Chapter 2

Cellular Neural Network

2.1 Introduction

The cellular neural network (abbreviated as CNN) is proposed by Chua and Yang in 1988 [1]-[2]. It is more general than Hopfield neural network. The state value of one node (cell) at the next time is influenced by inputs and outputs of nodes near this node. Cellular neural networks have the characteristic of parallel processing. The next states of all nodes can be evaluated at the same time. So the operation speed is very fast. And because CNN has the characteristic of local connection of nodes, so CNN is suited for realizing with VLSI. The structure of cellular neural networks is regular, parallel array and local connection etc., not only suitable for the integrated circuit, but also can deal with some special calculations that have the characteristic of local regular connection. Cellular neural networks mostly were applied to the image processing application at first, for example noise removal, edge detection etc.

2.2 The Structure of Cellular Neural Networks

The primary element of CNN is a cell. Cells are arranged in a two-dimensional array usually, as Fig. 2.1 shows. Every cell is only influenced directly by its neighboring cells in a cellular neural network, it is not influenced directly by other cells, and the cells not neighboring on each other are influenced each other indirectly by the characteristic of continuous time dynamical propagation of cellular neural networks.

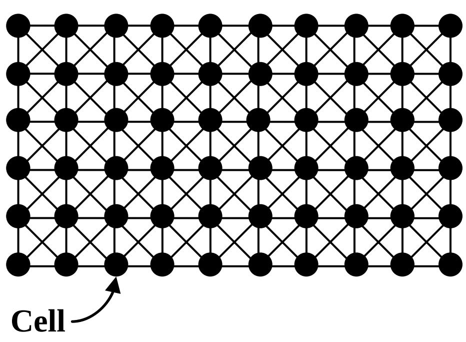

Fig. 2.1 A two-dimensional 6 $\times$ 10 cell array.

Let us consider one $M \times N$ two-dimensional cell array. There are total $M \times N$ cells arranged in M rows and N columns. The cell in ith row and jth column is labeled C_{ij}. The range of neighboring cells is called neighborhood. The number of neighboring cells of every cell is the same, it is determined by neighborhood radius, and this radius is different from the general round radius. The neighborhood of a cell is defined as the following.

$$N_{ij}(r) = \{ C_{kl} \mid \max(|k-i|, |l-j|) \le r, 1 \le k \le M; 1 \le l \le N \} \qquad (2\text{-}1)$$

$N_{ij}(r)$ represents the set of neighboring cells C_{kl} of cell C_{ij}, r is the radius of $N_{ij}(r)$, r is a positive integer. $N_{ij}(r)$ expresses one $(2r+1) \times (2r+1)$ cell array. For being simple and convenient, we are used to omitting r and express $N_{ij}(r)$ with N_{ij}. Such as $r = 1$, the range of cell C_{ij} and its neighboring cells is size of 3×3, as Fig. 2.2 shows [3]. The cell set contained by the grey square in Fig. 2.2 is $N_{ij}(1)$. $N_{ij}(1)$ represents a 3×3 cell array.

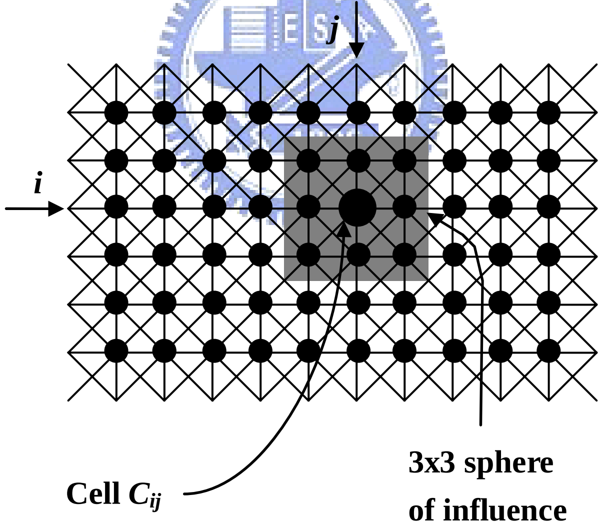

Fig. 2.2 Radius $r = 1$, the range of cell C_{ij} and its neighboring cells.

The relevant parameters of a cell are defined as follows and shown in Fig. 2.3 [3]:

u_{ij} : Input of cell C_{ij};

x_{ij} : State of cell C_{ij};

y_{ij} : Output of cell C_{ij};

I_{ij} : Threshold value of cell C_{ij}.

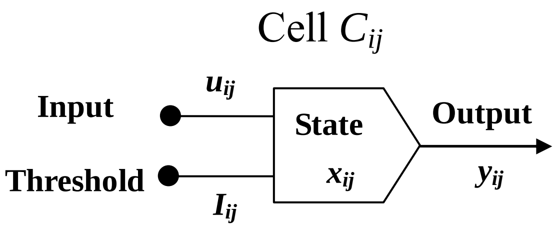

Fig. 2.3 The primary element of a cellular neural network - cell C_{ij}.

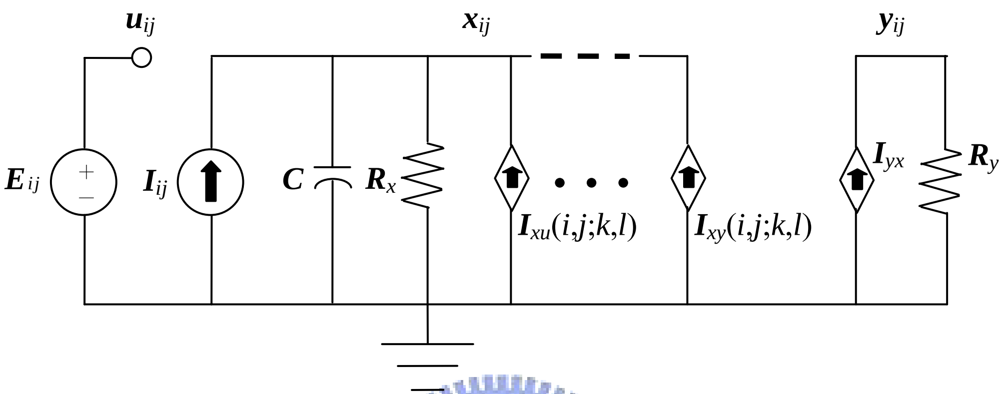

Fig. 2.4 The circuit structure of cell C_{ij}.

The basic circuit structure of a cell is shown in Fig. 2.4 [1]. According to this circuit structure, the motion equation of a cell C_{ij} can be written by

$$C\frac{dx_{ij}(t)}{dt} = -\frac{1}{R_x}x_{ij}(t) + \sum_{C_{kl}\in N_{ij}} A(i,j;k,l)y_{kl}(t) + \sum_{C_{kl}\in N_{ij}} B(i,j;k,l)u_{kl} + I_{ij} \qquad \text{(2-2)}$$
$$1 \le i \le M; 1 \le j \le N$$

For being simple and convenient and without losing generality, we can suppose $C = R_x = 1$, then Equation (2-2) can be rewritten to Equation (2-3):

$$\frac{dx_{ij}(t)}{dt} = -x_{ij}(t) + \sum_{C_{kl}\in N_{ij}} A(i,j;k,l)y_{kl}(t) + \sum_{C_{kl}\in N_{ij}} B(i,j;k,l)u_{kl} + I_{ij} \qquad \text{(2-3)}$$
$$1 \le i \le M;\ 1 \le j \le N$$

According to view of neural networks, the meaning of each parameter of Equation (2-3) is listed as follows:

x_{ij} : State of cell C_{ij};

y_{kl} : Output of neighboring cell C_{kl} of cell C_{ij};

u_{kl} : Input of neighboring cell C_{kl} of cell C_{ij};

I_{ij} : Threshold value of cell C_{ij};

$A(i, j; k, l)$: The weighting of output of neighboring cell C_{kl} of cell C_{ij};

$B(i, j; k, l)$: The weighting of input of neighboring cell C_{kl} of cell C_{ij}.

$A(i, j; k, l)$, $B(i, j; k, l)$ and I_{ij} are total $2\times(2r+1)^2+1$ values. They determine the behavior of a two-dimensional cellular neural network. We call that **A** = $[A(i, j; k, l)]$ is the feedback template and **B** = $[B(i, j; k, l)]$ is the control template. The output y_{ij} is a piecewise linear function of x_{ij}. It is given by the following output equation and shown in Fig. 2.5 [1]:

$$y_{ij}(t) = f\left(x_{ij}(t)\right) = \frac{1}{2}\left(\left|x_{ij}(t)+1\right| - \left|x_{ij}(t)-1\right|\right) \tag{2-4}$$

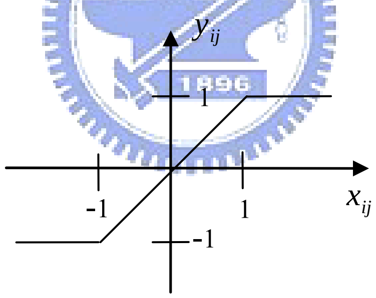

Fig. 2.5 Output function.

We can receive the following results from output function (2-4):

1. $\left|y_{ij}(t)\right| \le 1$

2. $\dfrac{dy_{ij}}{dx_{ij}} = \begin{cases} 1, & \left|x_{ij}\right| < 1 \\ 0, & \left|x_{ij}\right| \ge 1 \end{cases}$

3. If $\left|x_{ij}\right| < 1$, then $x_{ij} = y_{ij}$

Each cell is influenced by neighboring cells' inputs and outputs. Cells' inputs multiply template **B** and cell's outputs multiply template **A**, as Fig. 2.6 shows [3].

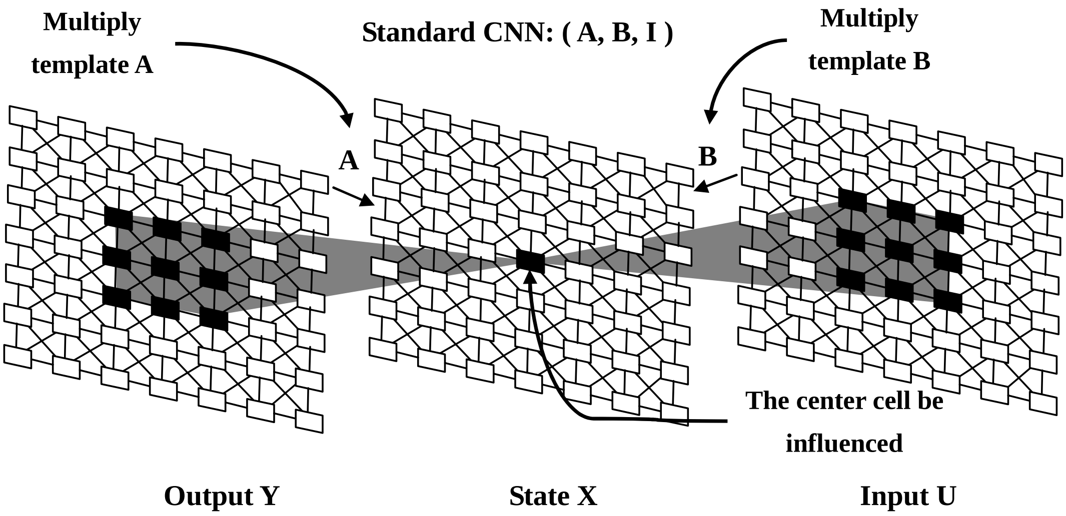

Fig. 2.6 The basic model of cellular neural networks.

The motion equation can be expressed by Fig. 2.7 [3]:

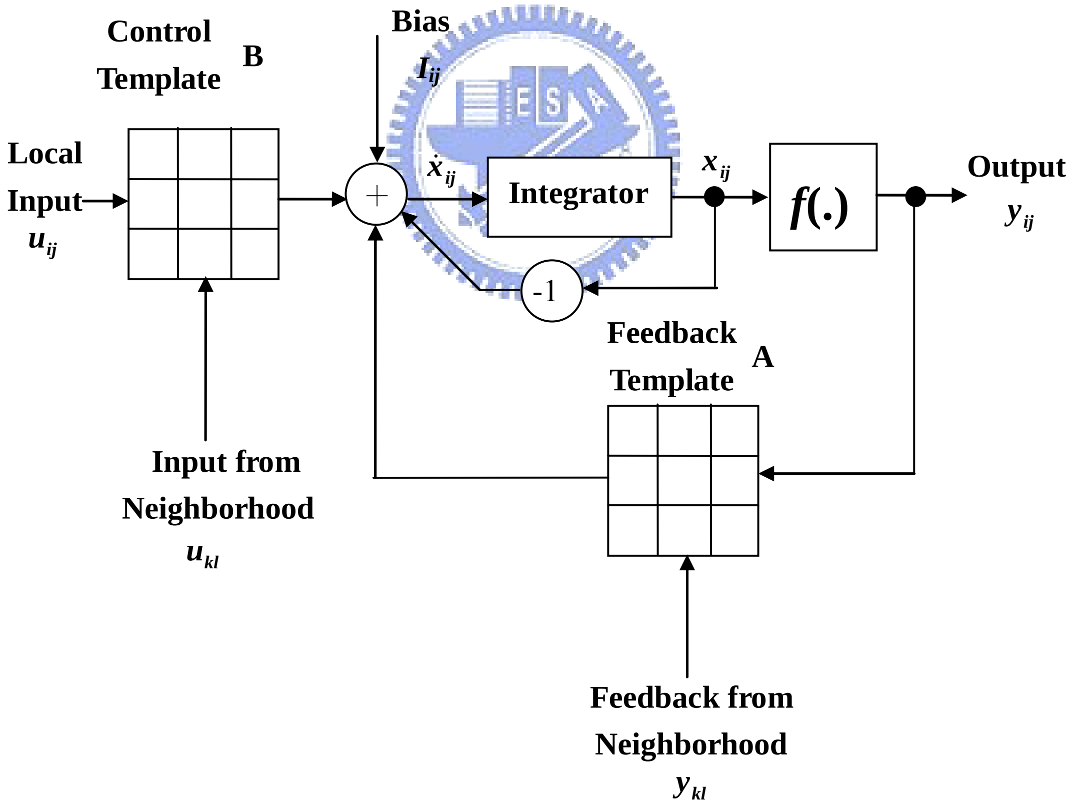

Fig. 2.7 Motion equation.

If $A(i, j; i, j) > 1$, then after the transient decayed to zero, each cell of a cellular neural network must stop in a stable equilibrium point. And the magnitude of each stable equilibrium point is greater than 1. In other words, we have the following characteristics:

$$\lim_{t\to\infty}|x_{ij}(t)| > 1, \quad 1 \le i \le M; \quad 1 \le j \le N \tag{2-5}$$

and

$$\lim_{t\to\infty} y_{ij}(t) = \pm 1, \quad 1 \le i \le M; \quad 1 \le j \le N \tag{2-6}$$

Namely if the center element $A(i, j; i, j)$ of template **A** satisfies $A(i, j; i, j) > 1$, then the absolute value of the output of each cell in stable equilibrium point is equal to 1. This guarantees that cellular neural networks have binary value output; this property is essential for solving classification problems in image processing applications.

2.3 Comparison of Cellular Neural Network and Hopfield Neural Network

It can be said that cellular neural networks are evolved from Hopfield neural network. Cellular neural networks are more general than Hopfield neural network. Because in cellular neural networks each cell is influenced by only near cells, it is not like Hopfield neural network, in which each cell is influenced by all other cells. So cellular neural networks can be easily realized by VLSI technique. The followings are some differences between cellular neural networks and Hopfield neural network:

(1) The weight matrix of Hopfield neural network must be symmetric, but the weight matrices **A** and **B** of a cellular neural network are not necessary to be symmetric.

(2) Hopfield neural network is allowed to operate asynchronously, but cellular neural networks must operate synchronously.

(3) The connections in the cellular neural network are local. Hopfield neural network is a fully connected neural network. In general, the number of interconnections in the cellular neural network is less than the number of interconnections in a Hopfield neural network.

(4) The self-feedback coefficient $A(i, j; i, j)$ of cell C_{ij} in a cellular neural network is greater than 1 in order to guarantee that the steady-state output of each cell is either +1 or -1. This condition is always violated in a Hopfield neural network since its diagonal coupling coefficients are all assumed to be zero [6].

Design of CNN System

The design of a CNN system is finding one or more templates that realize a certain input-output behavior. The design methods of templates found in literature can be divided into two classes:

Design by synthesis: given an explicit problem specification, a set of parameters is found that satisfies the specified requirements.

Design by learning: given a vague description of the task through a large number of input-target pairs, a learning process is applied that minimizes some kind of cost function.

Rather than being alternatives, these two types of design can be considered as supplementing methods. The choice between them is based on the type of information that is available about the problem at hand. In case an explicit description of the desired functional behavior is available, a synthesis method is used. In case only implicit knowledge is available, a learning method is the obvious choice.

2.4 Discrete-Time Cellular Neural Networks

Harrer and Nossek recommend that discrete time cellular neural networks (Discrete Time Cellular Neural Network, is abbreviated as DT-CNN) [8] is as the discrete time version of CNN that Chua and Yang introduced (is abbreviated as Chua−Yang-CNN here). The network dynamics of Chua−Yang-CNN is described by a set of differential equations:

$$\frac{dx_{ij}(t)}{dt} = -x_{ij}(t) + \sum_{C_{kl}\in N_{ij}} A(i,j;k,l)y_{kl}(t) + \sum_{C_{kl}\in N_{ij}} B(i,j;k,l)u_{kl}(t) + I_{ij}$$
$$1 \le i \le M;\ 1 \le j \le N$$

t in the previous equation is a discrete time variable, then we can rewrite the previous equation:

$$\frac{x_{ij}(h+\Delta t) - x_{ij}(h)}{\Delta t} = -x_{ij}(h) + \sum_{C_{kl}\in N_{ij}} A(i,j;k,l)y_{kl}(h) + \sum_{C_{kl}\in N_{ij}} B(i,j;k,l)u_{kl}(h) + I_{ij} \quad (2\text{-}7)$$
$$1 \le i \le M;\ 1 \le j \le N$$

If $\Delta t = 1$, the Equation (2-7) can be rewritten as the Equation (2-8):

$$x_{ij}(h+1) - x_{ij}(h) = -x_{ij}(h) + \sum_{C_{kl}\in N_{ij}} A(i,j;k,l)y_{kl}(h) + \sum_{C_{kl}\in N_{ij}} B(i,j;k,l)u_{kl}(h) + I_{ij}$$
$$x_{ij}(h+1) = \sum_{C_{kl}\in N_{ij}} A(i,j;k,l)y_{kl}(h) + \sum_{C_{kl}\in N_{ij}} B(i,j;k,l)u_{kl}(h) + I_{ij} \quad (2\text{-}8)$$
$$1 \le i \le M;\ 1 \le j \le N$$

DT-CNN is a timing system. A set of discrete motion equations describes its dynamic behavior. At discrete time *h*, the state $x_{ij}(h)$ of a cell C_{ij} depends on time-invariant input u_{kl} and time-variant output $y_{kl}(h-1)$ of its neighboring cell C_{kl}. The output function of DT-CNN is the hardlimiter activation function.

$$y_{ij}(h+1) = f(x_{ij}(h+1)) = \begin{cases} +1 & for \quad x_{ij}(h+1) \geq 0 \\ -1 & for \quad x_{ij}(h+1) < 0 \end{cases} \qquad (2\text{-}9)$$

$$1 \leq i \leq M;\ 1 \leq j \leq N$$

References

[1] L. O. Chua and Lin Yang, "Cellular Neural Networks: Theory," *IEEE Trans. on CAS*, vol. 35 no. 10, pp.1257-1272, 1988.

[2] L. O. Chua and Lin Yang, "Cellular Neural Networks: Applications," *IEEE Trans. on CAS*, vol.35, no.10, pp. 1273-1290, 1988.

[3] Leon O. Chua, CNN：A PARADIGM FOR COMPLEXITY. World Scientific, 1998.

[4] Special Issue on Chaotic Systems, *Proc. IEEE*, Aug. 1987.

[5] L. O. Chua and R. N. Madan, "The sights and sounds of chaos," *IEEE Circuits Devices Mag.*, pp. 3-13, Jan., 1988.

[6] J. J. Hopfield, "Neural networks and physical systems with emergent computational abilities," *Proc. Natl. Acad. Sci. USA.*, vol. 79, pp. 2554-2558, 1982.

[7] 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.

[8] H. Harrer and J. A. Nossek, "Discrete-time Cellular Neural Networks," *International Journal of Circuit Theory and Applications*, Vol. 20, pp. 453-468, 1992.