2004 International Symposium on Nonlinear
Theory and its Applications (NOLTA2004)
Fukuoka, Japan, Nov. 29 - Dec. 3, 2004

A Design Method of Cellular Neural Networks for Associative Memories based on Linear Programming

Tsuyoshi Yamakawa, Norikazu Takahashi and Tetsuo Nishi

Department of Computer Science and Communication Engineering, Kyushu University
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581 Japan
Email: yamakawa@kairo.csce.kyushu-u.ac.jp, {norikazu,nishi}@csce.kyushu-u.ac.jp

Abstract—A novel design method of cellular neural networks (CNNs) for associative memories is presented. The proposed method can realize as high recall probability as the method based on generalized eigenvalue minimization (GEVM), which is known as the most efficient CNN design method so far. On the other hand, since the proposed method is based on linear programming, its computation time is much shorter than the GEVM-based method.

1. Introduction

Realization of associative memories is one of the fundamental design problems for cellular neural networks (CNNs) [1, 2]. Recently a method of designing CNNs for associative memories based on generalized eigenvalue minimization (GEVM) has been proposed [3, 4]. This is regarded as one of the most effective CNN design methods because every prototype vector is stored as a memory vector and the basin of attraction of each prototype vector is maximized in a certain sense.

In this paper, a novel design method of CNNs based on linear programming (LP) will be presented. Since the proposed method relies on the basic idea of the GEVM-based method, the average recall probability [3] is kept at the same level as the GEVM-based method. On the other hand, computation time of the proposed method is much shorter than the GEVM-based method since LP problems can be solved more easily than GEVM problems in general.

In what follows, we first introduce a couple of theorems concerning the basin of attraction of a memory vector derived by Bise *et al.* [4] and the basic strategy used in the GEVM-based method. We next show that the strategy can be formulated as LP problems. We then give our CNN design procedure and show its efficiency by computer simulations.

2. Problem Formulation

Let us consider CNNs described by the following differential equations:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = -x_i + \sum_{j\in\bar{N}_i} A_{ij}y_j + I_i, \quad i = 1,2,\ldots,n \tag{1}$$

where x_i is the state of the i-th cell, y_i the output of the i-th cell determined by x_i through

$$y_i = f(x_i) \triangleq \frac{1}{2}(|x_i+1| - |x_i-1|), \tag{2}$$

A_{ij} the coupling coefficient from the j-th cell to the i-th cell, I_i the bias of the i-th cell, and $\bar{N}_i \subseteq \{1,2,\ldots,n\}$ the set of indices of the cells belonging to the neighborhood of the i-th cell. In the following, the neighborhood of the i-th cell excluding itself is represented by N_i, that is, $N_i \triangleq \bar{N}_i \setminus \{i\}$. Although it is often assumed in CNN literature that coupling coefficients between cells are space-invariant [5], we will not make this assumption in this paper.

A vector $\boldsymbol{y}^e = [y_1^e, y_2^e, \ldots, y_n^e]^T$ is referred to as a memory vector of a CNN described by (1) and (2) if the CNN has an asymptotically stable equilibrium point $\boldsymbol{x}^e = [x_1^e, x_2^e, \ldots, x_n^e]^T$ such that $y_i^e = f(x_i^e)$ for $i = 1,2,\ldots,n$. The set of initial states $\boldsymbol{x}(0)$ such that $\lim_{t\to\infty}\boldsymbol{x}(t) = \boldsymbol{x}^e$ is called the basin of attraction of the memory vector $\boldsymbol{y}^e$.

CNN design problem can be stated as follows: For given prototype vectors $\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \ldots, \boldsymbol{\alpha}^m \in \{1,-1\}^n$ and the sets $\bar{N}_1, \bar{N}_2, \ldots, \bar{N}_n$, find A_{ij}, $j \in \bar{N}_i$ and I_i for $i = 1,2,\ldots,n$ such that the synthesized CNN has the following properties: 1) all prototype vectors $\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \ldots, \boldsymbol{\alpha}^m$ are memory vectors, 2) the total number of spurious memory vectors, that is, the memory vectors of the CNN not contained in $\{\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \ldots, \boldsymbol{\alpha}^m\}$, is as small as possible, 3) the basin of attraction of each prototype vector is as large as possible, and 4) the CNN has no oscillatory solution. As well as the GEVM-based method [3, 4], we will focus our attention only on the first three properties in this paper.

3. Analysis

We introduce some analytical results derived by Bise *et al.* [4] concerning the basin of attraction of a memory vector which play important roles in the GEVM-based method.

Theorem 1 *Suppose a set $\bar{N}_i \subseteq \{1,2,\ldots,n\}$ and a binary vector $\boldsymbol{\alpha}^* = [\alpha_1^*, \alpha_2^*, \ldots, \alpha_n^*]^T \in \{1,-1\}^n$ are given. If the coupling coefficients A_{ij}, $j \in N_i$ and the bias I_i satisfy*

$$\alpha_i^*\left(\sum_{j\in N_i} A_{ij}\alpha_j^* + I_i\right) > \kappa_i \max_{j\in N_i}|A_{ij}| + (A_{ii}-1) \tag{3}$$

with $A_{ii} \geq 1$ and $\kappa_i \geq 0$, then any vector $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_n]^T \in \mathbb{R}^n$ such that $f(\beta_i) \neq \alpha_i^*$ and $\sum_{j\in N_i}|f(\beta_j) - \alpha_j^*| \leq \kappa_i$ has the following properties.

1. *$\boldsymbol{f}(\boldsymbol{\beta}) = [f(\beta_1), \ldots, f(\beta_n)]^T$ is not a memory vector.*

2. *If $\boldsymbol{x}(0) = \boldsymbol{\beta}$ then $x_i(t)$ moves toward α_i^* at $t = 0$.*

Theorem 2 *Suppose a set $\bar{N}_i \subseteq \{1, 2, \ldots, n\}$ and a binary vector $\boldsymbol{\alpha}^* = [\alpha_1^*, \alpha_2^*, \ldots, \alpha_n^*]^T \in \{1, -1\}^n$ are given. If the coupling coefficients A_{ij}, $j \in \bar{N}_i$ and the bias I_i satisfy*

$$\alpha_i^* \left(\sum_{j \in \bar{N}_i} A_{ij} \alpha_j^* + I_i \right) > \kappa_i \max_{j \in \bar{N}_i} |A_{ij}| - (A_{ii} - 1) \qquad (4)$$

with $\kappa_i \geq 0$, then any vector $\boldsymbol{\beta} \in \mathbb{R}^n$ such that $f(\beta_i) \neq \alpha_i^*$ and $\sum_{j \in \bar{N}_i} |f(\beta_j) - \alpha_j^*| \leq \kappa_i$ has the followings properties.

1. *$\boldsymbol{f}(\boldsymbol{\beta}) = [f(\beta_1), \ldots, f(\beta_n)]^T$ is not a memory vector.*

2. *If $\boldsymbol{x}(0) = \boldsymbol{\beta}$ then $x_i(t)$ moves toward α_i^* at $t = 0$.*

4. Design Method

4.1. Basic Idea

The most important thing in the CNN design problem is to store m prototype vectors $\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \ldots, \boldsymbol{\alpha}^m$ as memory vectors. This is achieved by choosing the coupling coefficients A_{ij}, $j \in \bar{N}_i$ and the bias I_i such that the set of inequalities

$$\alpha_i^k \left(\sum_{j \in \bar{N}_i} A_{ij} \alpha_j^k + I_i \right) > 1, \quad k = 1, 2, \ldots, m \qquad (5)$$

holds for $i = 1, 2, \ldots, n$.

The basic strategy of the CNN design method proposed by Bise *et al.* is as follows: For $i = 1, 2, \ldots, n$, first examine whether (5) is feasible with $A_{ii} = 1$ or not. If it is feasible then we set $A_{ii} = 1$ and determine the values of A_{ij}, $j \in N_i$ and I_i by solving the following optimization problem.

Problem 1 *Find A_{ij}, $j \in N_i$ and I_i which maximize κ_i under the constraints*

$$\alpha_i^k \left(\sum_{j \in N_i} A_{ij} \alpha_j^k + I_i \right) > \kappa_i \max_{j \in N_i} |A_{ij}|, \quad k = 1, 2, \ldots, m. \qquad (6)$$

If (5) is not feasible with $A_{ii} = 1$, we set $A_{ii} = 1 + \epsilon$ where ϵ is a positive constant, and determine the values of A_{ij}, $j \in N_i$ and I_i by solving the following optimization problem.

Problem 2 *Find A_{ij}, $j \in N_i$ and I_i which maximize κ_i under the constraints*

$$\alpha_i^k \left(\sum_{j \in N_i} A_{ij} \alpha_j^k + I_i \right) > \kappa_i \max\{1 + \epsilon, \max_{j \in N_i} |A_{ij}|\} - \epsilon, \quad k = 1, 2, \ldots, m, \qquad (7)$$

This method tries to make the basin of attraction of prototype vectors as large as possible by making use of Theorems 1 and 2 while guaranteeing (5). The total number of spurious memory vectors is expected to be reduced if the basins of attraction become large. Also, only binary vectors can become memory vectors as $A_{ii} \geq 1$, $i = 1, 2, \ldots, n$ are always satisfied.

4.2. Transformation to Linear Programming Problem

Park *et al.* have shown that Problem 1 can be transformed into a GEVM problem [3]. Based on their idea, Bise *et al.* have shown that Problem 2 can also be transformed into a GEVM problem [4]. In this section, we will show that both Problems 1 and 2 are intrinsically equivalent to LP problems.

First we assume as a special case that

$$\alpha_i^1 = \alpha_i^2 = \cdots = \alpha_i^m \qquad (8)$$

holds. In this case, it is apparent that Eq.(5) is feasible with $A_{ii} = 1$. In addition, if we set A_{ij} to 0 for all $j \in N_i$ and I_i to some positive (negative, resp.) value if $\alpha_i^k = 1$ ($\alpha_i^k = -1$, resp.) for all k then Eq.(6) becomes $|I_i| > \kappa_i \cdot 0$. Hence there is no upper bound for the value of κ_i. This means that

$$y_i(\infty) = \begin{cases} +1, & \text{if } \alpha_i^k = +1, \forall k \\ -1, & \text{if } \alpha_i^k = -1, \forall k \end{cases}$$

holds for any initial state $\boldsymbol{x}(0)$. Consequently, in the case where Eq.(8) holds, we do not have to solve Problem 1. In other words, it suffices for us to set the values of parameters as $A_{ij} = 0$, $\forall j \in N_i$ and $I_i = \mathrm{sgn}(\alpha_i^1)$.

In the following, we assume that there exists at least one pair (k_1, k_2) such that $\alpha_i^{k_1} \alpha_i^{k_2} = -1$. Let us now consider the following optimization problem.

Problem 3 *Find A_{ij}, $j \in N_i$ and I_i which maximize κ_i under the constraints*

$$\begin{aligned} \alpha_i^k \left(\sum_{j \in N_i} A_{ij} \alpha_j^k + I_i \right) &> \kappa_i, \quad k = 1, 2, \ldots, m \\ |A_{ij}| &\leq 1, \quad \forall j \in N_i \end{aligned}$$

For Problem 3, the following theorem holds.

Theorem 3 *Suppose that (5) is feasible with $A_{ii} = 1$. Then any optimal solution of Problem 3 is also an optimal solution of Problem 1.*

Proof: Let $(A_{ij_1}^*, A_{ij_2}^*, \ldots, A_{ij_{|N_i|}}^*, I_i^*)$ be any optimal solution of Problem 1 where $|N_i|$ denotes the cardinality of the set N_i. Then $(cA_{ij_1}^*, cA_{ij_2}^*, \ldots, cA_{ij_{|N_i|}}^*, cI_i^*)$ where c is any positive number, is also an optimal solution of Problem 1. This means Problem 1 always has an optimal solution satisfying $\max_{j \in N_i} |A_{ij}| = 1$. Therefore any optimal solution of Problem 4 given below is an optimal solution of Problem 1.

Problem 4 *Find A_{ij}, $j \in N_i$ and I_i which maximize κ_i under the constraints*

$$\begin{aligned} \alpha_i^k \left(\sum_{j \in N_i} A_{ij} \alpha_j^k + I_i \right) &> \kappa_i, \quad k = 1, 2, \ldots, m \\ \max_{j \in N_i} |A_{ij}| &= 1 \end{aligned}$$

Next we will show that any optimal solution $(\tilde{A}_{ij_1}, \tilde{A}_{ij_2}, \ldots, \tilde{A}_{ij_{|N_i|}}, \tilde{I}_i)$ of Problem 3 satisfies

$$\max_{j \in N_i} \left|\tilde{A}_{ij}\right| = 1. \tag{9}$$

If this is the case, we can easily see that any optimal solution of Problem 3 is an optimal solution of Problem 4 and vice versa. Let us assume Eq.(9) does not hold. Then there exists a sufficiently small positive number δ such that

$$\alpha_i^k \left(\sum_{j \in N_i} (1+\delta)\,\tilde{A}_{ij}\alpha_j^k + (1+\delta)\,\tilde{I}_i \right) > \alpha_i^k \left(\sum_{j \in N_i} \tilde{A}_{ij}\alpha_j^k + \tilde{I}_j \right), \quad k = 1, 2, \ldots, m \tag{10}$$

$$\left|(1+\delta)\,\tilde{A}_{ij}\right| \le 1, \quad \forall j \in N_i$$

which means κ_i can be increased if we set $A_{ij} = (1+\delta)\,\tilde{A}_{ij}$, $j \in N_i$ and $I_i = (1+\delta)\,\tilde{I}_i$. However, this contradicts the assumption that $(\tilde{A}_{ij_1}, \tilde{A}_{ij_2}, \ldots, \tilde{A}_{ij_{|N_i|}}, \tilde{I}_i)$ is an optimal solution. Hence Eq.(9) holds true.

From the above discussions on the relationship among optimal solutions of Problems 1, 3 and 4, we can conclude that any optimal solution of Problem 3 is also an optimal solution of Problem 1. ■

Let us next consider the following optimization problem.

Problem 5 *Find A_{ij}, $j \in N_i$ and I_i which maximize κ_i under the constraints*

$$\alpha_i^k \left(\sum_{j \in N_i} A_{ij}\alpha_j^k + I_i \right) > (1+\epsilon)\,\kappa_i - \epsilon, \quad k = 1, 2, \ldots, m$$

$$|A_{ij}| \le 1 + \epsilon, \quad \forall j \in N_i$$

Theorem 4 *Any optimal solution of Problem 5 is an optimal solution of Problem 2.*

Proof: Let $(A^*_{ij_1}, A^*_{ij_2}, \ldots, A^*_{ij_{|N_i|}}, I^*_i)$ be any optimal solution of Problem 2. Then the upper bound for κ_i in Problem 2 is given by the following equation:

$$\frac{\alpha_i^k \left(\sum_{j \in N_i} A^*_{ij}\alpha_j^k + I^*_i \right) + \epsilon}{\max\left\{1+\epsilon, \max_{j \in N_i} |A^*_{ij}|\right\}} \tag{11}$$

We will show in the following that any optimal solution $(A^*_{ij_1}, A^*_{ij_2}, \ldots, A^*_{ij_{|N_i|}}, I^*_i)$ of Problem 2 satisfies

$$\max_{j \in N_i} |A^*_{ij}| \le 1 + \epsilon. \tag{12}$$

Suppose (12) does not hold. Then there exists a sufficiently small positive number δ such that $\max_{j \in N_i} |(1-\delta)\,A^*_{ij}| > 1 + \epsilon$. We therefore have

$$\begin{aligned} &\frac{\alpha_i^k \left(\sum_{j \in N_i} (1-\delta)\,A^*_{ij}\alpha_j^k + (1-\delta)\,I^*_i \right) + \epsilon}{\max\left\{1+\epsilon, \max_{j \in N_i} (1-\delta)\,|A^*_{ij}|\right\}} \\ &= \frac{(1-\delta)\,\alpha_i^k \left(\sum_{j \in N_i} A^*_{ij}\alpha_j^k + I^*_i \right) + \epsilon}{(1-\delta)\,\max\left\{1+\epsilon, \max_{j \in N_i} |A^*_{ij}|\right\}} \\ &> \frac{\alpha_i^k \left(\sum_{j \in N_i} A^*_{ij}\alpha_j^k + I^*_i \right) + \epsilon}{\max\left\{1+\epsilon, \max_{j \in N_i} |A^*_{ij}|\right\}}, \quad k = 1, 2, \ldots, m \end{aligned}$$

which means that the upper bound for κ_i can be increased by setting $A_{ij} = (1-\delta)\,A^*_{ij}$, $j \in N_i$ and $I_i = (1-\delta)\,I^*_i$. However, this contradicts the assumption that $(A^*_{ij_1}, A^*_{ij_2}, \ldots, A^*_{ij_{|N_i|}}, I^*_i)$ is an optimal solution of Problem 2. Hence Eq.(12) holds. Since Problem 5 is derived by adding the constraint $|A_{ij}| \le 1+\epsilon, \forall j \in N_i$ to Problem 2, we can state from the above discussion that any optimal solution of Problem 2 is an optimal solution of Problem 5. Also, it is obvious that the opposite is also true. ■

4.3. CNN Design Procedure

Based on the analysis in the preceding subsection, we propose the following CNN design procedure.

CNN Design Procedure: Given n sets $\overline{N}_1, \overline{N}_2, \ldots, \overline{N}_n \subseteq \{1, 2, \ldots, n\}$ and m prototype vectors $\boldsymbol{\alpha}^1, \boldsymbol{\alpha}^2, \ldots, \boldsymbol{\alpha}^m \in \mathbb{B}^n$, execute the following procedure for $i = 1, 2, \ldots, n$.

1) Check whether the set of inequalities (5) is feasible with $A_{ii} = 1$. If it is feasible go to Step 2), otherwise go to Step 3).

2) Set $A_{ii} = 1$, and find A_{ij}, $j \in N_i$ and I_i which maximize κ_i under the constraints

$$\alpha_i^k \left(\sum_{j \in N_i} A_{ij}\alpha_j^k + I_i \right) \ge \kappa_i, \quad k = 1, 2, \ldots, m$$

$$|A_{ij}| \le 1, \quad \forall j \in N_i$$

3) Set $A_{ii} = 1 + \epsilon$ where ϵ is a positive number and find A_{ij}, $j \in N_i$ and I_i which maximize κ_i under the constraints

$$\alpha_i^k \left(\sum_{j \in N_i} A_{ij}\alpha_j^k + I_i \right) \ge (1+\epsilon)\,\kappa_i - \epsilon, \quad k = 1, 2, \ldots, m$$

$$|A_{ij}| \le 1 + \epsilon, \quad \forall j \in N_i$$

Since optimization problems in Step 2) and Step 3) are LP problems, they can be solved efficiently by using the simplex method or interior point methods. It is thus expected that the proposed method can find the parameter values much faster than the GEVM-based method proposed by Bise *et al.* [4]. It is also expected that the basin of attraction of each prototype vector for the proposed method is as large as that for the GEVM-based method because both methods rely on the same basic idea described in Sec.4.1.

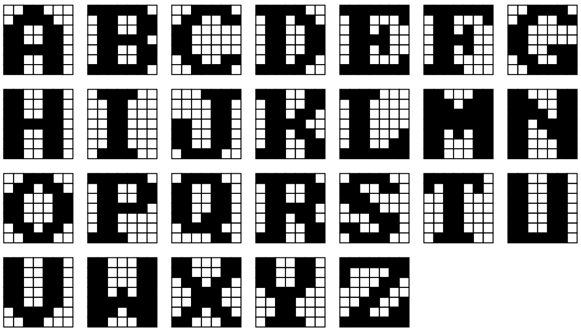

Figure 1: Prototype vectors

It should be noted here that the LP problem in Step 3) has a trivial solution $A_{ij} = 0, \forall j \in N_i$ and $I_i = 0$ because Step 3) is carried out only if Eq.(5) is not feasible. So one may claim that there is no need to solve the LP problem. However, as shown in the next section, the trivial solution lead to lower recall probability than nontrivial solutions.

5. Computer Simulation

In order to verify efficiency of the proposed method, we have applied the proposed method and the GEVM-based method proposed by Bise *et al.* [4] to the same set of prototype vectors shown in Fig.1, and investigated the computation time and the average recall probability [3] of the synthesized CNNs for both methods.

Table 1 shows the CPU time required for finding the parameter values with the GEVM-based method and the proposed method for some values of r, radius of neighborhood for each cell [5]. Both methods are implemented in MATLAB and executed on a PC with 1.2GHz Pentium III processor and 256MB RAM. As one can see, CPU time is considerably reduced by using the proposed method for each value of r. Moreover, as far as our simulation results are concerned, CPU time for the proposed method is kept almost constant, while it increases rapidly with r for the GEVM-based method.

Table 1: CPU time

r	Method	Time (s)
1	GEVM	27.143
	Proposed	6.123
2	GEVM	75.112
	Proposed	5.881
3	GEVM	197.250
	Proposed	6.597

Table 2 shows the average recall probabilities for the GEVM-based method, the proposed method, and the proposed method with trivial solution which is a special case of the proposed method where the parameters are set as $A_{ij} = 0, \forall j \in N_i$ and $I_i = 0$ in Step 3). As one can see from Table 2, the average recall probability for the proposed method is kept at the same level as the GEVM-based method in all cases, while for the proposed method with trivial solution it is lower than other two methods. This means it is important to find a nontrivial solution in Step 3) of the proposed CNN design procedure.

Table 2: Comparison of Average Recall Probability

r	Method	Average recall probability $P_{\mathrm{av}}(d)$				
		$d = 1$	$d = 2$	$d = 3$	$d = 4$	$d = 5$
1	GEVM	0.5479	0.2969	0.1623	0.0879	0.0455
	Proposed	0.5345	0.2894	0.1699	0.0906	0.0542
	Proposed/Triv.	0.3328	0.1017	0.0326	0.0114	0.0028
2	GEVM	0.9584	0.8898	0.8046	0.7080	0.6026
	Proposed	0.9427	0.8776	0.7895	0.6958	0.6009
	Proposed/Triv.	0.8831	0.7633	0.6606	0.5419	0.4409
3	GEVM	0.9937	0.9766	0.9585	0.9298	0.8891
	Proposed	0.9922	0.9691	0.9518	0.9213	0.8872
	Proposed/Triv.	0.9733	0.9367	0.9055	0.8624	0.8169

6. Conclusion

We have proposed a CNN design method for associative memories based on LP. The proposed method can reduce significantly the computation time compared to the GEVM-based method while keeping the average recall probability at the same level. In this sense, the proposed method is superior to the GEVM-based method.

Acknowledgement

This research was partly supported by Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Encouragement of Young Scientists, 15760268 and the 21st Century COE Program 'Reconstruction of Social Infrastructure Related to Information Science and Electrical Engineering'.

References

[1] D. Liu and A.N. Michel, "Sparsely interconnected neural networks for associative memories with applications to cellular neural networks," *IEEE Trans. Circuits Syst. II*, vol.41, pp.295–307, Apr. 1994.

[2] G. Grassi, "A new approach to design cellular neural networks for associative memories," *IEEE Trans. Circuits Syst. I*, vol.44, no.8, pp.835–838, 1997.

[3] J. Park, H.-Y. Kim, Y. Park and S.-W. Lee, "A synthesis procedure for associative memories based on space-varying cellular neural networks," *Neural Networks*, vol.14, no.1, pp.107–113, Jan. 2001.

[4] R. Bise, N. Takahashi and T. Nishi, "An Improvement of the Design Method of Cellular Neural Networks Based on Generalized Eigenvalue Minimization," *IEEE Trans. Circuits Syst.-I*, vol.50, no.12, pp.1569-1574, Dec., 2003.

[5] L.O. Chua and L. Yang, "Cellular neural networks: theory," *IEEE Trans. Circuits Syst. I*, vol.35, pp.1257–1272, Oct. 1988.