A TWO LAYER CNN IN IMAGE PROCESSING APPLICATIONS

Zonghuang Yang, Yoshifumi Nishio and Akio Ushida

Dept. of Electrical and Electronic Engineering, Tokushima University, Tokushima 770-8506, Japan
yangzh@ee.tokushima-u.ac.jp

Abstract — In this paper, a two-layer cellular neural network model is proposed, in which two new templates coupling between the two layers are introduced. This CNN array behaves more efficient for image process applications. As examples, several simulations in image processing applications such as object separation and center detection etc. are carried out, and their templates are given. In particular, their working mechanisms are described in detail.

I. INTRODUCTION

Since cellular neural networks (CNNs) were invented by Chua and Yang in 1988 [1], they have been successfully used for various high-speed parallel signals processing applications such as image processing, pattern recognition as well as computer vision [1]-[5]. Under certain conditions, the working mechanism of this structure arrays acts as the way to model physical structure or nonlinear phenomena: because its locally interconnected structure closely resembles the feature of human retina and active substrate or medium [2][6]. It has been used in modeling pattern formations as well as many other spectacular dynamic phenomena [6][7].

Up to the present, owing to the efforts of all researchers that have been done in this area, many templates for special purposes and applications have been developed. But many image processing tasks, such as skeletonizing, thinning and separating object etc., can only be completed by iteratively using different templates or time variant templates [3]-[5]. All of them are based on complex multi-layer, sometimes even non-uniform architectures. Here, it is necessary to notice the fact that this multi-layer CNN means that it consists of several single layer CNNs, each single layer CNN is iteratively used to perform a part of task, and only after it entries into steady state or reaches at some states, the next single layer CNN begins to perform the next part of task, such as the process continues iteratively until the novel task is completed. Therefore, they are tedious and belong to serial processing, result in decreasing run-speed. On the other hand, as it has been well known, in the traditional neural networks, if the multi-layer structure neural networks are taken into consideration, they have displayed more widely applications in very different fields such as non-linearly separation, optimization and pattern recognition etc. This mechanism is also valid to cellular neural networks. In order to improve its applications, a cellular neural network model with two layers structure is proposed in this paper. Several simulations for image processing applications are given and described in detail. All of them show that the two-layer CNN has more efficient for image processing.

II. TWO LAYER CNN ARCHITECTURE

The dynamical system equations associated with single layer CNNs have been given in reference [1]. Here, we extend them to two-layer CNNs by introducing two new templates C_{12} and C_{21}. For image processing purpose, an architecture composed of a two-dimensional M by N array of cells are only taken into consideration. Each cell in the array is denoted by $c(i,j)$. It has two state variables $x_1(i,j)$ and $x_2(i,j)$, where (i,j) stands for the position of cell in the array, $1 \le i \le M$ and $1 \le j \le N$. The state equations of each cell are given by two first-order differential equations Eq.(1), and the state output equations are given by Eq.(2), which is a piecewise-linear nonlinear function.

$$\left\{\begin{array}{rcl} \frac{dx_{1,ij}}{dt} & = & -x_{1,ij} + I_1 \\ & + & \sum_{C(k,l)\in N_r(i,j)} A_1(i,j;k,l)y_{1,kl} \\ & + & \sum_{C(k,l)\in N_r(i,j)} B_1(i,j;k,l)u_{1,kl} \\ & + & \sum_{C(k,l)\in N_r(i,j)} C_{12}(i,j;k,l)y_{2,kl} \\ \frac{dx_{2,ij}}{dt} & = & -x_{2,ij} + I_2 \\ & + & \sum_{C(k,l)\in N_r(i,j)} A_2(i,j;k,l)y_{2,kl} \\ & + & \sum_{C(k,l)\in N_r(i,j)} B_2(i,j;k,l)u_{2,kl} \\ & + & \sum_{C(k,l)\in N_r(i,j)} C_{21}(i,j;k,l)y_{1,kl} \end{array}\right. \qquad (1)$$

$$\left\{\begin{array}{l} y_{1,ij} = 0.5(|x_1+1| - |x_1-1|) \\ y_{2,ij} = 0.5(|x_2+1| - |x_2-1|). \end{array}\right. \qquad (2)$$

We define that the first layer of the two layer CNN is composed of all of $x_1(i,j)$, and the second layer of all of $x_2(i,j)$. u, x , and y refer to the input, state, and output variables of the cell, $A(i,j;k,l)$, $B(i,j;k,l)$, and I are feedback template, control template and bias current, index 1 and 2 stand for first layer and second layer of two-layer CNN array, respectively. $C_{12}(i,j;k,l)$ is used to represent the transferring template from second layer output to first layer input, and $C_{21}(i,j;k,l)$

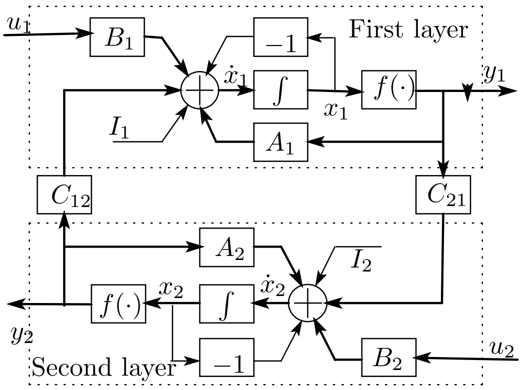


Figure 1: Block diagram of the two layer CNN

is the reverse. They also contain the weight of the local coupling between cells in a neighborhood $N_r(i,j)$ as the same as templates $A(i,j;k,l)$ and $B(i,j;k,l)$, where r is coupling radius. In generally, the r takes 1 or 2.

The processing of the two layer CNN is shown by block diagram in Fig.1. As we can see, the two layers of the CNN constitute a closed-loop system. They share out the tasks and cooperate with each other to perform some image processing applications that cannot be performed in single layer CNN. Comparing this CNN with the multi-layer CNN proposed in some literatures [1][3]-[5], one of their differences is that the former is parallel processing (i.e. the two layers execute at the same time), the later is serial processing. Moreover, when one of the templates C_{12} and C_{21} is zero, the CNN becomes an open-loop system and seems to be consisted of two single layer CNNs with cascade, one layer output is considered as another layer input. From this point, the developed templates in single layer CNNs seem to be valid to the two layer CNNs, but some compound tasks can be achieved.

III. SOME TEMPLATES USED FOR IMAGE PROCESSING APPLICATIONS

Image processing is one of important applications of the CNN. As well known, many templates used for this purpose have been developed based on signal layer CNN. In this section, we will give some examples to show that the two layer CNNs have more efficient for image processing. Their working mechanism will be described in detail.

.1. Object Separation

In this section, we use the two-layer CNN to solve a compound task – object separation. For example, referring to Fig.2(d), there are three objects, round, rectangle and triangle, respectively. If we want to separate round and triangle objects from the rectangle object, then this task can be divided into two operations, one is extracting the target objects by using pointers in the initial state in one layer, and the other one is erasing the extracted objects from the original image at the same time in another layer. This task can be realized in single step and described as the following.

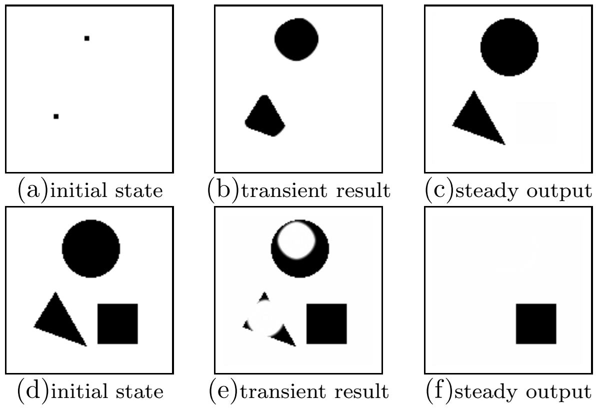


Figure 2: An example for solving compound task.

The original static binary image Fig.2(d) is fed to both layer inputs of the CNN. The first layer is used to extract objects under separation by using two pointers in the initial state Fig.2(a) with recall templates [5]. Its output is transferred to second layer as its input and operated with second layer initial state with logic difference by using logic difference templates [15]. Thus, the extracted objects are obtained in the first layer output, and the rest rectangle object remains in the second layer output. They are shown as Fig.2(c) and (f). Figure 2(b) and (e) are the first and second layer transient results, respectively. In this example, we choose zero-fixed boundary condition. The templates are

$$
\begin{aligned}
A_1 &= \begin{bmatrix} 0.5 & 0.5 & 0.5 \\ 0.5 & 2 & 0.5 \\ 0.5 & 0.5 & 0.5 \end{bmatrix}, B_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}, C_{12}=0, \\
A_2 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B_2 = 0, C_{21} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \\
I_1 &= 0.25, I_2 = -1. \qquad (3)
\end{aligned}
$$

.2. Centerline Extraction

In single layer CNN, the algorithm of the centerline extraction uses 2 steps or 2 single layer CNNs. The two CNNs are used continuously and circularly to peel off pixels from the object in two directions until only the centerline of object remains. This is a very complicated process. In contrast, by using the two layer CNNs, this task can be solved simply by one step. The algorithm will be described in detail in the following.

The two layers of the CNN are used respectively to continuously peel off the most left side and right side pixels of the object at same time until its centerline is

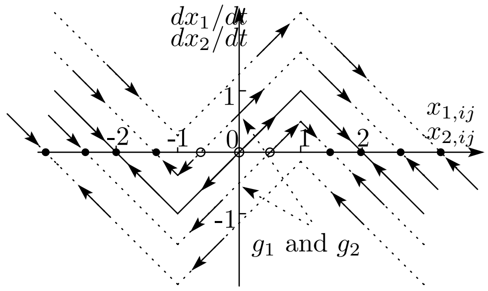


Figure 3: The dynamic routes of the cell are shifted up and down by g_1 and g_2. The stable and unstable equilibrium points are denoted by solid dot and circle, respectively.

left. For this purpose, we feed the object image to the initial states of the CNN both layers as shown Fig.6(a) and (g), and adopt the following templates:

$$A_1=\begin{bmatrix} 0 & 0 & 0 \\ 0.5 & 2 & -0.5 \\ 0 & 0 & 0 \end{bmatrix}, C_{12}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2.8 & -1 \\ 0 & 0 & 0 \end{bmatrix}, \begin{matrix} B_1=0, \\ I_1=-2, \end{matrix}$$
$$A_2=\begin{bmatrix} 0 & 0 & 0 \\ -0.5 & 2 & 0.5 \\ 0 & 0 & 0 \end{bmatrix}, C_{21}=\begin{bmatrix} 0 & 0 & 0 \\ -1 & 2.8 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{matrix} B_2=0, \\ I_2=-2, \end{matrix} \tag{4}$$

Thus, the cell state equations can be rewritten as the following forms:

$$\begin{cases} \dot{x}_{1;ij} = -f_1(x_{1;i,j}) + g_{1;i,j} \\ \dot{x}_{2;ij} = -f_2(x_{2;i,j}) + g_{2;i,j} \end{cases} \tag{5}$$

where

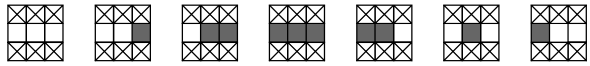
$$\begin{aligned} f_1(x_{1;i,j}) &= x_{1;i,j} - (|x_{1;i,j}+1| - |x_{1;i,j}-1|), \\ f_2(x_{2;i,j}) &= x_{2;i,j} - (|x_{2;i,j}+1| - |x_{2;i,j}-1|), \\ g_{1;i,j} &= 0.5y_{1;i,j-1} - 0.5y_{1;i,j+1} + 2.8y_{2;i,j} - y_{2;i.j+1} - 2, \\ g_{2;i,j} &= 0.5y_{2;i,j+1} - 0.5y_{2;i,j-1} - y_{1;i,j-1} + 2.8y_{1;i,j} - 2 \end{aligned} \tag{6}$$

Their dynamic routes can be demonstrated in Fig.3. To analyze the stable equilibrium point of a cell, here we specify a landscape or a local map of the cell by listing all those neighborhood states including the cell itself. Each cell has two states x_1 and x_2. In the 3×3 pattern matrix, the landscape has $2^9 \times 2^9$ possible combinations in binary image. In this simulation, at beginning the two states are the same, so the landscape can be divided into seven possible cases. All of them are listed in Fig.4, in which a black square denotes a cell having values 1, a white square denotes the -1 value cells, and a crossed square stands for those which 'don't care' with the center cell because of some of the template coefficients corresponding those positions being zero.

For the case of Fig.4(a), we have

$$\begin{aligned} g_{1;i,j} &= 0.5y_{1;i,j-1} - 0.5y_{1;i,j+1} + 2.8y_{2;i,j} - y_{2;i.j+1} - 2 \\ &= 0.5\times(-1) + 0.5\times(-1) + 2.8\times(-1) - (-1) - 2 \\ &= -3.8. \end{aligned}$$

From its dynamic route Fig.3, we can deduce that the state x_1 of the cell has a stable equilibrium within $x_1 < -1$ and outputs -1 (white), in other words, $y_1 = -1$. Similarly, for the other cases from Fig.4(b) to (g), their stable equilibrium points can be obtained and listed in Tab.1. As we can see, only in the case (c), the output of the center cell state x_1 is changed from 1 (black) to -1 (white) and steadied at -1 (white). In the other cases, they hold unchanged. Therefore, the first layer of the CNN performs peeling off the most left side pixels of the object as shown in Fig.6(b). By similar analysis method, we can obtain that the second layer perform peering off the most right side pixels of the object at the same time as shown in Fig.6(h). After the most left side pixels in first layer and the most right side pixels in second layer are peeling off from the object, the middle results are shown as Fig.6(c) and (i), respectively. As the first layer output is transferred to the second layer and second layer output to first layer, with the same analysis method, we can derive that the first layer begin to peer off the most right side pixels of the object as shown Fig.6(d), and the second layer to peel off the most left side pixels Fig.6(j). Figures 6.(e) and (k) show another transient results. So far the two layer CNN has completed a cycle. Such as this cycle continues until the object centerline is remained, while this two layer CNN enters this steady state. This dynamic process can be observed from the graph of the sum of state errors varying with the integration iterates Fig.7. As we can see, with appearing a peak, it means that a column pixel of the object is peeled off. After the object is peeled into centerline, the sum of state errors decay to zero, the system enters



(a) (b) (c) (d) (e) (f) (g)

Figure 4: All possible landscapes of a cell.

Table 1: `The Landscape and Stable Epuilibrium Point of a cell in the First Layer:`

Landscape	$g_{1,ij}$	initial state	stable output
case (a)	-3.8	-1	-1
case (b)	-6.8	-1	-1
case (c)	-1.2	1	-1
case (d)	-0.2	1	1
case (e)	2.8	1	1
case (f)	1.8	1	1
case (g)	-2.8	-1	-1

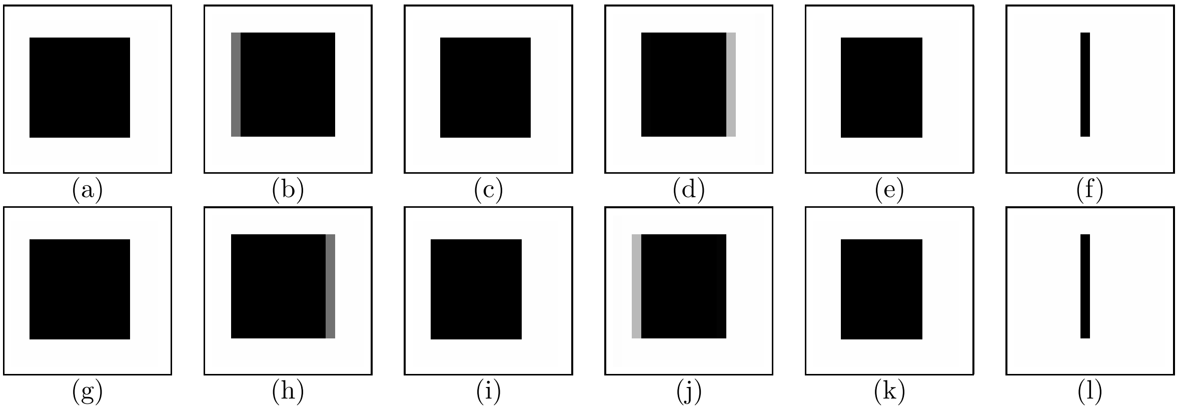


Figure 5: An example for extracting the centerline of object. From(a) to (f) are the output of the first layer with the time progressing, and from (g) to (l) are the output of second layer. Finally, the two layer CNN steadies at (f) and (l) states.

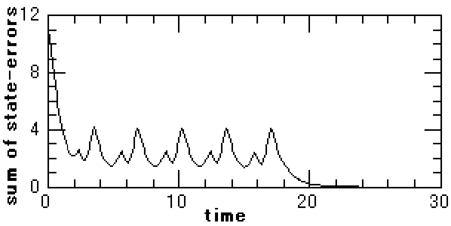


Figure 6: The sum of state errors via time for centerline extraction

steady state. The two layer stable outputs are shown as Fig.6(f) and (l), respectively. In this simulation, the zero-fixed boundary condition is adopted.

From the above two examples, we conclude that the two layer CNN can perform some compound tasks which are need to be performed with two or more steps by using single layer CNNs, and the developed templates in single layer CNN are also valid to the two layer CNNs. Furthermore, the two layers of the CNN can share out the tasks and cooperate with each other. They belong to parallel processing.

IV. CONCLUSIONS

In this paper, the proposed a two layer cellular neural network model in this paper has been introduced to complete some image processing applications such as object separation and centerline extraction. Thereby, we confirm the powerfulness of the two-layer CNN structure for parallel information processing. In particular, it is obvious that a multi-layer CNN structure is very suitable to complete those important and complex image processing applications such as the skeletonizing and thinning of object, which have to be carried out by repeatedly and circularly using the single layer CNN to perform. In spite of the complexity increasing with the CNN layers increasing, it makes us believe that a wide development of their applications will come alone with the possibility for dealing with the multi-layer CNN equations.

References

[1] L.O. Chua and L. Yang, "Cellular neural networks: theory and applications," *IEEE Trans. Circuits & Syst.*, vol.35 pp.1257–1290, Oct. 1988.

[2] L.O. Chua and T. Roska, "The CNN paradigm," *IEEE Trans. Circuits & Syst.*, vol.40, pp.147–156, Mar. 1993.

[3] P.L. Ventianer, F. werblin, T. Roska and L.O. Chua, "Analogic CNN algorithms for some image compression and restoratin tasks," *IEEE Trans. Circuits & Syst.*, vol.42, pp.278–284, Mar. 1995.

[4] H. Harrer and J. A. Nossek, "Skeletonization: A new application for discrete-time cellular neural networks using time-variant templates," *in proc. IEEE Int. Symp. Circuits & Syst.*, 1992, pp.2897-2900.

[5] T. Roska, L. Kek, L. Nemes, A. Zarandy, and P. Szolgay, "Analogical and neural computing laboratory," *Computer and Automation Institute of the Hungarian Academy of Sciences*, Version 7.3, 1999.

[6] L.O. Chua, Martin Hasler, George S. Mochytz, and Jacques Neirynck, "Autonomous cellular neural networks: a unified paradigm for pattern formation and active wave propagation," *IEEE Trans. Circuits & Syst.*, vol.42 pp.559-577, 1995.

[7] M.J. Ogorzalek, Z. Galias, A.M. Dabrowski, and W.R. Dabrowki, "Chaotic waves and spatiotemporal patterns in large arrays of doubly-coupled chua's circuits," *IEEE Trans. Circuits & Syst.*, vol.42 pp.706-714, 1995.