Templates and Algorithms for Two-Layer Cellular Neural Networks

Zonghuang Yang, Yoshifumi Nishio and Akio Ushida
Dept. of Electrical and Electronic Engineering,Tokushima University,
Minami-Josanjima 2-1, Tokushima 770-8506, JAPAN
yangzh@ee.tokushima-u.ac.jp

Abstract - This paper presents the two-layer cellular neural Networks for some important image processing applications, in which two new templates are introduced to couple between the two layers. Several simulations such as linear non-separable task, center point detection and skeletonizing, are executed with the two-layer CNN and their templates are given. All of them display that the two-layer CNNs behave more efficient for image processing compared with single-layer CNNs. In addition, the stability of the two-layer CNN with symmetric templates and/or special coupling templates is also discussed based on the Lyapunov function technique. Its equilibrium points are found from the trajectories in a phase plane. These results agree with those from simulations.

I. INTRODUCTION

Owing to the efforts of reseachers that have been done in Cellular Neural Networks (CNNs) [1], many kinds of templates have been already proposed for special purposes for single-layer CNNs. Many image processings, such as center point detection, skeletonizing, etc., can be carried out by the iterative use of different time-variant templates [2]-[4], where each single-layer CNN is iteratively used to perform a part of task. Namely, after the operation of CNN has attained to the steady state or reached at some state, the next single-layer CNN begins to perform the next part of the task. In this way, the process iteratively continues until the whole task is completed. It is obvious that this procedure is tedious and belongs to serial processing. On the other hand, it is well known in neural networks that if a multi-layer structure neural network is taken into consideration, it will have more widely applications in the different fields such as nonlinear separation, optimization and pattern recognition, etc. This mechanism will be also valid to cellular neural networks. In order to improve its applications, a cellular neural network model with two-layers structure is introduced to solve some complicated image processing problems in this paper. Throught our experimental examples, we observe that the two-layer CNN structure behave more efficient for image processing. Finally, the stability of the two-layer CNN with symmetric and/or special templates is also discussed based on the Lyapunov function technique; the equilibrium points are demonstrated by the trajectries in the phase plane.

II. STRUCTURE OF TWO-LAYER CNN

The two-layer CNNs used in this paper is composed of a two dimensional M by N array of cells. Each cell in the array is denoted by $c(i,j)$, and has two state variables $x_1(i,j)$, $x_2(i,j)$, where (i,j) stands for the position of a cell in the array, for $1 \le i \le M$ and $1 \le j \le N$.

Now, we are going to formulate the system equations of the two-layer CNNs by introducing two new counpling templates C_{12} and C_{21} between two layers. The state equations of each cell are given by two first-order differential equations Eq.(1), and the output equations are given by Eq.(2), where $f(\cdot)$ is a piecewise-linear nonlinear function defined by Eq.(3).

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

$$\left.\begin{array}{l}
y_{1,ij} = f(x_{1,ij}) \\
y_{2,ij} = f(x_{2,ij})
\end{array}\right\} \quad (2)$$

$$f(x) = 0.5(|x+1| - |x-1|). \quad (3)$$

We define state variables of the first layer by $x_1(i,j)$, and those of the second layer by $x_2(i,j)$. u and y refer to the input and output variables of the cell. $A(i,j;k,l)$, $B(i,j;k,l)$, and I are the feedback template, filter template and bias current, respectively. The index 1 and 2 stand for first layer and second layer of the two-layer CNN array. $C_{12}(i,j;k,l)$ is the coupling template to

0-7803-7278-6/02/$10.00 ©2002 IEEE

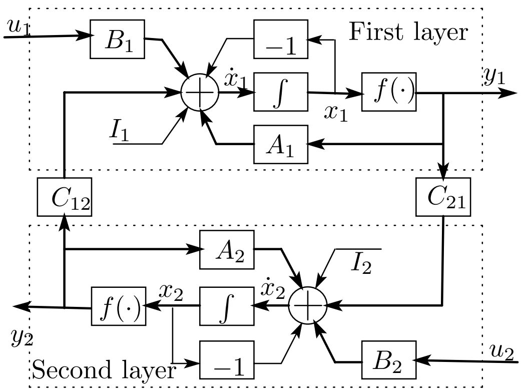

Fig. 1. Block diagram of the two-layer CNN

transfer the second layer output to the first layer input, and $C_{21}(i,j;k,l)$ is vice versa. The two new templates also show the weight of the local coupling among the cells in the neighborhood $N_r(i,j)$ the same as $A(i,j;k,l)$ and $B(i,j;k,l)$. r is the coupling radius. In generally, r takes the value of 1 or 2.

The block diagram of two-layer CNN is shown by Fig.1. When both two C templates are zero, the CNN array becomes obviously two independent single-layer CNNs. When at least one of the templates C_{12} and C_{21} is not zero, the CNN becomes an open-loop system with behaving as the cascade connections of two single-layer CNNs or constitutes a closed-loop system. The two layers of the CNN share out the tasks of some image processings and cooperate with each other, which cannot be performed with single-layer CNN. Note that CNNs proposed in some literatures [2]-[4] are serial image processings with the single-layer CNN, and on the contrary, our two-layer CNN is parallel image processings. Therefore, compared to the single-layer CNNs, the two-layer CNNs can more efficiently solve some problems in image processings, which can not be easily obtained by single-layer CNNs.

III. TEMPLATES AND ALGORITHMS FOR IMAGE PROCESSING

Image processing is one of the most important applications of the CNNs. In this section, we will give several important and interested examples of the two-layer CNNs which show efficiently for image processing compared to the single-layer CNNs.

A. Linearly Non-Separable Problems

Now, let us consider a two-layer uncoupled CNN. For simplification, we omit the subscript (i,j), because each cell in CNNs is coupled in the same way. Thus, the cell equations are written as follows;

$$\left.\begin{array}{rcl} \dot{x}_1 & = & -x_1 + a_{1,00}y_1 + b_{1,00}u_1 + c_{12,00}y_2 + I_1 \\ & = & -x_1 + a_{1,00}y_1 + g_1 \\ \dot{x}_2 & = & -x_2 + a_{2,00}y_2 + b_{2,00}u_2 + c_{21,00}y_1 + I_2 \\ & = & -x_2 + a_{2,00}y_2 + g_2 \end{array}\right\} \quad (4)$$

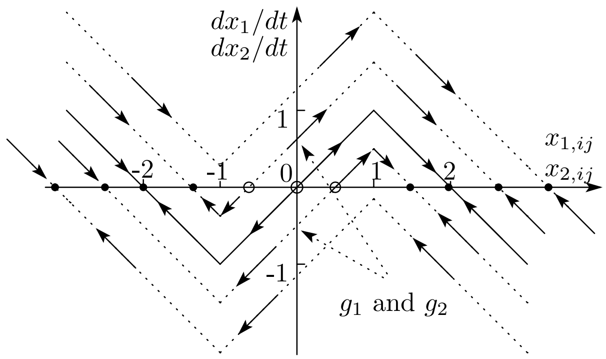

Fig. 2. The trajectories of state variables for various g_1 and g_2, where we set $a_{1,00} = a_{2,00} = 2$ and the stable and unstable equilibrium points are denoted by the solid dots and circles, respectively.

For the self-feedback coefficients $a_{1,00} > 1$ and $a_{2,00} > 1$, the trajectory of the state variables without g_1 and g_2 behave as shown by the solid line in Fig.2. By shifting g_1 and g_2, we have the trajectries shown by the dotted lines in the figure. Observe that if we choose the initial conditions $x_1(0)$ and $x_2(0)$ larger than the unstable equilibrium points, the steady-state output will be given by 1, and otherwise, the outputs given by -1. Thus, we have the following output relations depending the initial conditions;

$$\left\{\begin{array}{rcl} y_1 = & & \mathrm{sgn}[x_1(0) + \frac{g_1}{a_{1,00}-1}] \\ = & & \mathrm{sgn}[x_1(0) + \frac{b_{1,00}u_1 + c_{12,00}y_2 + I_1}{a_{1,00}-1}] \\ y_2 = & & \mathrm{sgn}[x_2(0) + \frac{g_2}{a_{2,00}-1}] \\ = & & \mathrm{sgn}[x_2(0) + \frac{b_{2,00}u_2 + c_{21,00}y_1 + I_2}{a_{2,00}-1}] \end{array}\right. \quad (5)$$

where $b_{1,00}$, $b_{2,00}$, $c_{12,00}$, $c_{21,00}$, I_1 and I_2 are the constant parameters. u_1 and u_2 are the inputs of cells, which are usually equal to 1 or -1 corresponding to black or white pixel in the binary image, or are between 1 and -1 for the grey scale image. Therefore, by suitably selecting these parameters and initial state conditions, we can obtain the expected outputs.

For example, consider the two-layer CNN to perform a linearly non-separable task – logic XOR, which cannot be directly solved by the single-layer CNNs. The logic function can be written as

$$F = X_1 \oplus X_2 \quad (6)$$

0-7803-7278-6/02/$10.00 ©2002 IEEE

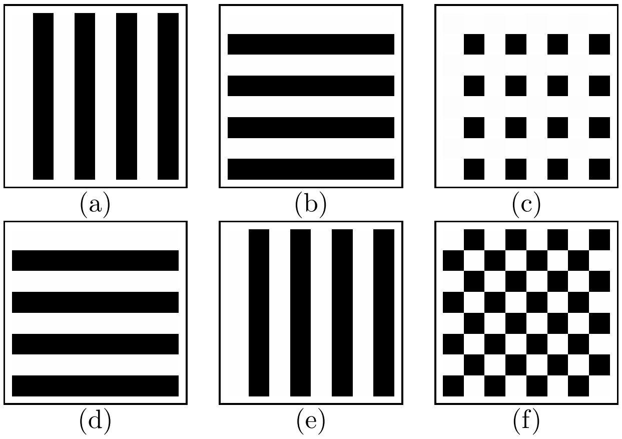

Fig. 3. An example for solving a non-linearly separable problem. (a), (b) and (c) are the input, the initial state condition and the output of first layer, (d), (e) and (f) are the input, the initial state condition and the output of second layer, respectively.

The input of two pixels has four possible combinations. The output will be black only if one of the two pixels is black. The templates are:

$$\begin{aligned} &A_1 = A_2 = 2, B_1 = B_2 = 1, \\ &C_{21} = -2, C_{12} = 0, I_1 = I_2 = -1. \end{aligned} \tag{7}$$

One of the two given images is set to the input u_1 of first layer CNN and the initial state $x_2(0)$ of the second layer. The another image is set the input u_2 of second layer and the state $x_1(0)$ of first layer. The execution result can be obtained from the output of second layer. They are shown by Fig.3, where (a) and (d) are two input images of the two layers, and (b) and (e) are the initial state. The execution result is shown by (f), where the black pixels are obtained only when one of the two inputs has a black pixel, and white pixels are obtained for every other cases. The output of first layer shows the result obtained by logic AND of the two images. These results are calculated by the relations (5). From the above simulation result, we have shown that the uncoupled two-layer CNNs are capable to perform non-linearly separable tasks.

In fact, the above example is executed based on the two layers of the CNN with cascade. Moreover, in this case, we can see that not only the developed templates and stability conditions in single-layer CNNs are valid to the two-layer CNNs, but also some compound tasks, which consist of logic operations, edge detection, etc., can be directly solved in one step. When the two C templates are not zero, the two-layer CNN constitutes a closed-loop case, every layer output is fed into the other layer input. It is because we use the mutual feed characteristics between the two layers of the CNN, the two-layer CNNs can execute some complicated tasks.

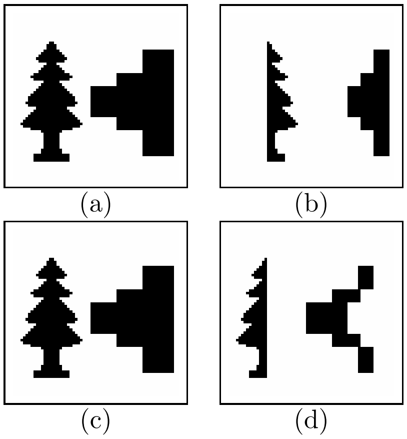

Fig. 4. An example for dividing object into half-and-half. (a) the input of first layer, (b) the output of first layer, (c) the input of second layer, (d) the output of second layer.

B. Divide Object into Half-and-half

Now let us consider a problem to divide object into half-and-half or two parts with the same area. This task can be executed by the two-layer CNN in one step. The basic idea of the algorithm is that, the two layers of the CNN are respectively and simultaneously used to peel pixels from two sides of the object, when the process reaches the centerline, the two-layer CNN arrives at the steady state, thus the object is divided into two parts which are respectively output in the CNN two layers. The templates are given as follows;

$$\begin{aligned} &A_1 = \begin{bmatrix} 1 & 2 & -1 \end{bmatrix}, B_1 = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}, I_1 = -2 \\ &A_2 = \begin{bmatrix} -1 & 2 & 1 \end{bmatrix}, B_2 = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}, I_2 = -2 \\ &C_{12} = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}, C_{21} = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}; \end{aligned} \tag{8}$$

In this case, the given binary image is set to all the inputs and initial states of the CNN as shown in Fig.4(a) and (c). Thus, we obtained the two different parts with the same area from the given object on the two layer outputs of the CNN, which are shown in Fig.4(b) and (d).

C. Center Point Detection

It is very important to detect the center point of a given object. Unfortunately, the definition of the center point is very ambiguous, because there are various objects such as convex, concave, disk with embeded holes, etc. In this paper, we define “the center point is located at the halfway from the furthermost points of a given object”. Therefore, Without the loss of generality, we can consider that the object image is a rectangle block such as

0-7803-7278-6/02/$10.00 ©2002 IEEE

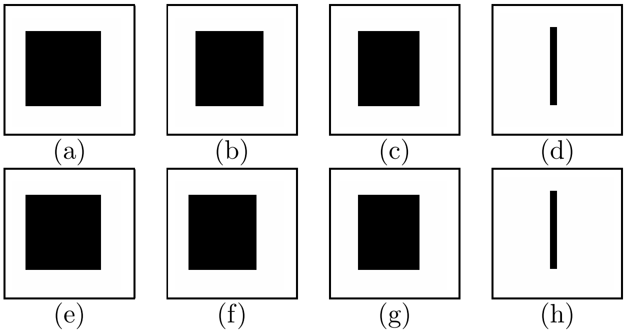

Fig. 5. An example for extracting the center line of an object. States from (a) to (d) are the output of the first layer in the time progress, and those from (e) to (h) are the output of second layer. Final results of the two-layer CNN are shown by (d) and (h).

Fig.5(a), because arbitrary shape objects can be changed into rectangle blocks with the same length and width by using the shadow template [5]. If a single-layer CNN is applied to this problem, the algorithm is based on 8 steps or 8 single-layer CNNs [5]. The eight single-layer CNNs are circularly and continuously used to peel off pixels from the object in eight directions until only the center point of the object is remained. This is a very complicated process. In contrast, by using the two-layer CNNs, this task can be solved in two steps, the first step is to detect the centerline of the object, and the second step is to detect the center point of the above centerline.

C.1 Detection of the Center Line of an Object

Now, consider a center line detection problem shown in Fig.5(a). The two layers of the CNN are respectively used to continuously peel off the most left side and/or right side pixels of the object at the same time, until its centerline is left. For this purpose, we set the object image to the initial states of the both CNN layers as shown Fig.5(a) and (e), and adopt the following templates:

$$A_1=\begin{bmatrix}0&0&0\\0.5&2&-0.5\\0&0&0\end{bmatrix},C_{12}=\begin{bmatrix}0&0&0\\0&2.8&-1\\0&0&0\end{bmatrix},\ \begin{matrix}B_1=0,\\I_1=-2,\end{matrix}$$

$$A_2=\begin{bmatrix}0&0&0\\-0.5&2&0.5\\0&0&0\end{bmatrix},C_{21}=\begin{bmatrix}0&0&0\\-1&2.8&0\\0&0&0\end{bmatrix},\ \begin{matrix}B_2=0,\\I_2=-2,\end{matrix}\quad(9)$$

At the beginning of this simulation, the first layer and the second layer of the CNN respectively perform a peeling-off the leftmost pixels and the rightmost pixels of the object. Thus we get the middle results as shown in Fig.5(b) and (f), respectively. After that, the rightmost pixels of the first layer and the leftmost pixels of the second layer are peeled off, then we get the next middle results as shown in Fig.5(c) and (g), respectively. As we can see that Fig.5(c) and (g) are the same with Fig.5(a) and (e), except for the both end pixels which have been peeled off. Thus, one cycle of the peeling-off is completed. Like this cycle continues until the vertical centerline is remained. The transient has arrived at the steady state. The two-layer stable outputs are shown in Fig.5(d) and (h), respectively. So far we obtain the center line of the the object. In this simulation, the zero-fixed boundary condition is also adopted. This dynamic process can be observed from the graph of the sum of state-errors varying with the integration time shown in Fig.6. The sum of state-errors is defined as

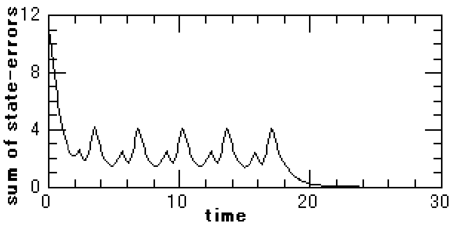

Fig. 6. The sum of state error via time for center line extraction

$$S_{CNN}(t)=\sum_{i=1}^{M}\sum_{j=1}^{N}((x_{1,ij}(t+\Delta t)-x_{1,ij}(t))^2 \\ +(x_{2,ij}(t+\Delta t)-x_{2,ij}(t))^2)\qquad(10)$$

The valleys show the states when the pixels from the two layers are unbalanced states. After the object is completely peeled off and becomes a line, the sum of state-errors decays to zero.

C.2 Detection of the Center Point

To detect the center point from the above center line, we need to apply the same algorithm to the center line with the transposed templates as follows;

$$A_1=\begin{bmatrix}0&0.5&0\\0&2&0\\0&-0.5&0\end{bmatrix},C_{12}=\begin{bmatrix}0&0&0\\0&2.8&0\\0&-1&0\end{bmatrix},\ \begin{matrix}B_1=0,\\I_1=-2,\end{matrix}$$

$$A_2=\begin{bmatrix}0&-0.5&0\\0&2&0\\0&0.5&0\end{bmatrix},C_{21}=\begin{bmatrix}0&-1&0\\0&2.8&0\\0&0&0\end{bmatrix},\ \begin{matrix}B_2=0,\\I_2=-2,\end{matrix}\quad(11)$$

The simulation results are shown in Fig.7. Thus, we can find the center point in two steps.

D. Skeletonization

Skeletonization is a very important task in image processing. The problem of skeletonization using CNN has

0-7803-7278-6/02/$10.00 ©2002 IEEE

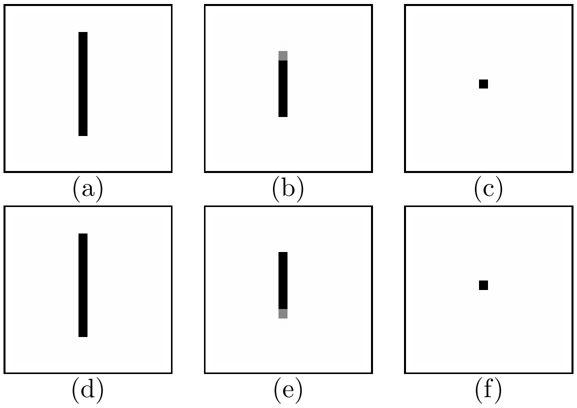

Fig. 7. An example for extracting the center point of line. (a), (b) and (c) are the initial state condition, the transient result and the output of first layer. (d), (e) and (f) are the initial state condition, the transient result and the output of the second layer, respectively.

already attacked in several literatures [2]-[4],etc. Similar to center point dectection, all of the solutions are also based on using complex multilayer, time variant templates or sometimes even nonuniform CNN architectures. similarly, by using the two-layer CNN, the problem can be efficiently solved. The algorithm is described in the following.

The templates are

$$A_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix}, C_{12} = \begin{bmatrix} 0.07 & 0.07 & 0.07 \\ 0.07 & -1.6 & 0.07 \\ 0.07 & 0.07 & 0.07 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 3.1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, C_{21} = \begin{bmatrix} -0.5 & -0.5 & -0.5 \\ -0.5 & 4.2 & -0.5 \\ -0.5 & -0.5 & -0.5 \end{bmatrix},$$

$$B_1 = B_2 = 0, I_1 = I_2 = -1.9, \tag{12}$$

In this simulation, the binary image shown in Fig.8(a) are set to each layer initial states of the CNN. The inputs are arbitrary and the boundary condition is zero-fixed. Thus, we obtained the result from the second layer stable output as shown in Fig.8(b). We have shown the powerfulness of the two-layer CNN for skeletonization, even though there are a few inperfections yet in the second output.

IV. Stability of Two-Layer CNNs

Since the CNNs are usually required to be stable for the applications, the studies on complete stability have been vigorously discussed and many criteria have been obtained. In this section, we will discuss the convergence property, and its related problems for the two-layer CNNs with symmetric and/or special templates.

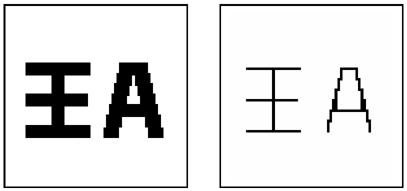

Fig. 8. An example for skeletonization. (a) is the initial states, (b) is stable output of the second layer.

For analyzing the convergence properties of dynamic nonlinear systems, one of the most effective techniques is Lyapunov's method, which has been successfully applied to the stability analysis of single-layer CNN [1]. In this section, we analyze simply the stability of the two-layer CNN with the same techniques. We define a Lyapunov function $E(t)$ of a two-layer CNN by the scalar function, which is similar to the one used by Hopfield in [5] and can be interpreted as the "generalized energy" for a two-layer CNN.

$$\begin{aligned} E(t) \quad = \quad & -\tfrac{1}{2}\textstyle\sum_{(i,j)}\sum_{(k,l)} A_1(i,j;k,l) y_{1,ij} y_{1,kl} \\ & + \textstyle\sum_{(i,j)}\sum_{(k,l)} B_1(i,j;k,l) y_{1,ij} u_{1,kl} \\ & -\tfrac{1}{2}\textstyle\sum_{(i,j)}\sum_{(k,l)} C_1(i,j;k,l) y_{1,ij} y_{2,kl} \\ & +\tfrac{1}{2}\textstyle\sum_{(i,j)} y_{1,ij}^2 - \sum_{(i,j)} I_1 y_{1,ij} \\ & -\tfrac{1}{2}\textstyle\sum_{(i,j)}\sum_{(k,l)} A_2(i,j;k,l) y_{2,ij} y_{2,kl} \\ & + \textstyle\sum_{(i,j)}\sum_{(k,l)} B_2(i,j;k,l) y_{2,ij} u_{2,kl} \\ & -\tfrac{1}{2}\textstyle\sum_{(i,j)}\sum_{(k,l)} C_2(i,j;k,l) y_{2,ij} y_{1,kl} \\ & +\tfrac{1}{2}\textstyle\sum_{(i,j)} y_{2,ij}^2 - \sum_{(i,j)} I_2 y_{1,ij} \end{aligned} \tag{13}$$

Therefore, we can show that the function $E(t)$ is bounded as follows;

$$\mathrm{Max}_t |E(t)| \leq E_{max} \tag{14}$$

Where

$$\begin{aligned} E_{max} \quad = \quad & +\tfrac{1}{2}\textstyle\sum_{(i,j)}\sum_{(k,l)} |A_1(i,j;k,l)| \\ & + \textstyle\sum_{(i,j)}\sum_{(k,l)} |B_1(i,j;k,l)| \\ & +\tfrac{1}{2}\textstyle\sum_{(i,j)}\sum_{(k,l)} |C_1(i,j;k,l)| \\ & +\tfrac{1}{2}\textstyle\sum_{(i,j)}\sum_{(k,l)} |A_2(i,j;k,l)| \\ & + \textstyle\sum_{(i,j)}\sum_{(k,l)} |B_2(i,j;k,l)| \\ & +\tfrac{1}{2}\textstyle\sum_{(i,j)}\sum_{(k,l)} |C_2(i,j;k,l)| \\ & +MN(1 + |I_1| + |I_2|) \end{aligned} \tag{15}$$

Now, we consider two cases having the following cloning templates;

- Case 1

$$\begin{cases} A_1(i,j;k,l) = A_1(k,l;i,j) \\ A_2(i,j;k,l) = A_2(k,l;i,j) \\ C_1(i,j;k,l) = C_1(k,l;i,j) = C(i,j;k,l) \\ C_2(i,j;k,l) = C_2(k,l;i,j) = C(i,j;k,l) \end{cases} \tag{16}$$

- Case 2

$$\left\{\begin{array}{l} A_1(i,j;k,l) = A_1(k,l;i,j) \\ A_2(i,j;k,l) = A_2(k,l;i,j) \\ C_1(i,j;k,l) = C_2(k,l;i,j) \\ C_2(i,j;k,l) = C_1(k,l;i,j) \end{array}\right. \tag{17}$$

From Eq.(13), we have the following relation for the cases of 1 and 2:

$$\frac{dE(t)}{dt} = -\sum_{(i,j)}[(\frac{dy_{1,ij}}{dt})^2 + (\frac{dy_{2,ij}}{dt})^2] \leq 0 \tag{18}$$

Thus, we found that the energy function is monotone decreasing, where we used the following constraint conditions;

$$\frac{dy_{1,ij}}{dx_{1,ij}} = \left\{\begin{array}{ll} 1 & |x_{1,ij}| < 1 \\ 0 & |x_{1,ij}| \geq 1 \end{array}\right. \tag{19}$$

$$\frac{dy_{2,ij}}{dx_{2,ij}} = \left\{\begin{array}{ll} 1 & |x_{2,ij}| < 1 \\ 0 & |x_{2,ij}| \geq 1 \end{array}\right. \tag{20}$$

We find that from Eq.(14) and (18), for any given inputs u_1, u_2 and the initial states x_1, x_2, we obtain that

$$\lim_{t\to\infty} E(t) = const. \tag{21}$$

and

$$\lim_{t\to\infty} \frac{dE(t)}{dt} = 0 \tag{22}$$

Thus, in the cases of 1 and 2, we always have the stable steady state outputs in both two layers after the transient. Now, we show that both the states x_1 and x_2 approaches to the equibrium points. Let us consider the system equations Eq.(1)-(3) again, and rewrite them in the following form;

$$\frac{dx_{1,ij}(t)}{dt} = -f_1(x_{1,ij}(t)) + g_1(t) \tag{23}$$

$$\frac{dx_{2,ij}(t)}{dt} = -f_2(x_{2,ij}(t)) + g_2(t) \tag{24}$$

where

$$f_1(x_{1,ij}) = x_{1,ij} - \frac{1}{2}A_1(i,j;i,j)(|x_{1,ij}+1| - |x_{1,ij}-1|) \tag{25}$$

$$f_2(x_{2,ij}) = x_{2,ij} - \frac{1}{2}A_2(i,j;i,j)(|x_{2,ij}+1| - |x_{2,ij}-1|) \tag{26}$$

$$g_1(t) = \sum_{(k,l)\neq(i,j)} A_1(i,j;k,l)y_{1,kl} + \sum_{(k,l)} C_{21}(i,j;k,l)y_{2,kl} + \sum_{(k,l)} B_1(i,j;k,l)u_{1,kl} + I_1 \tag{27}$$

$$g_2(t) = \sum_{(k,l)\neq(i,j)} A_2(i,j;k,l)y_{2,kl} + \sum_{(k,l)} C_{12}(i,j;k,l)y_{1,kl} + \sum_{(k,l)} B_2(i,j;k,l)u_{2,kl} + I_2 \tag{28}$$

Observe that $g_1(t)$ and $g_2(t)$ are not related to the states $x_{1,ij}$ and $x_{2,ij}$, and are only the functions of the outputs (y_1, y_2), inputs (u_1, u_2) and biases (I_1, I_2). Therefore, for $A_1(i,j;i,j) > 1$ and $A_1(i,j;i,j) > 1$ the trajectries have the same structures as shown in Fig.2, and have the stable and unstable equibrium points. Thus, we can conclude for the cases of 1 and 2 that the steady state of two-layer CNN is completely stable, in the meaning that both states and outputs are in the steady state. The validity of the above analysis are comfirmed from the above simulation examples.

V. Conclusions

In this paper, we preliminarily investigated the image processing applications and stability of the two-layer CNNs. Although the examples proposed in this paper can be solved by single-layer CNNs, the applications of two-layer CNNs to these problems are very efficient compared to the single-layers. Especially, the center point detection and skeletonzation problems with the two-layer CNN can be found the solution in simple methods, however, if we use the single-layer CNN, it will be found many steps depending on the size. We can prove the stability for the special cases of two-layer CNNs, where its necessary conditions are proved by the Lyapunov function and behaviors of the trajectories on the phase plane.

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] 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.

[3] 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.

[4] T. Roska, L. Kek, L. Nemes, A. Zarandy, and P. Szolgay, "CNN Software Libarary (Templates and Algorithms), Version 7.3," Analogical and neural computing laboratory, Computer and Automation Institute of the Hungarian Academy of Sciences, Budapest, Hungary, 1999.

[5] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," in Proc. Natl. acad. sci USA, vol.81, pp.3088-3092,1984.