2010 12th International Workshop on Cellular Nanoscale
Networks and their Applications (CNNA)

CNN Template Design Using Back Propagation Algorithm

Masashi Nakagawa, Takashi Inoue and Yoshifumi Nishio
Department of Electrical and Electronic Engineering, Tokushima University
2-1 Minami-Josanjima, Tokushima, 770-8506, Japan
Email: {miyabi, takashi, nishio}@ee.tokushima-u.ac.jp

Abstract— **In this article, detailed investigation of the template design method of cellular neural networks with back propagation algorithm, which was proposed by the authors before, is carried out. The performance of the method is also evaluated by using the average error which corresponds to the difference between the output produced by the designed template and the ideal output. Furthermore, the method is applied for gray scale image applications.**

I. Introduction

Cellular neural networks (CNN) were introduced by Chua and Yang in 1998 [1]. The idea of the CNN is inspired from the architecture of the cellular automata and the neural networks. Unlike the conventional neural networks, the CNN has local connectivity property. Since the structure of the CNN resembles the structure of animals retina, the CNN can be used for various image processing applications [2]-[4].

Wiring weights of the cells in CNN are established by parameters called template. The template is most important parameter in the CNN, because performance of the CNN is decided by the template. Thus, some template learning schedule as template learning method using GA algorithm are proposed [5]. These works are important subject in the studies of the CNN.

In the previous study, we have proposed a template learning method of cellular neural networks using the back propagation (BP) algorithm [6]. As well known, BP is one of the supervised learning methods [7]. In that method, the template of CNN is dynamically updated from the error between an output image and an ideal image as BP neural networks. We have confirmed that the method could work for a simple example using binary image by computer simulation. However, in the previous study, since we carried out the simulation only for a set of one input image and its corresponding teacher image, more detailed investigation of the method should be done in order to design better templates for a given tasks.

In this study, we carry out more detailed investigation of our template design method with various learning rates and different learning schedules. We also evaluate the performance by using the average error which corresponds to the difference between the output produced by the designed template and the ideal output. Furthermore, we try to use our method for gray scale image applications.

In the Sec. 2, we review the basic of the standard CNN. In the Sec. 3, we review the basic of BP neural networks. In the Sec. 4, we show the proposed template learning method using BP. In the Sec. 5, we show the simulation results. Section 6 concludes the article.

II. Cellular Neural Networks

In this section, we explain the basic structure of the CNN. The CNN has M by N processing unit circuits called cells. The 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 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.

The state equation and the output equation of the cell are shown as follows.

State equation:

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

Output equation:

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

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.

III. Back Propagation Algorithm

BP is the most common learning algorithm for feed forward neural networks. BP uses the error of an output signal and a teacher signal for learning. The total error E of the network is defined as the following equation.

$$E = \sum_{p=1}^{P} E_p = \sum_{p=1}^{P}\left\{\frac{1}{2}\sum_{i=1}^{N}(t_{pi}-o_{pi})^2\right\}, \qquad (3)$$

where P is the number of the input data, N is the number of the neurons in the output layer, t_{pi} denotes the value of the

978-1-4244-6678-8/10/$26.00 ©2010 IEEE

desired target data for the pth input data, and o_{pi} denotes the value of the output data for the pth input data. The goal of the learning is to set weights between all layers of the network to minimize the total error E. In order to minimize E, the weights are adjusted according to the following equation.

$$\begin{aligned} w_{i,j}^{k-1,k}(m+1) &= w_{i,j}^{k-1,k}(m) + \sum_{p=1}^{P} \delta_p w_{i,j}^{k-1,k}(m), \\ \delta_p w_{i,j}^{k-1,k}(m) &= -\eta \frac{\partial E_p}{\partial w_{i,j}^{k-1,k}}, \end{aligned} \tag{4}$$

where $w_{i,j}^{k-1,k}$ is the weight between the ith neuron of the layer $k-1$ and the jth neuron of the layer k, m is the learning time, and η is a proportionality factor known as the learning rate.

Thus, BP brings an output close to the ideal output by changing weight.

IV. Proposed Template Learning Method

In this section, we describe the proposed template learning method using BP algorithm. Our template learning method is inspired from the flexibility of BP algorithm. In particular, in the BP neural networks, parameter is changed by using error between an output and an ideal value, in order to bring an output close to an ideal value. From this perception of learning, template of CNN is changed by using error between an output image and an ideal image. Additionally, this template learning carry out dynamically.

First, we describe the structure of template learning method using BP algorithm. Second, we present four patterns template learning schedules for investigation of an output characteristics.

A. Procedure of Template Learning Method

Processing procedure for learning is explained as follows. Figure. 1 show the proposed template learning method using BP algorithm.

Step 1: An input image and an ideal image (a teacher image) are prepared. An initial template is decided at random, in order to be able to learn the template for any initial values. However, in the template learning, we should mention the stability of template in the CNN. Because the output value of each cell in the CNN may become unstable with the value of a template. The condition of stability is that the center element of template A maintains 1 and the symmetry of each element.

Step 2: CNN using Runge-Kutta method process an input image. At this time, unit width of Runge-Kutta method is made small, and only a few processes. It is for using the image currently processed not using the processed image in order to make learning dynamic.

Step 3: The error between an output image and an ideal image is calculated for each pixel. An error E_p is calculated as the following equation:

$$E_p = (t_p - o_p), \tag{5}$$

where p is the number of the pixel in input image, t_p and o_p are represent the value of the ideal image and output value of the pth pixel respectively. This step is shown in "Calculate error" of Fig. 1.

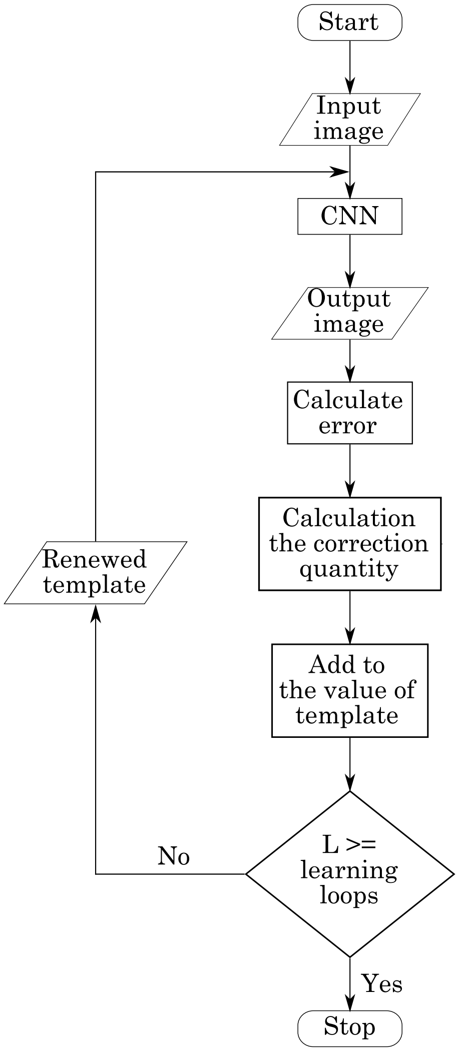

Fig. 1. Proposed template learning method.

Step 4: The corrected direction δ_p is decided using error E_p as the following equation:

$$\delta_p = E_p \times (Vx_p^{now} - Vx_p^{past}), \tag{6}$$

where Vx_p^{past} and Vx_p^{now} are represent past value of Vx_p and present value of Vx_p respectively.

Step 5: The correction quantity Δ_p for each pixel is calculated as the following equation:

$$\Delta_p = \eta \times \delta_p \times Vy_p^{now}, \tag{7}$$

where η is a proportionality factor known as the learning rate. This step is shown in "Calculation the correction quantity" of Fig. 1.

Step 6: The errors of all the pixels are added, hence the average of the all errors D is calculated as the following equation.

$$D = \frac{1}{p}\sum_{i=1}^{p}\Delta_i, \tag{8}$$

In this article, each element of template are decided as equation (9), because of an input image is only one-dimensional image.

$$\begin{aligned} A &= \begin{bmatrix} 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 0 \end{bmatrix}, \\ B &= \begin{bmatrix} 0 & 0 & 0 \\ b_{21} & b_{22} & b_{23} \\ 0 & 0 & 0 \end{bmatrix}, \\ I &= i \end{aligned} \tag{9}$$

The update equation of template is shown as follow.

$$\begin{aligned} a_{21}^{Renewed} &= a_{21}^{Before} + D, \\ a_{23}^{Renewed} &= a_{23}^{Before} + D, \\ b_{21}^{Renewed} &= b_{21}^{Before} + D, \\ b_{22}^{Renewed} &= b_{22}^{Before} + D, \\ b_{23}^{Renewed} &= b_{23}^{Before} + D. \end{aligned} \tag{10}$$

The average D adds one row of the middle elements in the template A and B. However, it does not include the center element of the template A, and the bias I. It is because that the center element of template A is fixed with 1 in order to protect the stability condition of a template. Namely, the reason of the above update method is that the stability of the template can be kept. This step is shown in "Add to the value of template" of Fig. 1.

Step 7: Step. 2 - Step. 6 are repeated until it reaches the set-up number of learning times L.

By using this learning algorithm, template of each cell is updated for bringing close to an ideal image.

B. Four patterns of learning schedule

In this study, we prepare the four patterns of leaning schedule for investigation of output characteristics. Figure 2 show the flow-chart of template learning method. In Fig. 2, we focus on the two parameters as follow. The number of learning in the BP algorithm and number of repetitions of group are described L and λ, respectively.

The parameter of four patterns of learning schedule are shown in TABLE I.

TABLE I
LEARNING SCHEDULE

	Number of learnings L	Set of images	Number of group repetitions λ	The total of learning times
Pattern 1	10	20	10	2000
Pattern 2	25	20	4	2000
Pattern 3	50	20	2	2000
Pattern 4	100	20	1	2000

"number of learnings" L shows the number of times of learning of the BP algorithm, and λ shows the number of times of a repetition group.

The each block of Fig. 2 is explained as follows.

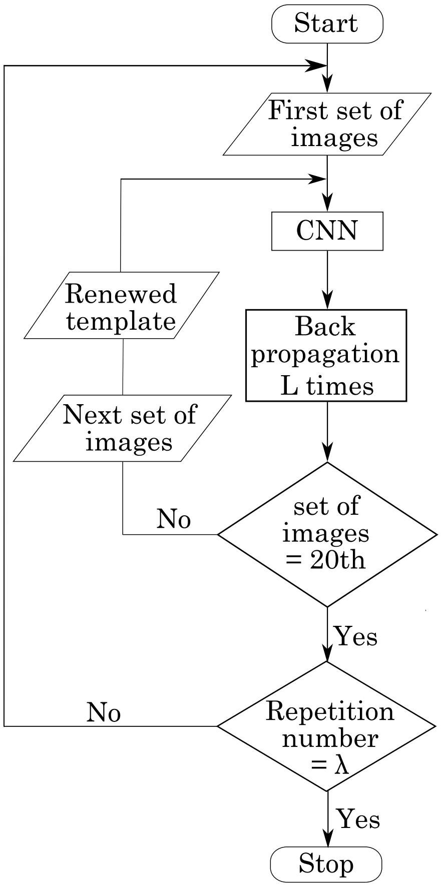

Fig. 2. Procedure for learning.

Step 1. *Preparation of input and ideal image*
First, 20 sets of input and ideal images are prepared.

Step 2. *Decision the learning rate and intial template*
Second, the learning rate is chosen 0.01, 0.05 and 0.1. Each values of an initial template are decided at the random as Step 1 of previous section.

Step 3. CNN

Third, one of the 20 sets of an input and an ideal image is chosen. CNN processes the input image by using a learning rate and an initial template. An input image is processed by using the learning rate decided at Step 2.

Step 4. BP

Forth, an initial template is updated by using BP. In the four patterns learning, the number of learnings L is chosen 10, 25, 50 or 100.

Step 5. *Changing the input and ideal images*

Sixth, after template learning using BP algorithm, input and ideal images are changed to another set of images.

Step 6. *Repetition*

Step 3 - Step 5 are repeated until 20th set of images finished. In this time, updated template using BP is continued to the next processing using next input and ideal image.

Step 7. *Group Repetition*

Step 3 - Step 6 are repeated until the number of set comes λ. where, λ is the whole number of times of a repetition.

Step 8. *Obtained learned template*

After all learning is finished, we can obtain the final learned template.

Although the schedule of learning is different, they can be referred to have the same total learning times of 2000.

V. Simulation results

A. Parameters for Learning

The following parameters are used in this simulation.

TABLE II

Parameters

Name of parameter	value
Step size	0.01
Maximum iteration number	1000
Number of pixels	40
Learning rate	0.01 0.05 0.1

"Step size" and "Maximum iteration number" are parameter of Runge-Kutta method. Learning rate is η of equation (7)

B. For binary image

First, we show the four patterns of template learning results for one-dimensional binary image. The number of pixels is 40 and they are placed at random. The black pixels correspond to the value +1, while the white pixels correspond to the value -1. An ideal image is made from the following two rules.

Rule 1. If the colors of the neighbors of a certain cell is the same color, the cell's color turns into its neighboring color.

Rule 2. If the colors of the neighbors of a certain cell is different, the cell's color does not change.

That is to say, this rule is like "Reversi".

First, ten kinds of different images as an input image are prepared. These pixels are also arranged at random. These images are processed as an input image using the template of twelve patterns. The average error when processing ten images in each template is shown in TABLE III.

TABLE III

Average error

		Number of learnings L			
		10	25	50	100
Learning rate	0.01 [%]	15.75	30.75	35.75	25
	0.05 [%]	21.5	21.75	11.75	23
	0.1 [%]	0	18.5	14.75	27
	0.25 [%]	8.75	18.5	11.75	21.75
	0.5 [%]	4	12.75	9.5	21.75
	0.75 [%]	4	8.75	13.5	12.75
	1 [%]	3.25	3.75	5.5	14

An error is so low that a learning rate is high. Because, if a learning rate becomes large, the updating width of a template becomes large and more dynamic learning can be carried out. Moreover, an error is so low that the number of times of learning is lessened and a group is learned repeatedly. If there is much number of times of learning, it will be completed by updating and will not become dynamic in many cases. Then, there is little number of times which learns a group, and dynamic updating is not carried out. That is, if there is much number of times of learning, that dynamic renewal of a template is carried out will decrease. From this result, a learning rate is large and there should be just little number of times of learning moderately.

C. For gray-scale image

Second, we show learning result for one-dimensional gray scale image. The number of pixels is 40 and they are put the value of pixels at random. The value of range is from +1 to -1. We consider the task of gradient detection. An ideal image of this task is made from the following three rules.

Rule 1. The absolute value between a certain pixel and its right-hand pixel is calculated.

Rule 2. If the absolute value is one or larger, the pixel changes to black.

Rule 3. If the absolute value is less than one, the pixel changes to white.

We prepare 20 one-dimensional input images and the corresponding ideal images and one example is shown in Figs. 3 and 4.

Input image:

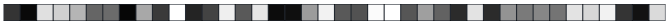

Fig. 3. Input image.

Ideal image:

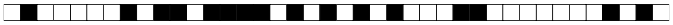

Fig. 4. Ideal image.

Template learning is carried out to gray scale on these condition. However, only the following a result have come out.

TABLE IV
PARAMETER AND RESULT

Learning rate	0.01
The number of times of BP	50
The whole number of a repetition times	2
Error Rate [%]	40.75

Only the result in which an error is very high is obtained. The error of each pixel of an input image and a teacher image is large, and it seems by the present proposed method that learning was not carried out. It is necessary to change the proposal technique itself into the technique of the ability to respond also to a big error. Since most image processing cannot be carried out if a gray scale image cannot be processed, the proposed technique should be improved for gray scale images.

VI. CONCLUSIONS

In this article, we have investigated the template design method of cellular neural networks with back propagation algorithm in more detail. The performance of the method was evaluated by using the average error. Furthermore, the method was applied for a gray scale image application.

At the moment, we do not say that the proposed template learning method exhibited a superior performance than the existing other template learning methods. However, we feel that we obtained some results to broaden the research on the template learning of CNN.

Improving the method for gray scale images or/and 2-dimensional images is our important future works.

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," Prof. of ECCLES'03, vol. 1, pp. 229-232, Sep. 2003.

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

[4] H. Koeppl and L.O. Chua, "An Adaptive Cellular Nonlinear Network and its Application," Proc. of NOLTA'07, pp. 15-18, Sep. 2007.

[5] T. Kozek, T. Roska and L.O. Chua, "Genetic Algorithm for CNN Template Learning," IEEE Trans. Circuits Syst., vol. 40, pp. 392-401, Jun. 1993.

[6] M. Nakagawa, T. Inoue and Y. Nishio, "Template Learning of CNN Using Back Propagation Algorithm," (being submitted to a conference), 2009.

[7] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning Internal Representations by Error Propagation," Parallel Distributed Processing, vol. 1, pp. 318-362, 1986.