2011 International Workshop on Nonlinear Circuits,
Communication and Signal Processing
NCSP'11, Tianjin, China, March 1-3, 2011

Investigation of Output Characteristics of Cellular Neural Networks with Dynamic Template

Masakazu KAWAHARA†, Yoko UWATE† and Yoshifumi NISHIO†

†Tokushima University
2-1 Minami-Josanjima, Tokushima, Japan
Phone:+81-88-656-7470,FAX:+81-88-656-7471
Email: {kawahara, uwate, nishio}@ee.tokushima-u.ac.jp

Abstract

In the previous study, we have proposed CNN with dynamic template (D-CNN). In D-CNN, template is dynamically changed at each update of learning. In this study, we consider the two types of update equations for the dynamic template. We investigate the characteristics of the output value of D-CNN by using two update equations.

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 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 however 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 CNN with dynamic template (D-CNN) [2]. 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 output value of cells. From the simulation results of the previous study, we have confirmed that the converged value of each cell is divided to two or three values. Then, we have investigated update template in D-CNN for motion pictures [3][4].

In this study, we investigate the output characteristics of D-CNN for four types of motion pictures and compare the performance of D-CNN for extracting moving object by using two types of update equations. In order to make clear the mechanism of D-CNN, we focus on the output values of the D-CNN after image processing.

The rest of this paper is structured as follows. In the Sec. 2, we show the algorithm of the proposed D-CNN. In the Sec. 3, simulation results using the proposed D-CNN are shown. The Sec. 4 concludes the article.

2. Cellular Neural Networks with Dynamic Template [2]

In this section, we explain the algorithm of D-CNN. The input images are changed when a certain calculation time comes. In this proposed 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 state equation and output equation.

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:

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

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 by 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 the 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).

$Template^{now}$:

$$A^{now}_{(i,j)} = \begin{bmatrix} a^{now}_{11} & a^{now}_{12} & a^{now}_{13} \\ a^{now}_{21} & a^{now}_{22} & a^{now}_{23} \\ a^{now}_{31} & a^{now}_{32} & a^{now}_{33} \end{bmatrix},$$
$$B^{now}_{(i,j)} = \begin{bmatrix} b^{now}_{11} & b^{now}_{12} & b^{now}_{13} \\ b^{now}_{21} & b^{now}_{22} & b^{now}_{23} \\ b^{now}_{31} & b^{now}_{32} & b^{now}_{33} \end{bmatrix},$$
$$I^{now}_{(i,j)} = I^{now}. \tag{4}$$

For example, we consider the case that the "winner" is $C_{(i,j)}$ and the "second" is $C_{(i-1,j-1)}$. 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.

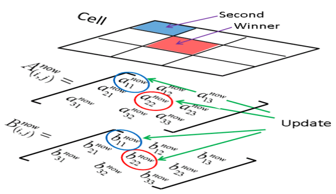


Figure 1: Decision of updated elements of template in D-CNN.

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:

$$R_1 = R_{10}\left(1 - \frac{Number\ of\ calculation}{Number\ of\ calculation_{max}}\right). \tag{5}$$
$$R_2 = R_{20}\left(1 - \frac{Number\ of\ calculation}{Number\ of\ calculation_{max}}\right). \tag{6}$$

The parameters of R_{10} and R_{20} are set to 0.1 and 0.025, respectively. And $Number\ of\ calculation_{max}$ 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.

In this study we define two types update equations ("Type1" and "Type2") of D-CNN as follows.

Update Equation (Type1):

$$a^{updated}_{winner} = a^{now}_{winner} + R_1(v^{past}_{y,(i,j)} - v^{now}_{y,(i,j)}). \tag{7}$$
$$a^{updated}_{second} = a^{now}_{second} + R_2(v^{past}_{y,(i,j)} - v^{now}_{y,(i,j)}). \tag{8}$$

Update Equation (Type2):

$$a^{updated}_{winner} = a^{now}_{winner} - R_1(v^{past}_{y,(i,j)} - v^{now}_{y,(i,j)}). \tag{9}$$
$$a^{updated}_{second} = a^{now}_{second} - R_2(v^{past}_{y,(i,j)} - v^{now}_{y,(i,j)}). \tag{10}$$

Where, R_1 and R_2 decrease according to the Eqs. (5) and (6). In the case of Type1, the amount of element change is summed to the current element value. While, in the case of Type2, the amount of element change is took away from the current element value.

The initial learning rates are given as follows.

Initial Learning rate:

$$Winner \quad : \quad R_{10} \quad (0 \le R_{10} \le 0.1). \tag{11}$$
$$Second \quad : \quad R_{20} = R_{10}/4. \tag{12}$$

After the update using Eqs. (7) and (8) or Eqs. (9) and (10), the updated template is shown as follows. In Eq. (13), $a^{updated}_{11}$ and $a^{updated}_{22}$ are the updated values. Also, $b^{updated}_{11}$ and $b^{updated}_{22}$ are updated similarly.

$Template^{updated}$:

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

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

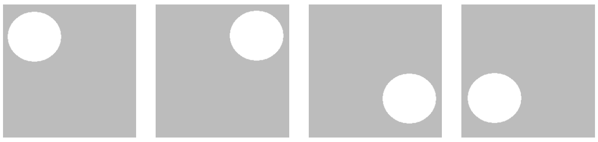
3. Simulation Results

In this section, we show the simulation results by using two types update equations ("Type1" and "Type2") of D-CNN. Figure 2 to 5 show the four types input images as motion pictures. We can recognize the position of object is changed through the four figures.

For simulations, an initial template is set as follows.

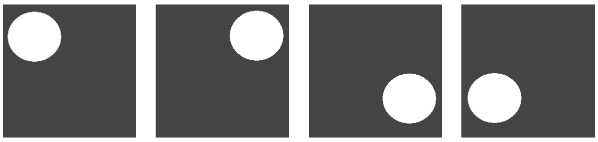
Initial Template:

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0.07 & 0.1 & 0.07 \\ 0.1 & 0.32 & 0.1 \\ 0.07 & 0.1 & 0.07 \end{bmatrix}, I = 0. \tag{14}$$



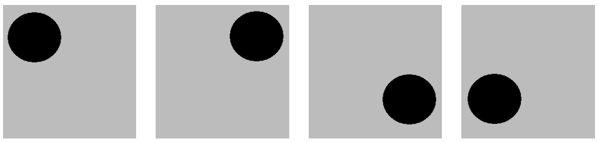
(a) (b) (c) (d)

Figure 2: Input images. (a) Input image 1. (b) Input image 2. (c) Input image 3. (d) input image 4.



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

Figure 3: Input images. (a) Input image 1. (b) Input image 2. (c) Input image 3. (d) input image 4.



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

Figure 4: Input images. (a) Input image 1. (b) Input image 2. (c) Input image 3. (d) input image 4.

Figures 6 to 9 show the output images for the input images in Figs. 2 to 5. From these results, we can say that the output images have different characteristics depending on the two update equations.

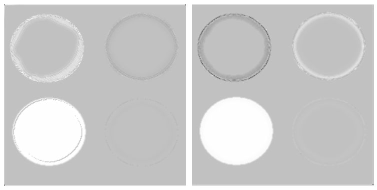
Next, we investigate the distribution of the output values in output images. We divided output image to four area like Fig. 10.

As an example, the distributions of output value in A area of Figs. 6 to 9 are summarized in Tab. 1. In the results of Figs. 6 and 8, we can recognize the value of background is concentrate on -0.6 to -0.4. While, we can recognize the value of background is concentrate on 0.4 to 0.6 in the results of Figs. 7 and 9. Furthermore, we can confirm that the distribution of the output values by using "Type2" equation is spread widely than the case of "Type1".

Next, we propose the evaluation method how clear moving object is extracted by using two update equations. We expect that if the number of pixel which is belong to far value form

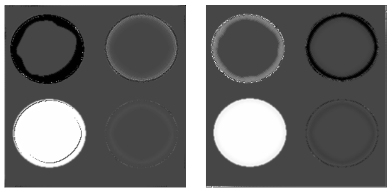
(a) (b) (c) (d)

Figure 5: Input images. (a) Input image 1. (b) Input image 2. (c) Input image 3. (d) input image 4.



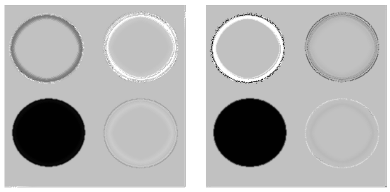
(a) (b)

Figure 6: Output images for Figs. 2. (a) Output image using "Type1". (b) Output image using "Type2".



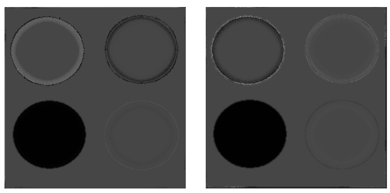
(a) (b)

Figure 7: Output images for Fig. 3. (a) Output image using "Type1" in Fig. 3. (b) Output image using "Type2" in Fig. 3.



(a) (b)

Figure 8: Output images for Fig. 4. (a) Output image using "Type1". (b) Output image using "Type2".



(a) (b)

Figure 9: Output images for Fig. 5. (a) Output image using "Type1" in Fig. 5. (b) Output image using "Type2" in Fig. 5.

Table 1: Distribution of output image in *A* area.

The value of pixel	The number of pixel							
	Figure 6		Figure 7		Figure 8		Figure 9	
	Type1	Type2	Type1	Type2	Type1	Type2	Type1	Type2
$-1.0 \sim -0.8$	739	4	10	93	85	1672	0	0
$-0.8 \sim -0.6$	2036	87	10	35	158	104	0	0
$-0.6 \sim -0.4$	12338	13669	4	23	13007	12853	2	2
$-0.4 \sim -0.2$	16	1166	4	72	563	97	4	7
$-0.2 \sim 0$	0	112	16	849	806	53	5	54
$0 \sim 0.2$	0	36	16	1045	502	59	135	186
$0.2 \sim 0.4$	0	27	153	629	4	37	2138	446
$0.4 \sim 0.6$	0	39	11791	12364	1	54	12735	14219
$0.6 \sim 0.8$	0	0	352	38	1	18	71	703
$0.8 \sim 1.0$	0	0	2870	12	2	182	152	1

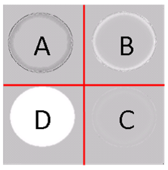


Figure 10: Four areas of output images.

the background value is large, it is easier to extract the moving object. We calculate the number of pixel in whole image exclude the value of $-0.8 \sim -0.2$ (background value) in Figs. 6 and 8. Table 2 shows the calculated results. For the case of Fig. 6, the number of pixel is large by using "Type1" which means that "Type1" equation is better to extract the moving object when background color is more dark than the object. On the other hand, in the case of Fig. 8, moving object is more extracted by using "Type2" when background color is more bright than the object.

Table 2: Number of pixel (value of background is near to -1).

Number of pixel exclude the value of background	Figure 6		Figure 8	
	Type1	Type2	Type1	Type2
	739	218	1401	2075

Also, we calculate the number of pixel in whole images exclude the value of $0.2 \sim 0.8$ (background color) in Figs. 7 and 9. Table 3 shows the calculated results. For the case of Fig. 7, the number of pixel is large by using "Type1" which means that "Type1" equation is better to extract the moving object when background color is more dark than the object. On the other hand, in the case of Fig. 9, the number of pixel of "Type1" and "Type2" has similar values. This is because, for the input image Fig. 5, the background and the object color are not so difference. So, there are not so much difference between "Type1" and "Type2".

Table 3: Number of pixel (value of background is near to 1).

Number of pixel exclude the value of background	Figure 7		Figure 9	
	Type1	Type2	Type1	Type2
	2930	2129	298	250

4. Conclusions

In this research, we have investigated the characteristics of output values in D-CNN for motion pictures. We compared the output values of the cell by two update equations. From simulation results, we can see that when the moving object is more bright than the background, "Type1" equation performs better than "Type2" for the extracting moving object. On the other hand, when the moving object is more dark than the background, "Type2" equation performs well.

References

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

[2] 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

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

[4] M. Kawahara, Y. Uwate and Y. Nishio, "Characteristics of Cellular Neural Networks with Dynamic Template for Motion Pictures" Proc. of NOLTA'10, pp. 23-26, Sep. 2010.