tilde{x}_0, \tag{4}$$

where $\tilde{x}(k) \in \mathbb{B}^{\tilde{n}}$, $\tilde{W} \in \mathbb{R}^{\tilde{n} \times \tilde{n}}$, $\tilde{b} \in \mathbb{R}^{\tilde{n}}$, and $\tilde{F}(\tilde{x}) = [f(\tilde{x}_1), f(\tilde{x}_2), \ldots, f(\tilde{x}_{\tilde{n}})]^T \in \mathbb{R}^{\tilde{n}}$, with $\tilde{n} \geq n$. It is not difficult to show that if

$$\tilde{W}V = VW, \quad \tilde{b} = Vb, \quad \tilde{F}(Vx) = VF(x),$$

hold, then $\tilde{x}(k, Vx_0) = Vx(k, x_0)$ for all $k \geq 0$ and $x_0 \in \mathbb{B}^n$. That is, the restriction of $\tilde{\mathscr{S}}$ on the column space of V is exactly the original network $\mathscr{S}$. We call $\tilde{\mathscr{S}}$ an expansion of $\mathscr{S}$, and $\mathscr{S}$ a restriction of $\tilde{\mathscr{S}}$.

The overlapping decomposition of x corresponds to a disjoint decomposition of $\tilde{x}$, which in turn, induces a disjoint decomposition of the expanded network $\tilde{\mathscr{S}}$ into two subnetworks described as

$$\tilde{\mathscr{S}}_1: \quad \tilde{x}_1(k+1) = \tilde{F}_1(\tilde{W}_1 \tilde{x}_1(k) + \tilde{W}_{12} \tilde{x}_2(k) + \tilde{b}_1),$$

$$\tilde{\mathscr{S}}_2: \quad \tilde{x}_2(k+1) = \tilde{F}_2(\tilde{W}_2 \tilde{x}_2(k) + \tilde{W}_{21} \tilde{x}_1(k) + \tilde{b}_2),$$

where $\tilde{x}_1 = [x_1^T\ x_2^T]^T$, $\tilde{x}_2 = [x_2^T\ x_3^T]^T$,

$$\tilde{W}_1 = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}, \qquad \tilde{W}_{12} = \begin{bmatrix} 0 & W_{13} \\ 0 & W_{23} \end{bmatrix},$$

$$\tilde{b}_1 = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \tag{5}$$

$$\tilde{W}_2 = \begin{bmatrix} W_{22} & W_{23} \\ W_{32} & W_{33} \end{bmatrix}, \qquad \tilde{W}_{21} = \begin{bmatrix} W_{21} & 0 \\ W_{31} & 0 \end{bmatrix},$$

$$\tilde{b}_2 = \begin{bmatrix} b_2 \\ b_3 \end{bmatrix}, \tag{6}$$

and

$$\tilde{F}_1(\tilde{x}_1) = [f(\tilde{x}_{11}), f(\tilde{x}_{12}), \ldots, f(\tilde{x}_{1,n_1+n_2})]^T,$$

$$\tilde{F}_2(\tilde{x}_2) = [f(\tilde{x}_{21}), f(\tilde{x}_{22}), \ldots, f(\tilde{x}_{2,n_2+n_3})]^T.$$

Clearly, if $\tilde{W}_{12} = 0$ and $\tilde{W}_{21} = 0$, then the two subnetworks are decoupled and therefore can be designed independently. This, however, puts some restrictions on the structures of the $\tilde{W}_1$ and $\tilde{W}_2$ matrices of the subsystems. They have to be of the form

$$\tilde{W}_1 = \begin{bmatrix} W_{11} & W_{12} \\ 0 & W_{22} \end{bmatrix}, \qquad \tilde{W}_2 = \begin{bmatrix} W_{22} & 0 \\ W_{32} & W_{33} \end{bmatrix}. \tag{7}$$

Now if $\tilde{y}_q^l$, $q = 1,2$, $l = 1,2,\ldots,\tilde{m}_q$, are the equilibria of $\tilde{\mathscr{S}}_q$, then each of the $\tilde{m} = \tilde{m}_1 \times \tilde{m}_2$ possible combinations of the form $\tilde{y}^l = [\tilde{y}_1^{l_1\ T}\ \tilde{y}_2^{l_2\ T}]^T$ is an equilibrium of $\tilde{\mathscr{S}}$. However, among these, only the ones that satisfy $\tilde{y}^l = Vy^l$ correspond to the equilibria of $\mathscr{S}$. Moreover, the basin of attraction of each such y^l is exactly the restriction of the basin of attraction of the corresponding $\tilde{y}^l$ to the column space of V. Conversely, if the desired memory vectors can be decomposed into overlapping components which can be realized by subnetworks having the bias inputs as in (5), (6) and the weight matrices of the form in (7), then the problem can be solved by designing smaller-dimensional networks.

4. Design and decomposition algorithms

Design by overlapping decompositions explained in the previous section is summarized by the following algorithm.

4.1. Overlap-design algorithm

1. Find a transformation matrix V and expand the memory vectors as

$$\tilde{Y} = VY = \begin{bmatrix} \tilde{Y}_1 \\ \tilde{Y}_2 \end{bmatrix}. \tag{8}$$

2. Design subnetworks with $\tilde{W}_q$ as in (7) and $\tilde{b}_q$ as in (5), (6) to store the memory matrices $\tilde{Y}_q$, $q = 1,2$.
3. Compute W and b by restricting $\tilde{W}$ and $\tilde{b}$ on the column space of V.

Step 3 of the algorithm is trivial as the matrices $\tilde{W}_q$ and $\tilde{b}_q$, $q = 1,2$, contain nothing more than the required parameters of W and b.

In Step 2 of the algorithm, any design method can be used provided it takes into account the additional constraints on $\tilde{W}_q$ and $\tilde{b}_q$, $q = 1,2$. Concerning this step, we would like to emphasize that the whole idea behind the overlap-design algorithm is not to suggest a new design method for the subnetworks, but rather to apply the existing methods to smaller (overlapping) pieces of the whole network.

The key step of the algorithm is Step 1. The block diagonal structure of $\tilde{W}_q$ reveals that each subnetwork $\tilde{\mathcal{S}}_q$ should be designed in a hierarchical manner, first the part common to both subnetworks (corresponding to x_2) and then the remaining parts (corresponding to x_1 for $\tilde{\mathcal{S}}_1$ and x_3 for $\tilde{\mathcal{S}}_2$). In other words, not only the design of the overlapped part is independent of the design of the rest, but also the designs of the remaining parts of the subnetworks depend only on the overlapped part and are independent of each other. Therefore, identification of a common part that allows for such a hierarchical design is what makes an overlapping decomposition useful. Another reason for the decomposition to be the most critical step of the algorithm is the fact that the advantage of overlapping decomposition heavily depends on how large the overlapping block is. Sometimes the decomposition is inherent in the physical problem itself as we will be demonstrating in the pattern recognition example in Section 5. Most of the times it is not, in which case we want to decompose the equilibria set such that the overlapping block size is as small as possible. Below we provide an algorithm to decompose the rows of a memory matrix into two groups overlapping on a single row which allows for a hierarchical design.

4.2. Decomposition algorithm

1. Fix one of the rows of Y and index the other rows from 1 to $n - 1$.
2. Let $\Psi = \emptyset$, $\Omega = \{1,2,3,\ldots,n-1\}$. Ψ and Ω will contain the indices of the rows forming the two equilibria sets overlapping on the fixed row.

3. Initialize $i = 1$, and min $= 2^n$.
4. while ($i < n - 1$) and (min $> m$)
   4.1. Initialize $j = 1$.
   4.2. while ($j < |\Omega|$) and (min $> m$)
       4.2.1. $r =$ row of Y corresponding to jth element of Ω.
       4.2.2. $\mathscr{A} = \Psi \cup \{r\}$, $\mathscr{B} = \Psi - \{r\}$.
       4.2.3. $M =$ total equilibrium points corresponding to groups $\mathscr{A}$ and $\mathscr{B}$.
       4.2.4. If $M <$ min, then min $= M$ and index $= j$.
       4.2.5. $j = j + 1$.
   4.3. $\Psi = \Psi \cup \{\text{index}\}$, $\Omega = \Omega - \{\text{index}\}$.
   4.4. $i = i + 1$.

Since the decomposition algorithm provided above checks only for two equilibria sets with only one overlapping row, it fails when such a decomposition does not exist. In this case, either we take the optimal solution, or we revise the algorithm for the search of two equilibria sets with two or more overlapping rows. To modify the algorithm for two rows overlapping, for instance, we proceed as follows: We fix a pair of rows (among $n(n - 1)/2$ possible) to be overlapped, and then search for two sets Ψ and Ω ($\Psi = \emptyset$, $\Omega = \{1,2,\ldots,n-2\}$, initially in Step 2 of the algorithm) so that these sets combined with the two overlapped rows will form the decoupled equilibria sets in the expanded space. Decomposing the equilibria set into two sets with three or more rows overlapping can be carried out similarly. When one wants to decouple the equilibria set into more than two equilibria sets, this can be done by first finding an overlapping pair of equilibria sets using the Decomposition Algorithm, and then employing the algorithm iteratively within each set independently to come up with three or more overlapping equilibria sets.

5. Examples

In this section, two examples are presented to illustrate the effectiveness of the overlap-design algorithm. In the first example, neural network parameters are synthesized using both the existing methods and the overlap-design algorithm, and the performance of the resulting neural networks are compared. The second example is a pattern recognition application for which design with the existing methods is not possible.

Example 1. This example is taken from Li et al. (1989). The desired memory vectors are the rows of the matrix

$$Y^{\mathrm{T}} = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

Applying the decomposition algorithm to Y, we find out that there is no equivalent decomposition with only one row overlapped. However, if we apply the algorithm with two rows overlapped, we see that various decompositions exist. Among them, let us take the groups (1,3,4,5,7,8) and (2,3,4,6,9,10). Then the expansion matrix becomes

$$V = [e_1^T \ e_3^T \ e_4^T \ e_5^T \ e_7^T \ e_8^T \ e_2^T \ e_3^T \ e_4^T \ e_6^T \ e_9^T \ e_{10}^T]^T,$$

where e_i is a 1 by n row vector with 1 in the ith entry and zeros elsewhere. Transforming the equilibria matrix Y by V, the problem is reduced to solving for two subnetworks of dimension 6 which have equilibria as the columns of the matrices

$$\tilde{Y}_1 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \qquad \tilde{Y}_2 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}.$$

Note that while the number of desired patterns to be stored by the original network is five, each of the subnetworks are required to store only four patterns. This is due to the fact that parts of different patterns corresponding to the subnetworks may be the same.

Straightforward design for $\tilde{Y}_1$ and $\tilde{Y}_2$ subject to the structural constraints on the weight and bias matrices yields globally stable subnetworks with no spurious states. Since each subnetwork stores exactly four equilibria, the decoupled neural network will store $4 \times 4 = 16$ equilibria in the expanded state space. However, among these 16 possible equilibrium points, only five of them correspond to actual equilibria in the original space, and the remaining eleven degenerate to one of these five. Not surprisingly, these five equilibria are exactly the desired equilibria in the original set, since the decomposition algorithm has provided an exact overlapping decomposition. After contraction, the neural network with the parameters,

results in a globally stable network with no spurious states.

In Table 1, we summarize the performances of various neural networks designed using different methods, where OPM refers to the outer product method of Hopfield (1982), PLR to the projection learning rule of Personnaz et al (1985), and ESM to the eigenstructure method of Li et al. (1989). In the table, M is the total number of stable equilibria, and $\mu(y,k)$ denotes the total number of vectors converging to the equilibrium y and with Hamming distance to y being less than or equal to k. Therefore, $\sum_{l=1}^{5} \mu(y^l,1)$ denotes the total number of vectors within unit Hamming distance converging to the desired memory vectors. These parameters are a measure of the basin of attraction of the desired memory vectors. It is clear from this table that the design by overlapping decompositions outperforms the other methods since it leads to a network with no spurious states. Relatively larger basins of attraction of equilibria as compared with other designs is an additional gain, although that is not the major objective of the overlap-design algorithm.

Example 2. This example is an application of the proposed design method to a pattern recognition problem. The objective is to realize the character set given in Fig. 1 as asymptotically stable equilibria of a 25-neuron neural network. Decomposing the 5×5 grid on which the characters are defined into four 3×3 grids overlapping on the middle row and column of pixels, the problem is decomposed to designing four neural networks of nine neurons each. This numbering and decomposition process is shown in Fig. 2. Note that this is a natural decomposition, and is more general than one that can be produced by the decomposition algorithm as pixel pairs $\{3,8\}$, $\{11,12\}$, $\{14,15\}$ and $\{18,23\}$ are each shared by two subnetworks, and pixel $\{13\}$ by all four.

Independent designs, after coding the black and white pixels by 1s and 0s, yield globally stable subnetworks with 33, 62, 35 and 38 equilibria, respectively. By contraction, we obtain a neural network with 6769 equilibria. The number of patterns converging to each desired prototype is listed in Table 2.

We note that synthesis of a neural network for this character set by using known methods is not possible.

$$W = \begin{bmatrix} 1 & 0 & 0 & -2 & -1 & 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1.33 & -0.67 & 2 & 0 & -1 & 0.67 & 0 & 0 \\ 0 & -1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 2 & -1 & 1 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & -4 & 3 & 1 & 0 & -2 & -3 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} -0.5 \\ -0.5 \\ -0.5 \\ -0.5 \\ 0.5 \\ 0.5 \\ 0.5 \\ 1.5 \\ -0.5 \\ -0.5 \end{bmatrix},$$

Table 1
Results for different methods in Example 1

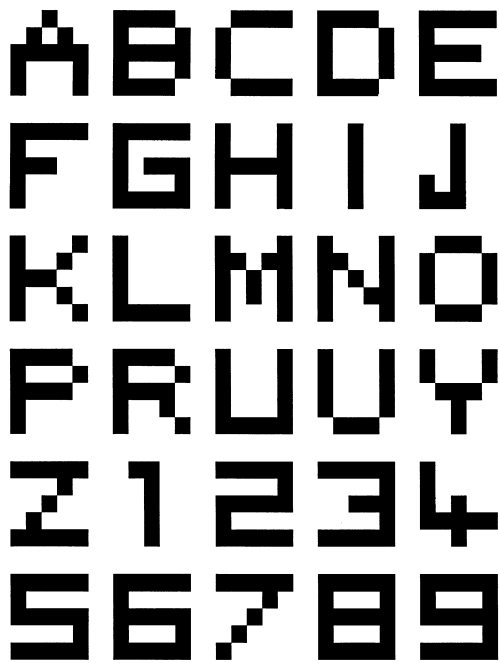|	OPM	PLR	ESM	Lillo et al.	Perfetti	Overlap-design
M	41	122	20	8	10	5
$\mu(y^1,10)$	38	24	84	143	112	240
$\mu(y^2,10)$	41	28	75	62	58	128
$\mu(y^3,10)$	73	34	113	16	82	512
$\mu(y^4,10)$	38	24	84	29	133	16
$\mu(y^5,10)$	68	33	105	56	127	128
$\sum_{l=1}^{5} \mu(y^l,1)$	33	30	40	9	46	35

Fig. 1. Character set to be recognized by the neural network.
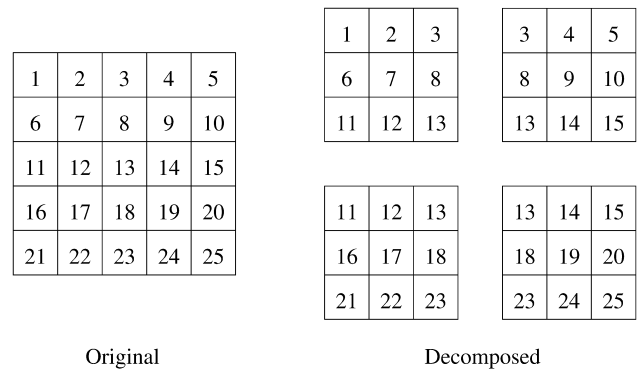

Fig. 2. Original and expanded systems.

The outer product rule will be able to store only three of the patterns, the projection learning rule, eigenstructure method and the method of Lillo et al. will store all the 2^{25} patterns. Perfetti's method is not even applicable, since there are prototype patterns which are at unit Hamming distance from each other (e.g., 5 & 6, 8 & 9).

6. Conclusion

In this paper, the use of overlapping decompositions in associative memory design via artificial neural networks has been investigated. The proposed method consists of decomposing the design problem into subproblems that interact through the overlapping components, finding a solution to each subproblem independently (that is, synthesizing subnetworks that solve the subproblems), and combining the subnetworks in a suitable way to obtain a solution for the overall problem. The increase in dimensionality of the problem due to duplication of the overlapped parts, and the additional structural constraints on the weight matrices of the subnetworks are

Table 2
Total number of vectors converging to prototype characters

Pattern (π)	A	B	C	D	E	F	G	H
$\mu(\pi,25)$	10,240	864	288	6016	120	384	768	122880
Pattern (π)	I	J	P	R	U	V	Y	K
$\mu(\pi,25)$	1344	2688	7680	368	5120	56832	39376	768
Pattern (π)	L	M	N	O	Z	1	2	3
$\mu(\pi,25)$	1920	79680	651392	6720	128	5376	24	576
Pattern (π)	4	5	6	7	8	9	—	—
$\mu(\pi,25)$	56	432	144	512	216	648	—	—

offset by the advantage of dealing with smaller subproblems independently of each other. The particular method used in the design of subnetworks is not important as long as it can handle the structural constraints on the weight matrices, although some methods may be more suitable than others.

Several topics are open to further research: The first is to improve the decomposition algorithm so that it can generate a decomposition of a given set of desired equilibria into more than two components overlapping on more than one row. Although finding the optimal decomposition into a fixed number of components seems to be an NP problem, some heuristics can be used to achieve a suboptimal decomposition with reasonable effort. A second problem is to find ways to relax the requirement that the subnetworks be decoupled in the expanded space. This not only relaxes the constraints on the weight matrices, but also nonzero off-diagonal blocks of the weight matrices can be used to eliminate some of the spurious states of the expanded network. A final topic of investigation is associative memory design via neural networks with multilevel threshold functions.

References

Akar, M., & Özgüner, Ü. (1999). Decentralized parallel distributed compensator design for Takagi–Sugeno fuzzy systems. *Proceedings of IEEE conference on decision and control*, Phoenix, AZ (pp. 4834–4839).

Calvet, J. L., & Titli, A. (1989). Overlapping vs. partitioning in block-iteration methods: Application in large-scale system theory. *Automatica*, 25, 137–145.

Courtois, P. J. (1977). *Decomposability*. New York: Academic Press.

Himmelblau, D. M. (1973). *Decomposition of large scale problems*. New York: Elsevier.

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

Ikeda, M., & Šiljak, D. D. (1980). Overlapping decompositions, expansions and contractions of dynamic systems. *Large Scale Systems*, 1, 29–38.

Ikeda, M., Šiljak, D. D., & White, D. E. (1981). Decentralized control with overlapping information sets. *Journal of Optimization Theory and Applications*, 34, 279–310.

Ikeda, M., Šiljak, D. D., & White, D. E. (1984). An inclusion principle for dynamic systems. *IEEE Transactions on Automatic Control*, 29, 244–249.

Li, J. H., Michel, A. N., & Porod, W. (1989). Analysis and synthesis of a class of neural networks: Variable structure systems with infinite gain. *IEEE Transactions on Circuits and Systems*, 36, 713–731.

Lillo, W. E., Miller, D. C., Hui, S., & Zak, S. H. (1994). Synthesis of Brain-State-in-a-Box (BSB) based associative memories. *IEEE Transactions on Neural Networks*, 5(5), 730–737.

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

Perfetti, R. (1995). A synthesis procedure for Brain-State-in-a-Box neural networks. *IEEE Transactions on Neural Networks*, 6(5), 1071–1080.

Personnaz, L., Guyon, I., & Dreyfus, G. (1985). Information storage and retrieval in spin-glass like neural networks. *Journal de Physique Letters*, 46, 359–365.

Šiljak, D. D. (1990). *Decentralized control of complex systems*. New York: Academic Press.

Xu, Z. -B., & Kwong, C. -P. (1998). Associative memories. In C. T. Leondes (Ed.), *Neural network systems techniques and applications* (pp. 183–258). New York: Academic Press.

Mehmet Akar was born in Gulnar, Turkey in 1973. He received his B.S. and M.S. degrees from Bilkent University, Ankara in 1994 and 1996, respectively, and his Ph.D. degree from the Ohio State University, Columbus, OH, in 1999, all in electrical engineering. Currently, he is a post-doctoral research associate at Yale University, New Haven, CT. His research interests include stability and control of hybrid systems and resource allocation in wireless systems.
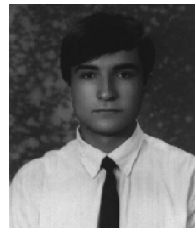
M. Erol Sezer received his Ph.D. from the Middle East Technical University, Turkey, in 1978. After working at METU, Santa Clara University (CA), University of Petroleum and Minerals (S. Arabia), and METU again, in 1986 he joined the Bilkent University, where he is currently a professor. His research interests include multivariable and large-scale control systems, robot dynamics and control, and sampled-data systems. Dr. Sezer is a member of the IEEE Control Systems Society and Associate Editor of "Automatica" and "Computers and Electrical Engineering".