2010 International Symposium on Nonlinear Theory and its Applications
NOLTA2010, Krakow, Poland, September 5-8, 2010

Characteristics of Cellular Neural Networks with Dynamic Template for Motion Pictures

Masakazu KAWAHARA†, Yoko UWATE† and Yoshifumi NISHIO†

Email:{kawahara, uwate, nishio}@ee.tokushima-u.ac.jp
†Dept. Electrical and Electronic Eng.,
Tokushima University

Abstract—In our previous research, cellular neural networks with dynamic template (D-CNN) have been proposed. In D-CNN, the wiring weights of template are dynamically changed at each update by learning. In this study, we investigate the characteristics of the update template in D-CNN when input images are changed during D-CNN process. We express the variation of the update template through the gray scale. We also calculate the increasing or decreasing ratios of element in the update template. From these obtained results, we confirmed that D-CNN is effective for motion pictures.

1. Introduction

Cellular neural networks (CNN) were proposed by Chua and Yang in 1988 [1]. The idea of CNN was inspired from the architecture of the cellular automata and the neural networks. Unlike the original neural networks, the CNN has local connectivity property. Wiring weights of the cells are established by parameters called the template. The performance of the CNN is decided by the template. Also, the CNN has been successfully used for various high-speed parallel signals processing applications such as image processing application [2][3]. Usually, the templates of all the cells in the CNN are identical and those values do not change during the processing. This is good for implementation but restrict the performance, namely the conventional CNN can not perform image processing based on the local features of input images.

In the previous study, we have proposed cellular neural networks with dynamic template (D-CNN) [4]. In D-CNN, template is dynamically changed at each update by learning. This learning method is inspired from the rank order learning. The updated template depends on the value of cells. From the simulation results of the previous study, we confirmed that the converged value of each cell is divided to two or three values. Also, convergence process is much more rapid than that of the conventional CNN. Then, we investigate update template in D-CNN for motion pictures [5]. However, the mechanism of D-CNN has not been made clear. In this study, we investigate the characteristic of the updated template using CNN with dynamic template in detail. We also set to threshold value initial template for comparison of the updated template. We expect the appearance of characteristics in updated template by changing input image. From the investigation of the value of the updated template, we confirmed the characteristic of updated template in D-CNN by changing input images. The rest of this paper is structured as follows. In the Sec. 2, we review the basic of the standard CNN. In the Sec. 3, we explain the algorithm of the proposed D-CNN. In the Sec. 4, we show the characteristic of the updated template in simple binary images using D-CNN. In the Sec. 5, we show the characteristic of the updated template in real gray scale images using D-CNN. The Section 6 concludes the article.

2. Cellular Neural Networks [1]

In this section, we explain the basic structure of the CNN. The CNN has M by N processing unit circuits called cells. Cells are arranged in a reticular pattern to M line N row. We represent a cell $C(i, j)$ using a variable i which denotes vertical position and a variable j which denotes horizontal position. The cell contains linear and nonlinear circuit elements. The CNN is an array of cells. Each cell is connected to only its neighboring cells according to a template. Usually, the template is the same for all cells except for boundary cells. The CNN has the features of time continuity, spatial discreteness, nonlinearity and parallel processing capability

State equation:

$$\frac{dv_{xij}}{dt} = -v_{xij} + \sum_{k=i-r}^{i+r}\sum_{l=j-r}^{j+r} A_{(i,j;k,l)}v_{ykl}(t) + \sum_{k=i-r}^{i+r}\sum_{l=j-r}^{j+r} B_{(i,j;k,l)}v_{ukl}(t) + I. \quad (1)$$

Output equation:

$$v_{yij}(t) = \frac{1}{2}(|v_{xij}(t) + 1| - |v_{xij}(t) - 1|). \quad (2)$$

where v_x, v_y and v_u represent a state, an output and an input of cell, respectively. In the equation (1), A is the feedback template and B is the control template. These and bias I are collectively called general template. In this equation, the control template depends on input value.

3. CNN with dynamic template

In this section, we explain the algorithm of D-CNN. In our research, we change input images when a certain calculation times comes. In our D-CNN, the templates are updated at every iterations by rank order learning. The learning steps in our D-CNN are described as follows.

STEP 1: The state values and the output values of all the cells in D-CNN are updated according to the discretized model of Eqs. (1) and (2).

STEP 2: Calculate the comparison of the output value of each cell with the one-step-past outputs of the cell and its neighbor cells. The comparison equation for the cell (i, j) is described Eq. (3).

Comparison Equation:

$$Dif(i, j; k, l) = |v^{past}_{y,(i,j)} - v^{now}_{y,(k,l)}|. \quad (3)$$

STEP 3: Among the 9 calculated values of $Dif(i, j; k, l)$, the cells with the smallest and the second smallest values are defined as "winner" and "second", respectively. In our update algorithm, we change the learning rate in two elements. By this step, we find the position of cells with the nearest and the second nearest values to the corresponding cell (i, j).

STEP 4: Update the elements of the template corresponding to the positions of the "winner" and the "second". Note that in our proposed learning algorithm only two elements are updated. The update method and the update function are described as follows.

Update Method:

Assume that the template before update is given as Eq. (4).

$Template^{now}$:

$$A^{now}_{(i,j)} = \begin{bmatrix} a^{now}_{11} & a^{now}_{12} & a^{now}_{13} \\ a^{now}_{21} & a^{now}_{22} & a^{now}_{23} \\ a^{now}_{31} & a^{now}_{32} & a^{now}_{33} \end{bmatrix},$$

$$B^{now}_{(i,j)} = \begin{bmatrix} b^{now}_{11} & b^{now}_{12} & b^{now}_{13} \\ b^{now}_{21} & b^{now}_{22} & b^{now}_{23} \\ b^{now}_{31} & b^{now}_{32} & b^{now}_{33} \end{bmatrix},$$

$$I^{now}_{(i,j)} = I^{now}. \quad (4)$$

For example, we consider the case that the "winner" is (i, j) and the "second" is $(i - 1, j - 1)$. In that case, only a^{now}_{22}, b^{now}_{22}, a^{now}_{11} and b^{now}_{11} in Eq. (4) are updated. The threshold value I is not updated in our learning method.

In our update algorithm, we change the learning rate in two elements. The learning rates of the "winner" and the "second" are shown as follows.

Learning rate:

$$R_1 = R_{10}\left(1 - \frac{Number\ of\ calculation}{Number\ of\ calculation_{max}}\right). \quad (5)$$

$$R_2 = R_{20}\left(1 - \frac{Number\ of\ calculation}{Number\ of\ calculation_{max}}\right). \quad (6)$$

In this study, we decide $Number\ of\ calculation_{max}$ in Eqs. (5) and (6) to be set to 10. Namely, the learning rates of "winner" and "second" are changed until 10 calculations. Then, after $Number\ of\ calculation_{max}$ becomes over 10, the learning rates of "winner" and "second" become 0 and the templates are not updated. By using the learning rate, the elements of the template are updated according to the following update equation.

Update Equation:

$$a^{updated}_{winner} = a^{now}_{winner} - R_1(v^{past}_{y,(i,j)} - v^{now}_{y,(i,j)}). \quad (7)$$

$$a^{updated}_{second} = a^{now}_{second} - R_2(v^{past}_{y,(i,j)} - v^{now}_{y,(i,j)}). \quad (8)$$

R_1 and R_2 decrease according to the Eqs. (5) and (6). The initial learning rates are given as follows.

Initial Learning rate:

$$Winner \ : \ R_{10} \quad (0 \le R_{10} \le 0.1). \quad (9)$$

$$Second \ : \ R_{20} = R_{10}/4. \quad (10)$$

After the update using Eqs. (7) and (8), the updated template is shown as follows. In Eq. (11), $a^{updated}_{11}$ and $a^{updated}_{22}$ are the updated values. Also, $b^{updated}_{11}$ and $b^{updated}_{22}$ are updated similarly.

$Template^{updated}$:

$$A^{updated}_{(i,j)} = \begin{bmatrix} a^{updated}_{11} & a^{now}_{12} & a^{now}_{13} \\ a^{now}_{21} & a^{updated}_{22} & a^{now}_{23} \\ a^{now}_{31} & a^{now}_{32} & a^{now}_{33} \end{bmatrix},$$

$$B^{updated}_{(i,j)} = \begin{bmatrix} b^{updated}_{11} & b^{now}_{12} & b^{now}_{13} \\ b^{now}_{21} & b^{updated}_{22} & b^{now}_{23} \\ b^{now}_{31} & b^{now}_{32} & b^{now}_{33} \end{bmatrix},$$

$$I^{updated}_{(i,j)} = I^{now}. \quad (11)$$

STEP 5: The steps from 1 to 4 are repeated.

These learning steps inspired from the rank order learning.

4. Investigation Results

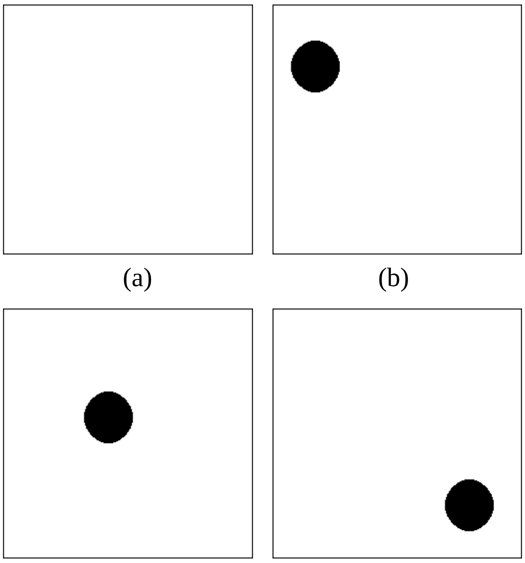
In this section, we show the investigation results for two types of motion pictures. We express the variation of the updated template in D-CNN through the graduation value. In the first step of this investigation, an initial template is set to D-CNN. The elements of the initial template are updated by using updated method in the previous section. The initial template used in this study is described as follows.

Initial Template:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0.07 & 0.1 & 0.07 \\ 0.1 & 0.32 & 0.1 \\ 0.07 & 0.1 & 0.07 \end{bmatrix}, I = 0. \quad (6)$$

4.1. The Updated Template in Binary images

Firstly, we investigate updated template for simple binary image. We change 4 input images every 10 [τ] like a motion picture.



(a) (b) (c) (d)

Figure 1: Motion pictures 1. (a) Input image 1. (b) Input image 2. (c) Input image 3. (d) Input image 4.

Figure 1 shows binary input images for motion pictures. In Fig. 1, we recognize the black object appeared and moved up to down through Figs. 1(a) to (d). By changing these input images, we investigate the characteristic of the updated templates.

In this investigation, we show that how to update the dynamical template. Namely, we calculate the difference of elements in updated template from the initial template. We set to the threshold value in updated template. The threshold value of A and B templates are described as follows.
Threshold value of template:

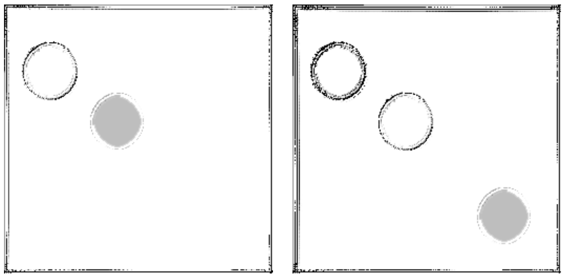
$$Th_A(i, j) = \sum_{i=1}^{3}\sum_{j=1}^{3} a_{ij} \tag{12}$$

$$Th_B(i, j) = \sum_{i=1}^{3}\sum_{j=1}^{3} b_{ij}. \tag{13}$$

Namely, the threshold value is summed all of the elements in the each template. From our update algorithm, the rate of update is same rate in Th_A and Th_B.

Figure 2 shows the variation of the update template through the gray scale. In Fig. 2, if the average of update template is over Th, the pixels near to black. While, if the average of update template is under Th, the pixels near to white. From these simulation results, we can confirm that the dynamical templates of the changing area from the previous image is more updated than the constant image area.

Figure 3 shows the percentage of updated template whether the value of all the elements of updated template under Th and over Th. In these graphs, we can say that the number of the update template is varied by changing input image.



(a) (b)

Figure 2: The graduation of updated template. (a) The update template after processing for the Input image 3. (b) The final update template.

4.2. The Updated Template in Gray Scale Images

Next, we investigate updated template for gray scale image. Similar to the previous investigation, we change 4 input images every 10 [τ] like a motion pictures.

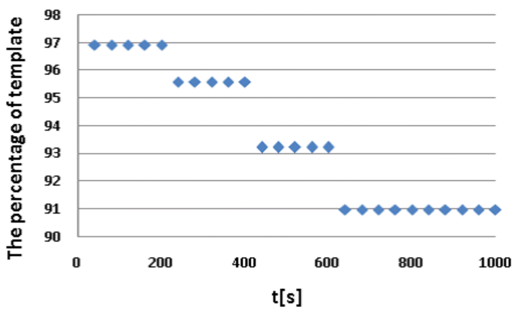
Figure 4 shows the input images as motion pictures. We confirm the doll is appeared and gradually move left to right through Figs. 3(a) to (d). Using these input images, we investigate the characteristic of update template in D-CNN.

Figure 5 shows the variation of the update template through the graduation value with gray scale. In Fig. 5, the area of the edge is most changed by changing input image. From these results, we can say that the edge of new object is more updated by inputted next input image.

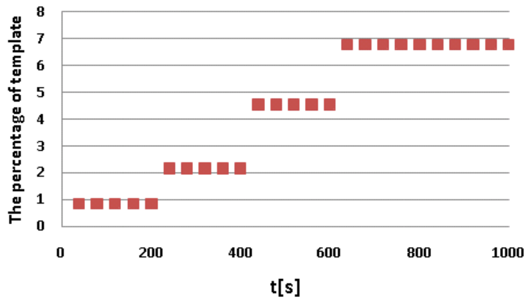
Next, we investigate the process of the number of the updated template. Figure 6 shows the the number of the updated template is over Th and under Th. In Fig. 6, the number of template over Th decreases gradually. On the other hand, the number of template under Th increases gradually. From these results, we confirm that the average of the changing rate in updated template is varied widely when the next input image is inputted. Even if next input image is inputted, the updated template influence of before input image. Therefore, we can say that our proposed CNN is effective for the motion picture processing.

5. Conclusions

In this study, we have investigated the characteristics of the update template in D-CNN when input images were changed during D-CNN process. In D-CNN, the template was changed by rank order learning. We have expressed the variation of the update template through the gray scale. We also calculate the increasing or decreasing ratios of element in the update template. From instigation results, we have confirmed that the D-CNN is effective for the application of motion pictures.



(a)



(b)

Figure 3: The number of updated template in Fig. 1. (a) The percentage of template where under $Th_{(i,j)}$. (b) The percentage of template where over $Th_{(i,j)}$.

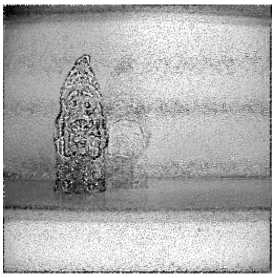
(a)

(b)

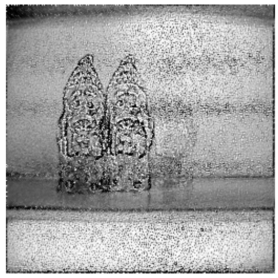
(c)

(d)

Figure 4: Input images. (a) Input image1. (b) Input image2 (after 10 [τ]). (c) Input image3 (after 10 [τ]). (d) Output image before input image is changed.



(a)



(b)

Figure 5: The graduation of the updated template. (a) The updated template during simulation. (b) The final updated template.

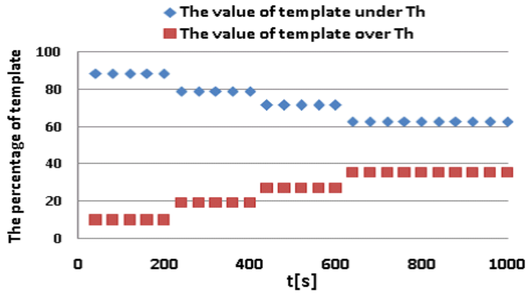


Figure 6: The number of updated template in Fig. 5.

References

[1] L. O. Chua and L. Yang, "Cellular Neural Networks:Theory," IEEE Trans. Circuits Syst., vol. 32, pp. 1257-1272, Oct. 1988.

[2] F. Dirk and T. Ronald, "Coding of Binary Image Data using Cellular Neural Networks and Iterative Annealing," Proc. of ECCTD'03, vol. 1, pp. 229-232, Sep. 2003.

[3] M. Namba and Z. Zhang, "Cellular Neural Network for Associative Memory and Its Application to Braille Image Recognition," Proc. of IJCNN'06, pp. 4716-4721, Jul. 2006.

[4] M. Kawahara, T. Inoue and Y. Nishio, "Cellular Neural Network with Dynamic Template and its Output Characteristics," Proc. of IJCNN'09, pp. 1552-1558, Jun. 2009

[5] M. Kawahara, T.Inoue and Y. Nishio, "Image Processing Application Using CNN with Dynamic Template" Proc. of CNNA'10, pp. 41-46, Feb. 2010.