Cellular Neural Networks: Theory

LEON O. CHUA, FELLOW, IEEE, AND LIN YANG, STUDENT MEMBER, IEEE

Abstract — A novel class of information-processing systems called *cellular neural networks* is proposed. Like a *neural network*, it is a large-scale nonlinear analog circuit which processes signals in *real* time. Like *cellular automata*, it is made of a massive aggregate of regularly spaced circuit clones, called *cells*, which communicate with each other directly only through its nearest neighbors. Each cell is made of a *linear* capacitor, a *nonlinear* voltage-controlled current source, and a few *resistive* linear circuit elements.

Cellular neural networks share the best features of both worlds; its continuous time feature allows *real-time* signal processing found wanting in the digital domain and its local interconnection feature makes it tailor made for VLSI implementation.

Cellular neural networks are uniquely suited for *high-speed parallel* signal processing. Some impressive applications of cellular neural networks to image processing is presented in a companion paper [1].

I. INTRODUCTION

ANALOG CIRCUITS have played a very important role in the development of modern electronic technology. Even in our digital computer era, analog circuits still dominate such fields as communications, power, automatic control, audio and video electronics because of their *real-time* signal processing capabilities.

Conventional digital computation methods have run into a serious speed bottleneck due to their serial nature. To overcome this problem, a new computation model, called "neural networks," has been proposed, which is based on some aspects of neurobiology and adapted to integrated circuits [2]–[4]. The key features of neural networks are asynchronous parallel processing, continuous-time dynamics, and global interaction of network elements. Some encouraging if not impressive applications of neural networks have been proposed for various fields such as optimization, linear and nonlinear programming, associative memory, pattern recognition and computer vision [5]–[12].

In this paper, we will present a new circuit architecture, called a *cellular neural network*, which possesses some of the key features of neural networks and which has important potential applications in such areas as image processing and pattern recognition. This architecture is presented in *Section II*. An in-depth analysis of cellular neural networks then follows. In particular, the practical question of *dynamic range* is derived in *Section III* and a
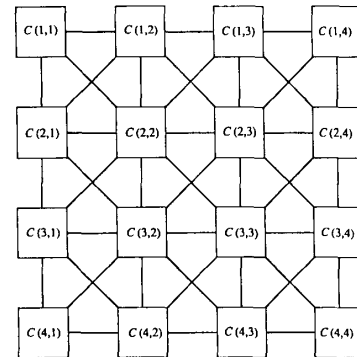
Fig. 1. A two-dimensional cellular neural network. The circuit size is 4×4. The squares are the circuit units called *cells*. The links between the *cells* indicate that there are interactions between the linked *cells*.

stability analysis is presented in *Section IV*. Computer simulations and typical dynamic behaviors of a simple cellular neural network will be discussed in *Section V*. All of these results, though given only for single-layer cellular neural networks, is generalized, *mutatis mutandis*, to *multi-layered* cellular neural networks in *Section VI*. Finally, two similar mathematical models are compared with our cellular neural network in *Section VII*.

II. ARCHITECTURE OF CELLULAR NEURAL NETWORKS

The basic circuit unit of cellular neural networks is called a *cell*. It contains linear and nonlinear circuit elements, which typically are linear capacitors, linear resistors, linear and nonlinear controlled sources, and independent sources. The structure of cellular neural networks is similar to that found in cellular automata; namely, any cell in a cellular neural network is connected only to its neighbor cells. The adjacent cells can interact directly with each other. Cells not directly connected together may affect each other indirectly because of the propagation effects of the *continuous*-time dynamics of cellular neural networks. An example of a two-dimensional cellular neural network is shown in Fig. 1. Theoretically, we can define a cellular neural network of any dimension, but in this paper, we will concentrate on the two-dimensional case because we will focus our attention on image processing problems. The results can be easily generalized to higher dimension cases.

Consider an $M \times N$ cellular neural network, having $M \times N$ cells arranged in M rows and N columns. We call

$r = 1$ $r = 2$ $r = 3$
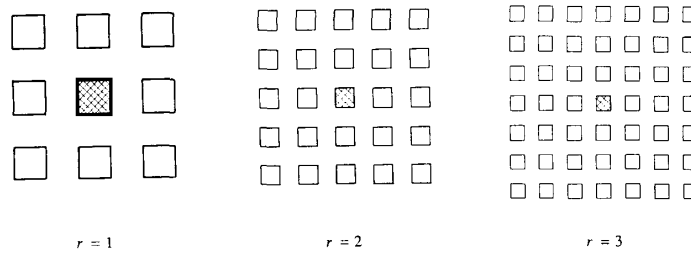
Fig. 2.   The neighborhood of cell $C(i, j)$ defined by (1) for $r = 1$, $r = 2$ and $r = 3$, respectively.

$I_{xu}(i,j;k,l) = B(i,j;k,l)v_{ukl}$, $I_{xy}(i,j;k,l) = A(i,j;k,l)v_{ykl}$, $I_{yx} = \frac{1}{2R_y}(|v_{xij} + 1| - |v_{xij} - 1|)$.
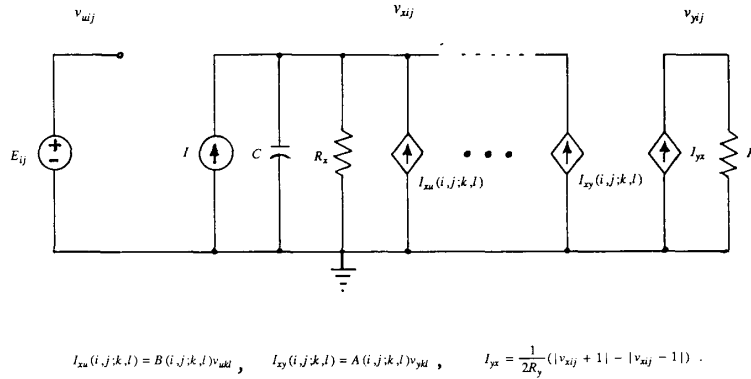
Fig. 3.   An example of a *cell* circuit. C is a linear capacitor; R_x, R_u, and R_y are linear resistors; I is an independent voltage source; $I_{xu}(i, j; k, l)$ and $I_{xy}(i, j; k, l)$ are linear voltage-controlled current sources with the characteristics $I_{xy}(i, j; k, l) = A(i, j; k, l)v_{ykl}$ and $I_{xu}(i, j; k, l) = B(i, j; k, l)v_{ukl}$ for all $C(i, j) \in N(i, j)$; $I_{yx} = (1/R_y)f(v_{xij})$ is a piecewise-linear voltage-controlled current source with its characteristic $f(\cdot)$ as shown in Fig. 4; E_{ij} is an independent voltage source.

the cell on the ith row and the jth column *cell* (i, j), and denote it by $C(i, j)$ as in Fig. 1. Now let us define what we mean by a *neighborhood* of $C(i, j)$.

Definition 1: r-neighborhood

The *r-neighborhood* of a cell $C(i, j)$, in a cellular neural network is defined by

$$N_r(i, j) = \{ C(k, l) | \max\{|k - i|, |l - j|\} \leq r,$$
$$1 \leq k \leq M; 1 \leq l \leq N \} \quad (1)$$

where r is a positive integer number.[1]

Fig. 2 shows 3 neighborhoods of the same cell (located at the center and shown shaded) with $r = 1$, 2 and 3, respectively. Usually, we call the $r = 1$ neighborhood a "3×3 neighborhood," the $r = 2$ neighborhood a "5×5 neighborhood," and the $r = 3$ neighborhood a "7×7 neighborhood." It is easy to show that the neighborhood system defined above exhibits a *symmetry* property in the sense that if $C(i, j) \in N_r(k, l)$, then $C(k, l) \in N_r(i, j)$, for all $C(i, j)$ and $C(k, l)$ in a cellular neural network.

A typical example of a cell $C(i, j)$ of a cellular neural network is shown in Fig. 3, where the suffices u, x, and y denote the *input*, *state*, and *output*, respectively. The node voltage v_{xij} of $C(i, j)$ is called the state of the cell and its

initial condition is assumed to have a magnitude less than or equal to 1. The node voltage v_{uij} is called the input of $C(i, j)$ and is assumed to be a constant with magnitude less than or equal to 1. The node voltage v_{yij} is called the output.

Observe from Fig. 3 that each cell $C(i, j)$ contains one independent voltage source E_{ij}, one independent current source I, one linear capacitor C, two linear resistors R_x and R_y, and at most $2m$ *linear* voltage-controlled current sources which are *coupled* to its neighbor cells via the *controlling input* voltage v_{ukl}, and the *feedback* from the *output* voltage v_{ykl} of each neighbor cell $C(k, l)$, where m is equal to the number of neighbor cells. In particular, $I_{xy}(i, j; k, l)$ and $I_{xu}(i, j; k, l)$ are linear voltage controlled current sources with the characteristics $I_{xy}(i, j; k, l) = A(i, j; k, l)v_{ykl}$ and $I_{xu}(i, j; k, l) = B(i, j; k, l)v_{ukl}$ for all $C(k, l) \in N_r(i, j)$. The *only nonlinear* element in each cell is a piecewise-linear voltage-controlled current source $I_{yx} = (1/R_y)f(v_{xij})$ with characteristic $f(\cdot)$ as shown in Fig. 4.

All of the linear and piecewise-linear controlled sources used in our cellular neural networks can be easily realized using operational amplifiers (op amps) [13], [14]. A simple example of an op amp implementation of a cell circuit is given in the Appendix. Without loss of generality, the cell circuit architecture in Fig. 3 will be used throughout this paper.

[1] To avoid clutter, we will often suppress the subscript r.

Applying KCL and KVL, the circuit equations of a cell are easily derived as follows:

State equation:

$$C\frac{dv_{xij}(t)}{dt} = -\frac{1}{R_x}v_{xij}(t)$$

$$+ \sum_{C(k,l)\in N_r(i,j)} A(i,j;k,l)v_{ykl}(t)$$

$$+ \sum_{C(k,l)\in N_r(i,j)} B(i,j;k,l)v_{ukl} + I,$$

$$1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N. \quad (2a)$$

Output equation:

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

$$1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N. \quad (2b)$$

Input equation:

$$v_{uij} = E_{ij}, \qquad 1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N. \quad (2c)$$

Constraint conditions:[2]

$$|v_{xij}(0)| \leqslant 1, \qquad 1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N \quad (2d)$$

$$|v_{uij}| \leqslant 1, \qquad 1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N. \quad (2e)$$

Parameter assumptions:

$$A(i,j;k,l) = A(k,l;i,j),$$

$$1 \leqslant i,k \leqslant M; 1 \leqslant j,l \leqslant N. \quad (2f)$$

$$C > 0, R_x > 0. \quad (2g)$$

Remarks:

(a) All *inner cells* of a cellular neural network have the same circuit structures and element values. The *inner cell* is the cell which has $(2r+1)^2$ neighbor cells, where r is defined in (1). All other cells are called *boundary cells*. A cellular neural network is completely characterized by the set of all nonlinear differential equations (2) associated with the cells in the circuit.

(b) Each cell of a cellular neural network has at most three nodes. (Sometimes we will choose $E_{ij} = 0$ if $B(i,j;k;l) = 0$ for all cells in a cellular neural network. In this case, there are only two nodes in a cell

circuit.) Since all cells have the same datum node, and since all circuit elements are voltage controlled, our cellular neural networks are ideally suited for nodal analysis. Moreover, since the interconnections
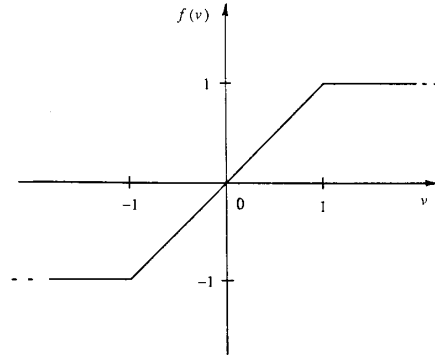
$$f(v) = \frac{1}{2}\big[\ |v+1| - |v-1|\ \big]$$

Fig. 4. The characteristic of the nonlinear controlled source.

are *local*, the associated matrix node equation is extremely sparse for large circuits.

(c) The dynamics of a cellular neural network has both output feedback and input control mechanisms. The output feedback effect depends on the interactive parameter $A(i,j;k,l)$ and the input control effect depends on $B(i,j;k,l)$. Consequently, it is sometimes instructive to refer to $A(i,j;k,l)$ as a *feedback operator* and $B(i,j;k,l)$ as a *control operator*. The assumptions in (2f) are reasonable because of the symmetry property of the neighborhood system.

(d) The values of the circuit elements can be chosen conveniently in practice. R_x and R_y determine the power dissipated in the circuits and are usually chosen to be between 1 kΩ and 1 MΩ. CR_x is the time constant of the dynamics of the circuit and is usually chosen to be 10^{-8}–10^{-5} s.

III. DYNAMIC RANGE OF CELLULAR NEURAL NETWORKS

Before we design a physical cellular neural network, it is necessary to know its dynamic range in order to guarantee that it will satisfy our assumptions on the dynamical equations stipulated in the preceding section. The following theorem provides the foundation for our design.

Theorem 1

All *states* v_{xij} in a cellular neural network are bounded for all time $t > 0$ and the bound v_{max} can be computed by the following formula for *any* cellular neural network:

$$v_{max} = 1 + R_x|I| + R_x \max_{1 \leqslant i \leqslant M, 1 \leqslant j \leqslant N}\left[\sum_{C(k,l)\in N_r(i,j)} \big(|A(i,j;k,l)| + |B(i,j;k,l)|\big)\right]. \quad (3)$$

Proof: First, let us recast the *cell* dynamical equation (2) as

$$\frac{dv_{xij}(t)}{dt} = -\frac{1}{R_x C}v_{xij}(t) + f_{ij}(t) + g_{ij}(u) + \hat{I},$$

$$1 \leqslant i \leqslant M, 1 \leqslant j \leqslant N, \quad (4a)$$

where

$$f_{ij}(t) = \frac{1}{C} \sum_{C(k,l) \in N_r(i,j)} A(i,j;k,l) v_{ykl}(t),$$

$$1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N \quad \text{(4b)}$$

$$g_{ij}(u) = \frac{1}{C} \sum_{C(k,l) \in N_r(i,j)} B(i,j;k,l) v_{ukl},$$

$$1 \leqslant i \leqslant M, 1 \leqslant j \leqslant N \quad \text{(4c)}$$

and

$$\hat{I} = \frac{I}{C} \quad \text{(4d)}$$

where $u \equiv [E_{ij}]_{1 \times MN}$ indicates an MN-dimensional constant input vector. Equation (4a) is a first-order ordinary differential equation and its solution is given by

$$v_{xij}(t) = v_{xij}(0) e^{-t/R_x C}$$
$$+ \int_0^t e^{-(t-\tau)/R_x C} \left[f_{ij}(\tau) + g_{ij}(u) + \hat{I} \right] d\tau. \quad \text{(5)}$$

It follows that

$$|v_{xij}(t)| \leqslant |v_{xij}(0) e^{-t/R_x C}|$$
$$+ \left| \int_0^t e^{-(t-\tau)/R_x C} \left[f_{ij}(\tau) + g_{ij}(u) + \hat{I} \right] d\tau \right|$$
$$\leqslant |v_{xij}(0)| |e^{-t/R_x C}$$
$$+ \int_0^t e^{-(t-\tau)/R_x C} \left[|f_{ij}(\tau)| + |g_{ij}(u)| + |\hat{I}| \right] d\tau$$
$$\leqslant |v_{xij}(0)| |e^{-t/R_x C}$$
$$+ \left[F_{ij} + G_{ij} + |\hat{I}| \right] \int_0^t e^{-(t-\tau)/R_x C} d\tau$$
$$\leqslant |v_{xij}(0)| + R_x C \left[F_{ij} + G_{ij} + |\hat{I}| \right] \quad \text{(6)}$$

where

$$F_{ij} \equiv \max_t |f_{ij}(t)|$$
$$\leqslant \frac{1}{C} \sum_{C(k,l) \in N_r(i,j)} |A(i,j;k,l)| \max_t |v_{ykl}(t)| \quad \text{(7a)}$$

and

$$G_{ij} \equiv \max_u |g_{ij}(u)|$$
$$\leqslant \frac{1}{C} \sum_{C(k,l) \in N_r(i,j)} |B(i,j;k,l)| \max_u |v_{ukl}|. \quad \text{(7b)}$$

Since $|v_{xij}(0)|$ and $|v_{uij}|$ satisfy the conditions in (2d) and (2e), and since $|v_{yij}(t)|$ satisfies the condition

$$|v_{yij}(t)| \leqslant 1, \quad \text{for all } t \quad \text{(8)}$$

in view of its characteristic function (2b), it follows from

(6) and (7) that

$$|v_{xij}(t)| \leqslant |v_{xij}(0)|$$
$$+ R_x \left[\sum_{C(k,l) \in N_r(i,j)} \left(|A(i,j;k,l)| \max_t |v_{ykl}(t)| \right. \right.$$
$$\left. \left. + |B(i,j;k,l)| \max_u |v_{ukl}| \right) + |I| \right]$$
$$\leqslant 1 + R_x \left[\sum_{C(k,l) \in N_r(i,j)} \left(|A(i,j;k,l)| \right. \right.$$
$$\left. \left. + |B(i,j;k,l)| \right) + |I| \right],$$

$$1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N. \quad \text{(9)}$$

Now let

$$v_{max} = \max_{(i,j)} \left\{ 1 + R_x |I| + R_x \sum_{C(k,l) \in N_r(i,j)} \left(|A(i,j;k,l)| \right. \right.$$
$$\left. \left. + |B(i,j;k,l)| \right) \right\} \quad \text{(10)}$$

then since v_{max} is independent of the time t and the cell $C(i, j)$ for all i and j, we have

$$\max_t |v_{xij}| \leqslant v_{max}, \quad \text{for all } 1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N. \quad \text{(11)}$$

For any cellular neural network, the parameters R_x, C, I, $A(i,j;k,l)$ and $B(i,j;k,l)$ are finite constants, therefore the bound on the states of the cells, v_{max}, is finite and can be computed via formula (3). $\square$

Remark:

In actual circuit design, it is convenient to choose the scale of the circuit parameters such that $R_x |I| \approx 1$, $R_x |A(i,j;k,l)| \approx 1$ and $R_x |B(i,j;k,l)| \approx 1$, for all i, j, k, and l. Hence, we can easily estimate the upper bound on the dynamic range of our cellular neural networks. For example, if a neighborhood of the cellular neural network is 3×3, then we can have $v_{max} \approx 20$ V, which is within the typical power supply voltage range of IC circuits.

IV. STABILITY OF CELLULAR NEURAL NETWORKS

One application of cellular neural networks is in image processing, which we present in a companion paper [1]. The basic function of a cellular neural network for image processing is to map or transform an input image into a corresponding output image. Here, we restrict our output images to binary images with -1 and 1 as the pixel values. However, the input images can have multiple gray levels, provided that their corresponding voltages satisfy (2e). This means that our image processing cellular neural network must always converge to a constant steady state following any transient regime which has been initialized and/or driven by a given input image. How can we guarantee the convergence of cellular neural networks?

What are the conditions or restrictions for such convergence to be possible? In this section, we will discuss the convergence property and its related problems for cellular neural networks.

One of the most effective techniques for analyzing the convergence properties of dynamic nonlinear circuits is *Lyapunov's method*. Hence, let us first define a Lyapunov function for cellular neural networks.

Definition 2:

We define the Lyapunov function, $E(t)$, of a cellular neural network by the scalar function

$$E(t) = -\frac{1}{2} \sum_{(i,j)} \sum_{(k,l)} A(i,j;k,l) v_{yij}(t) v_{ykl}(t)$$

$$+ \frac{1}{2R_x} \sum_{(i,j)} v_{yij}(t)^2$$

$$- \sum_{(i,j)} \sum_{(k,l)} B(i,j;k,l) v_{yij}(t) v_{ukl} - \sum_{(i,j)} I v_{yij}(t).$$

$$\tag{12}$$

Remarks:

(a) Observe that the above Lyapunov function, $E(t)$, is only a function of the input, v_u, and output, v_y, voltages of the circuit. Although it does not possess the complete information contained in the state variables v_{xij}, we can nevertheless derive the steady-state properties of the state variables from the properties of $E(t)$.

(b) The Lyapunov function, $E(t)$, defined above, can be interpreted as the "generalized energy" of a cellular neural network, although its exact physical meaning is not very clear. As the following theorems will show, $E(t)$ always converges to a local minimum, where the cellular neural network produces the desired output.

In the following theorem, we will prove that $E(t)$ is bounded.

Theorem 2
The function $E(t)$ defined in (12) is bounded by

$$\max_t |E(t)| \leqslant E_{\max} \tag{13a}$$

where

$$E_{\max} = \frac{1}{2} \sum_{(i,j)} \sum_{(k,l)} |A(i,j;k,l)|$$

$$+ \sum_{(i,j)} \sum_{(k,l)} |B(i,j;k,l)|$$

$$+ MN \left(\frac{1}{2R_x} + |I| \right) \tag{13b}$$

for an $M \times N$ cellular neural network.

Proof: From the definition of $E(t)$, we have

$$|E(t)| \leqslant \frac{1}{2} \sum_{(i,j)} \sum_{(k,l)} |A(i,j;k,l)| |v_{yij}(t)| |v_{ykl}(t)|$$

$$+ \frac{1}{2R_x} \sum_{(i,j)} v_{yij}^2$$

$$+ \sum_{(i,j)} \sum_{(k,l)} |B(i,j;k,l)| |v_{yij}(t)| |v_{ukl}|$$

$$+ \sum_{(i,j)} |I| |v_{yij}|. \tag{14}$$

Since $v_{yij}(t)$ and v_{uij} are bounded as stipulated in (2d) and (2e), we have

$$|E(t)| \leqslant \frac{1}{2} \sum_{(i,j)} \sum_{(k,l)} |A(i,j;k,l)| + MN \frac{1}{2R_x}$$

$$+ \sum_{(i,j)} \sum_{(k,l)} |B(i,j;k,l)| + MN|I|. \tag{15}$$

It follows from (13b) and (15) that $E(t)$ is bounded as claimed in (13a).□

Not only can we show that $E(t)$ is bounded, but we can also prove that it is a monotone decreasing function.

Theorem 3
The scalar function $E(t)$ defined in (12) is a monotone-decreasing function, that is

$$\frac{dE(t)}{dt} \leqslant 0. \tag{16}$$

Proof: To differentiate $E(t)$ in (12) with respect to time t, take the derivative of $v_{yij}(t)$ on the right side of (12) with respect to $v_{xij}(t)$, and then differentiate $v_{xij}(t)$ with respect to time t:

$$\frac{dE(t)}{dt} = - \sum_{(i,j)} \sum_{(k,l)} A(i,j;k,l) \frac{dv_{yij}}{dv_{xij}} \frac{dv_{xij}(t)}{dt} v_{ykl}(t)$$

$$+ \frac{1}{R_x} \sum_{(i,j)} \frac{dv_{yij}}{dv_{xij}} \frac{dv_{xij}(t)}{dt} v_{yij}(t)$$

$$- \sum_{(i,j)} \sum_{(k,l)} B(i,j;k,l) \frac{dv_{yij}}{dv_{xij}} \frac{dv_{xij}(t)}{dt} v_{ukl}$$

$$- \sum_{(i,j)} I \frac{dv_{yij}}{dv_{xij}} \frac{dv_{xij}(t)}{dt}. \tag{17}$$

Here we have used the symmetry assumption (2f).

From the output functions in (2b), we obtain the following relations

$$\frac{dv_{yij}}{dv_{xij}} = \begin{cases} 1, & |v_{xij}| < 1 \\ 0, & |v_{xij}| \geqslant 1 \end{cases} \tag{18a}$$

and

$$v_{xij} = v_{yij}, \quad |v_{xij}| < 1. \tag{18b}$$

(Here, we define $(dv_{yij}/dv_{xij}) = 0$, for $|v_{xij}| = 1$.) And

according to our definition of cellular neural networks, we have

$$A(i, j; k, l) = 0, \quad B(i, j; k, l) = 0,$$
$$\text{for } C(k, l) \notin N_r(i, j). \quad (18c)$$

It follows from (17) and (18) that

$$\frac{dE(t)}{dt} = -\sum_{(i, j)} \frac{dv_{yij}}{dv_{xij}} \frac{dv_{xij}(t)}{dt}$$

$$\cdot \left[\sum_{C(k, l) \in N_r(i, j)} A(i, j; k, l) v_{ykl}(t) - \frac{1}{R_x} v_{yij}(t) \right.$$

$$\left. + \sum_{C(k, l) \in N_r(i, j)} B(i, j; k, l) v_{ukl} + I \right]$$

$$= -\sum_{|v_{xij}| < 1} \frac{dv_{xij}(t)}{dt}$$

$$\cdot \left[\sum_{C(k, l) \in N_r(i, j)} A(i, j; k, l) v_{ykl}(t) - \frac{1}{R_x} v_{xij}(t) \right.$$

$$\left. + \sum_{C(k, l) \in N_r(i, j)} B(i, j; k, l) v_{ukl} + I \right] \quad (19)$$

Substituting the cell circuit equation (2) into (19), and recalling $C > 0$ in assumption (2g), we obtain

$$\frac{dE(t)}{dt} = -\sum_{|v_{xij}| < 1} C \left[\frac{dv_{xij}(t)}{dt} \right]^2 \leqslant 0. \quad \square \quad (20a)$$

Remarks:

(1) For future analysis (e.g., corollary to Theorem 4), it is convenient to rewrite (20a) as follows:

$$\frac{dE(t)}{dt} = -\sum_{|v_{xij}| < 1} C \left[\frac{dv_{xij}(t)}{dt} \right]^2$$

$$= -\sum_{|v_{yij}| < 1} C \left[\frac{dv_{yij}(t)}{dt} \right]^2$$

$$= -\sum_{(i, j)} C \left[\frac{dv_{yij}(t)}{dt} \right]^2$$

$$\leqslant 0. \quad (20b)$$

(2) In the above proof, we have assumed that $v_{yij} = f(v_{xij})$ is a piecewise-linear function and have defined its derivative at the break points, $|v_{xij}| = 1$. In any hardware (e.g., VLSI) inplementation, $v_{yij} = f(v_{xij})$ is a smooth function in the sense that it is continuously differentiable. Fortunately, *Theorem* 3 can be proved to hold for any *sigmoid* function[3] $v_{yij} = f(v_{xij})$, which satisfies the condition $(dv_{yij}/dv_{xij}) \geqslant 0$. To prove this, simply replace the

---

[3] The sigmoid function $f(x)$ is defined by the properties $|f(x)| \leqslant M$ and $(df(x)/dx) \geqslant 0$, where M is a constant.

---

second expression in the Lyapunov function $E(t)$ in (12) by

$$\frac{1}{R_x} \sum_{(i, j)} \int_0^{v_{yij}(t)} f^{-1}(v) \, dv \quad (20c)$$

and then differentiating $E(t)$ directly to obtain

$$\frac{dE(t)}{dt} = -\sum_{(i, j)} C \frac{dv_{yij}}{dv_{xij}} \left[\frac{dv_{xij}(t)}{dt} \right]^2 \leqslant 0. \quad (20d)$$

This Lyapunov function is similar to the one used by Hopfield in [26] and can be interpreted as the total *co-content* function of an associated nonlinear resistive circuit [27].

From *Theorems* 2 and *3*, we can easily prove the following important result:

Theorem 4

For any given input v_u and any initial state v_x of a cellular neural network, we have

$$\lim_{t \to \infty} E(t) = constant \quad (21a)$$

and

$$\lim_{t \to \infty} \frac{dE(t)}{dt} = 0. \quad (21b)$$

Proof: From *Theorems* 2 and *3*, $E(t)$ is a bounded monotone decreasing function of time t. Hence $E(t)$ converges to a limit and its derivative converges to 0.$\square$

Corollary

After the transient of a cellular neural network has decayed to zero, we always obtain a constant dc output. In other words, we have

$$\lim_{t \to \infty} v_{yij}(t) = constant, \quad 1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N \quad (21c)$$

or

$$\lim_{t \to \infty} \frac{dv_{yij}(t)}{dt} = 0, \quad 1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N. \quad (21d)$$

Let us investigate next the steady-state behavior of cellular neural networks. It follows from the proof of *Theorem* 3 that under the condition $(dE(t)/dt) = 0$, there are three possible cases for the state of a cell as t tends to infinity:

(1) $\dfrac{dv_{xij}(t)}{dt} = 0$ and $|v_{xij}(t)| < 1$ (22a)

(2) $\dfrac{dv_{xij}(t)}{dt} = 0$ and $|v_{xij}(t)| \geqslant 1$ (22b)

(3) $\dfrac{dv_{xij}(t)}{dt} \neq 0$ and $|v_{xij}(t)| \geqslant 1$ (22c)

because of the characteristic of the piecewise-linear output function (2b).

This will be clear if we consider Fig. 4. When $|v_{xij}(t)| < 1$, we have $v_{yij}(t) = v_{xij}(t)$ and therefore $(dv_{yij}(t)/dt) = (dv_{xij}(t)/dt)$. From *Theorem* 4 and its corollary, case (1) follows. But for $|v_{xij}(t)| > 1$, since $v_{yij}(t) \neq v_{xij}(t)$, where

Fig. 5. The characteristic of the nonlinear resistor in the equivalent *cell* circuit.

Fig. 6. The steady-state equivalent circuit of a *cell* in a cellular neural network.

$v_{yij}(t) = \pm 1$ is a constant, we do not have the precise waveform of $v_{xij}(t)$. In the case where $v_{xij}(t) = constant$, we have case (2). Otherwise, case (3) applies and $v_{xij}(t)$ may be a periodic or an aperiodic but bounded function of time, in view of *Theorem 1*.

Is it possible for all three cases to co-exist among the different state variables when a cellular neural network is in its steady state, or can only one or two of three cases exist? We claim that only case (2) can exist for *all* v_{xij} in the *steady state* under a mild assumption on the range of the circuit parameters $A(i, j; i, j)$ and R_x.

To prove this claim, let us rewrite cell equation (2) as follows:

$$C\frac{dv_{xij}(t)}{dt} = -f\big(v_{xij}(t)\big) + g(t) \qquad (23a)$$

where

$$f\big(v_{xij}(t)\big) = -0.5A(i, j; i, j)\big(\big|v_{xij}(t)+1\big| - \big|v_{xij}(t)-1\big|\big)$$
$$+ \frac{1}{R_x}v_{xij}(t) \qquad (23b)$$

and

$$g(t) = \sum_{\substack{C(k, l) \in N_r(i, j), \\ C(k, l) \neq C(i, j)}} \big(A(i, j; k, l)v_{ykl}(t)$$
$$+ B(i, j; k, l)v_{ukl}\big) + I. \qquad (23c)$$

Let us first make some restrictions on the function $f(\cdot)$ in (23b). Suppose that $A(i, j; i, j) > 1/R_x$; for convenience and without loss of generality, let $A(i, j; i, j) = 2$, $R_x = 1$, and $C = 1$, in the following analysis. Then $f(v_{xij})$ has the characteristic shown in Fig. 5.

Consider next the equivalent circuit of a cell in a cellular neural network as shown in Fig. 6. There are only three circuit elements: a linear capacitor with a positive capacitance C, a piecewise-linear voltage controlled resistor with its driving point characteristic $i_R = f(v_r)$ ($f(\cdot)$ is the same function as in Fig. 5), and a time-varying independent current source whose output is given by $g(t)$. The two circuits in Figs. 3 and 6 are equivalent because they are
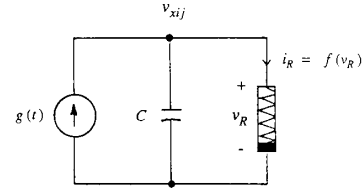
both described by (23a), which we rewrite for simplicity as follows:

$$\frac{dv(t)}{dt} = -f(v(t)) + g(t) \equiv \hat{f}(v, t). \qquad (24)$$

For $g(t) = 0$, the *equilibrium points* and the *dynamic route* [13], [14] of the equivalent circuit are shown in Fig. 7a. There are three *equilibrium points* in this circuit, one of them, $v = 0$, denoted by a circle is *unstable*; the other two, $v = -2$ and $v = 2$, are *stable*, and are denoted by solid points. The unstable equilibrium point is never observed in physical electronic circuits, because of unavoidable thermal noise. Therefore, after the transient, and depending on the initial state, the circuit will always approach one of its *stable equilibrium points* and stay there thereafter. For example, if the initial state of the circuit is $v = 0.5$, then the steady state will be observed at the *stable equilibrium point* $v = 2$; but if the initial state of the circuit is $v = -0.5$, then the steady state will be observed at the *stable equilibrium point* $v = -2$.

If $g(t) = constant \neq 0$, there are six different cases of the dynamic behavior of the equivalent circuit as shown in Fig. 7(b)–(g). For the cases in Fig. 7(b) and (c), there are also three *equilibrium points*; one of them is *unstable*, while the other two are *stable*. For the cases in Fig. 7(d) and (e), there are two *equilibrium points*; one is *unstable* and the other is *stable*. For the dynamic route in Fig. 7(f) and (g), there is only one *equilibrium point* for the circuit, and it is stable. Observe that all of the *stable equilibrium points* corresponding to the seven *dynamic routes* associated with the equivalent circuit of a *cell* in a cellular neural network share the common property $|v| > 1$.

Let us return now to the basic cell circuit of our cellular neural networks. Since $g(t)$ is a function of only the outputs, $v_{ykl}(t)$, and the inputs, v_{ukl}, of the neighborhood of the cell, it follows from the results of *Theorem 4* that all of the steady-state outputs of our cellular neural network are constants. Hence, after the initial transients our assumption $g(t) = constant$ is valid for the study of the steady-state behavior of cellular neural networks. Let us summarize our above observation as follows:

Theorem 5

If the circuit parameters satisfy
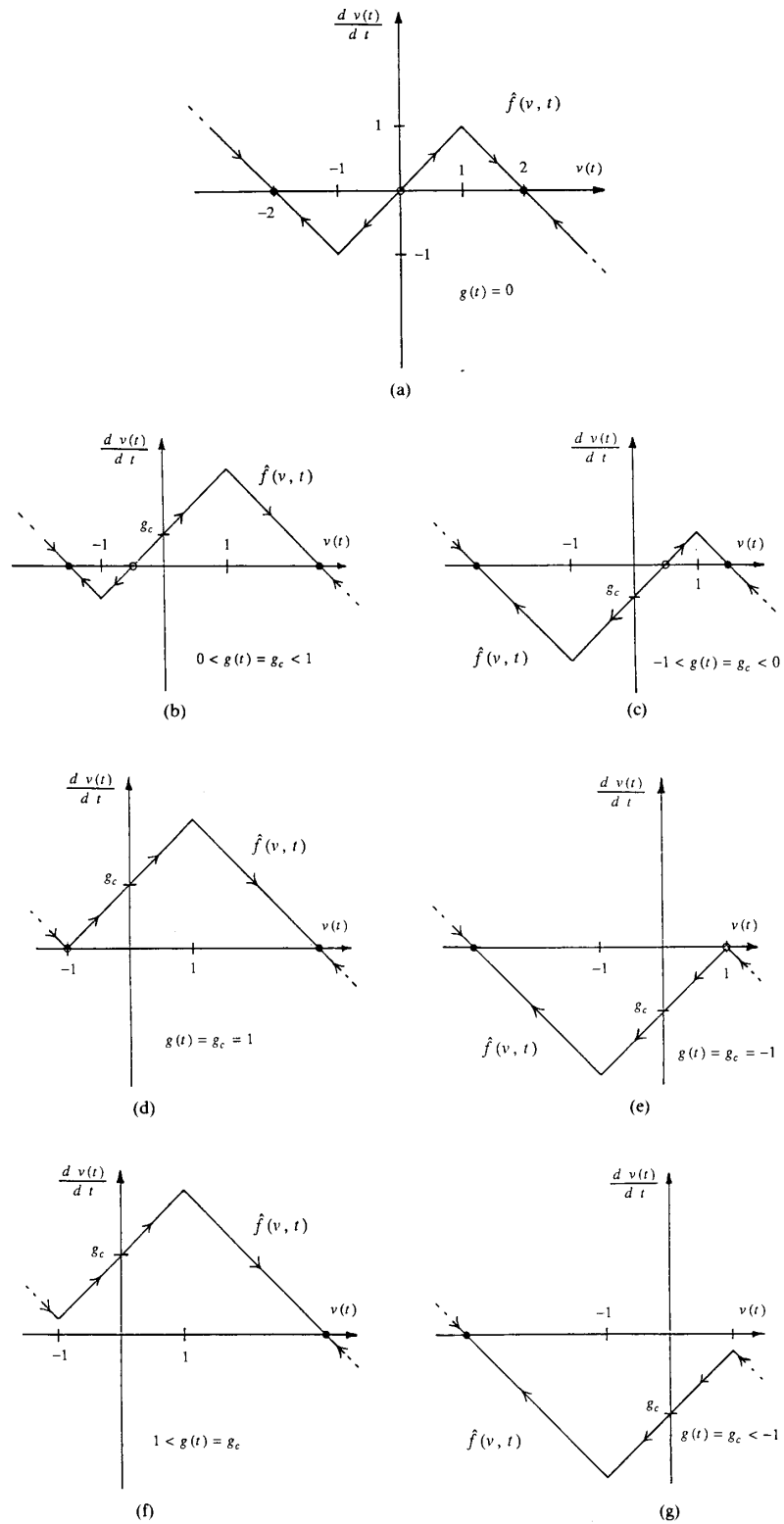
$$A(i, j; i, j) > \frac{1}{R_x}, \qquad (25)$$

Fig. 7. *Dynamic routes* and *equilibrium* points of the equivalent circuit for different values of $g(t)$.

$$\begin{bmatrix} 0.0 & 1.0 & 0.0 \\ 1.0 & 2.0 & 1.0 \\ 0.0 & 1.0 & 0.0 \end{bmatrix} \qquad \begin{bmatrix} 0.0 & -0.25 & 0.0 \\ -0.25 & 2.0 & -0.25 \\ 0.0 & -0.25 & 0.0 \end{bmatrix}$$

(a)   (b)

Fig. 8.   Cloning templates of an interactive cell operator. The unit used here is $10^{-3}\ \Omega^{-1}$.

then each cell of our cellular neural network must settle at a *stable equilibrium point* after the *transient* has decayed to zero. Moreover, the magnitude of all *stable equilibrium points* is greater than 1. In other words, we have the following properties:

$$\lim_{t \to \infty} \left| v_{xij}(t) \right| > 1, \qquad 1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N \quad (26a)$$

and

$$\lim_{t \to \infty} v_{yij}(t) = \pm 1, \qquad 1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N. \quad (26b)$$

Remarks:

(a) The above theorem is significant for cellular neural networks because it implies that the circuit will not oscillate or become chaotic [15], [16].

(b) *Theorem 5* guarantees that our cellular neural networks have binary-value outputs. This property is crucial for solving *classification* problems in image processing applications.

(c) It can be easily shown by the same technique that without the constraint of (25) both case (1) and case (2) can co-exist but case (3) cannot. This implies that remark (a) is true even without the condition (25).

(d) Since $A(i, j; i, j)$ corresponds to a *feedback* from the output of cell $C(i, j)$ into its input, condition (25) stipulates a *minimum* amount of *positive feedback* in order to guarantee that the steady-state output of each cell is either $+1$ or -1. Note that this condition is always violated in a Hopfield neural network since its diagonal coupling coefficients are all assumed to be zero [2]. To guarantee a similar ± 1 binary output in the Hopfield model, it is necessary to choose an *infinite* slope [26] in the linear region of the nonlinear function $f(\cdot)$ in Fig. 4. In contrast, the corresponding slope in a cellular neural network is always chosen to be equal to *one*.

V.   COMPUTER SIMULATIONS OF A SIMPLE CELLULAR NEURAL NETWORK

In this section, we will present a very simple example to illustrate how the cellular neural network described in Section II works. This example will also help to provide a better understanding of the theorems proved in the preceding sections.

The cellular neural network for this example is the same as that shown in Fig. 1, that is, the network size is 4×4. The circuit element parameters of the cell $C(i, j)$ are chosen as follows.

For any $C(k, l) \in N_r(i, j)$ with $r = 1$, i.e., for any 3×3 neighborhood system (see Fig. 2), let

$$A(i, j; i-1, j-1) = 0;$$
$$A(i, j; i-1, j) = 10^{-3}\ \Omega^{-1};$$
$$A(i, j; i-1, j+1) = 0$$
$$A(i, j; i, j-1) = 10^{-3}\ \Omega^{-1};$$
$$A(i, j; i, j) = 2.0 \times 10^{-3}\ \Omega^{-1};$$
$$A(i, j; i, j+1) = 10^{-3}\ \Omega^{-1}$$
$$A(i, j; i+1, j-1) = 0;$$
$$A(i, j; i+1, j) = 10^{-3}\ \Omega^{-1};$$
$$A(i, j; i+1, j+1) = 0$$
$$C = 10^{-9}\ \text{F};\quad R_x = 10^3\ \Omega;\quad I = 0;$$
$$B(i, j; k, l) = 0, \qquad \text{for } C(k, l) \in N_1(i, j).$$

Since $B(i, j; k, l) = 0$, the 3×3 coefficients $A(i, j; k, l)$ alone determine the transient behaviors of the cellular neural network. We will often specify these coefficients in the form of a square array as shown in Fig. 8(a), henceforth called the *cloning template* which specifies the *dynamic rule* of the cellular neural network.

The dynamic equations of the cellular neural network corresponding to the above parameters are given by

$$\frac{dv_{xij}(t)}{dt} = 10^6 \Big[-v_{xij}(t) + 2v_{yij}(t)$$
$$+ v_{yij-1}(t) + v_{yi-1j}(t) + v_{yi+1j}(t) + v_{yij+1}(t) \Big] \quad (27a)$$

and

$$v_{yij}(t) = 0.5 \big(\left| v_{xij}(t) + 1 \right| - \left| v_{xij}(t) - 1 \right| \big),$$
$$\text{for } 1 \leqslant i \leqslant 4; 1 \leqslant j \leqslant 4. \quad (27b)$$

It is convenient to recast the right-hand side of (27a) into the *symbolic form*

$$\frac{dv_{xij}(t)}{dt} = -10^6 v_{xij}(t) + 10^6 \underbrace{\begin{bmatrix} & 1 & \\ 1 & 2 & 1 \\ & 1 & \end{bmatrix}}_{T} * v_{yij}(t) \quad (28)$$

with the help of the *two-dimensional convolution operator* $*$ defined below:

Definition 3:

For any *cloning template* T (such as the example shown in Fig. 8(a)) which defines the dynamic rule of the cell circuit, we define the convolution operator $*$ by

$$T * v_{ij} = \sum_{C(k, l) \in N_r(i, j)} T(k-i, l-j) v_{kl} \quad (29)$$

where $T(m, n)$ denotes the entry in the mth row and nth column of the cloning template, $m = -1, 0, 1$ and $n = -1, 0, 1$, respectively.

Note that in the above definition $A(i, j; k, l)$ is assumed to be independent of i and j for this cellular neural network. This property is said to be *space invariant*, which implies that $A(i, j; k, l)$ can be expressed as $A(k-i, l-$

j). Unless stated otherwise, all cellular neural networks are assumed to have the space invariant property. This property allows us to specify the *dynamic rules* of cellular neural networks by using cloning templates.

To study the transient behavior of (27a), let us apply an initial voltage $v_{xij}(0)$ across the capacitor of each cell $C(i, j)$. Each initial voltage may be assigned any voltage between -1 and 1, as stipulated in (2d).

The circuit simulator we used to obtain our transient response is PWLSPICE [17], which is a modified version of SPICE3 [18] for piecewise-linear circuit analysis. The input and output files are the same as those for SPICE3.

The transient behavior of the above cellular neural network with the initial condition specified in the array of Fig. 9(a) has been simulated. The state variables of the circuit, v_x, at $t = 5$ μS are shown in Fig. 9(b). The maximum absolute value of the state variables at $t = 5$ μS is equal to 6, approximately. The upper bound v_{max} of v_x as computed from equation (3) of *Theorem 1* is equal to 7, which is very close to 6!

The corresponding outputs, v_y, at $t = 5$ μS are shown in Fig. 9(c). Observe that all output variables assume binary values, either 1 or -1, as predicted by *Theorem 5*. (Here the condition $A(i, j; i, j) > 1/R_x$ is satisfied.)

Since it would take too much space to display the transient of the entire circuit, we only display the transient behavior of one cell $C(2,2)$ in Fig. 9(d). The initial value of the state variable is equal to 1.0, and the value at $t = 5$ μS is equal to 2.02, approximately. The maximum value of $v_{x22}(t)$ is equal to 3 and occurs at $t = 0.8$ μS, approximately. Since the state variable is kept above 1.0 during the entire transient regime, the corresponding output remains constant at 1.0, as predicted from Fig. 4.

Before we investigate how many distinct values that $v_{xij}(t)$ can assume in the *steady state*, consider first the following:

Definition 4:

A *stable cell equilibrium state*, v_{xij}^*, of a typical cell of a cellular neural network is defined as the state variable v_{xij} of cell $C(i, j)$, which satisfies

$$\left.\frac{dv_{xij}(t)}{dt}\right|_{v_{xij} = v_{xij}^*} = 0 \text{ and } |v_{xij}^*| > 1 \qquad (30)$$

under the *assumption* $v_{ykl} = \pm 1$ for all neighbor cells $C(k, l) \in N_r(i, j)$.

Remark:

Definition 4 holds for *any assumed* combination of $v_{ykl} = \pm 1$, and, therefore, may *not* represent an actual component of an equilibrium state of the overall circuit.

For our current example, equivalently, the stable cell equilibrium states of an inner[4] cell circuit $C(i, j)$ are the solutions v_{xij} of the dc cell circuit equations obtained by

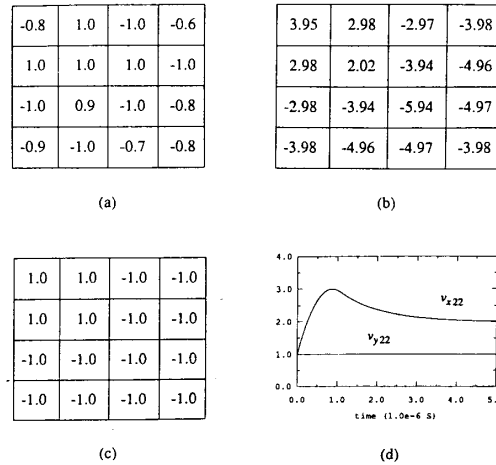[4]The same method is applicable for the boundary cells as well.

-0.8	1.0	-1.0	-0.6
1.0	1.0	1.0	-1.0
-1.0	0.9	-1.0	-0.8
-0.9	-1.0	-0.7	-0.8

(a)

3.95	2.98	-2.97	-3.98
2.98	2.02	-3.94	-4.96
-2.98	-3.94	-5.94	-4.97
-3.98	-4.96	-4.97	-3.98

(b)

1.0	1.0	-1.0	-1.0
1.0	1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0

(c)

(d)

Fig. 9. Computer simulation of a 4×4 cellular neural network. (a) Initial states. (b) Final states. (c) Outputs at steady state. (d) Transient waveforms of cell circuit $C(2,2)$.

replacing all capacitors by open circuits; namely,

$$v_{xij} = 2v_{yij} + v_{yi-1j} + v_{yi+1j} + v_{yij-1} + v_{yij+1}, \qquad (31a)$$

under the conditions:

$$|v_{xkl}| \geqslant 1, \qquad 1 \leqslant k, l \leqslant 4 \qquad (31b)$$

and

$$|v_{ykl}| = 1, \qquad 1 \leqslant k, l \leqslant 4. \qquad (31c)$$

Substituting condition (31c) into the dc equation (31a) and using the $\text{sgn}(\cdot)$ function defined by

$$\text{sgn}(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases}$$

we obtain

$$v_{xij} = 2\,\text{sgn}(v_{yij}) + \text{sgn}(v_{yi-1j})$$
$$+ \text{sgn}(v_{yi+1j}) + \text{sgn}(v_{yij-1}) + \text{sgn}(v_{yij+1}). \qquad (32)$$

Furthermore, since $\text{sgn}(v_{yij}) = \text{sgn}(v_{xij})$ from (2b), it follows that

$$v_{xij} - 2\,\text{sgn}(v_{xij}) = \text{sgn}(v_{yi-1j}) + \text{sgn}(v_{yi+1j})$$
$$+ \text{sgn}(v_{yij-1}) + \text{sgn}(v_{yij+1}). \qquad (33)$$

Observe that the right-hand side of (33) can only assume five possible values; namely, -4, -2, 0, 2, and 4. It follows that the corresponding values that can be assumed by the state variable v_{xij} are -6, -4, $(-2$ or 2), 4, and 6.

It follows from the above analysis that each *inner* cell circuit for our present example can have only six possible *stable* cell equilibrium states; namely, -6, -4, -2, 2, 4, and 6.

The actual stable cell equilibrium state attained by each cell clearly depends on its initial state as well as those of its neighbor cells. Hence a cell may eventually approach any one of its stable equilibrium state even if its initial state remains unchanged. For example, consider the
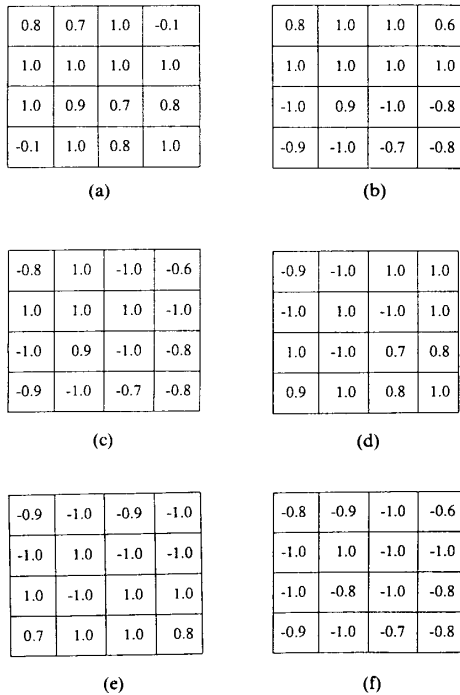
0.8	0.7	1.0	-0.1
1.0	1.0	1.0	1.0
1.0	0.9	0.7	0.8
-0.1	1.0	0.8	1.0

(a)

0.8	1.0	1.0	0.6
1.0	1.0	1.0	1.0
-1.0	0.9	-1.0	-0.8
-0.9	-1.0	-0.7	-0.8

(b)

-0.8	1.0	-1.0	-0.6
1.0	1.0	1.0	-1.0
-1.0	0.9	-1.0	-0.8
-0.9	-1.0	-0.7	-0.8

(c)

-0.9	-1.0	1.0	1.0
-1.0	1.0	-1.0	1.0
1.0	-1.0	0.7	0.8
0.9	1.0	0.8	1.0

(d)

-0.9	-1.0	-0.9	-1.0
-1.0	1.0	-1.0	-1.0
1.0	-1.0	1.0	1.0
0.7	1.0	1.0	0.8

(e)

-0.8	-0.9	-1.0	-0.6
-1.0	1.0	-1.0	-1.0
-1.0	-0.8	-1.0	-0.8
-0.9	-1.0	-0.7	-0.8

(f)

Fig. 10. Six different sets of initial conditions in which the initial states of cell $C(2,2)$ are the same.

3.98	4.97	4.97	3.97
4.97	5.97	5.97	4.97
4.97	5.97	5.96	4.97
3.96	4.97	4.97	3.98

(a)

3.98	4.97	4.97	3.98
2.99	3.99	3.98	2.99
-2.98	-3.94	-3.97	-2.99
-3.98	-4.96	-4.97	-3.98

(b)

3.95	2.98	-2.97	-3.98
2.98	2.02	-3.94	-4.96
-2.98	-3.94	-5.94	-4.97
-3.98	-4.96	-4.97	-3.98

(c)

-3.98	-2.95	2.98	3.98
-2.95	-1.92	3.99	4.97
2.98	3.99	5.95	4.97
3.98	4.97	4.97	3.98

(d)

-3.98	-4.97	-4.97	-3.98
-2.98	-3.96	-3.97	-2.99
2.98	3.96	3.97	2.99
3.98	4.97	4.97	3.98

(e)

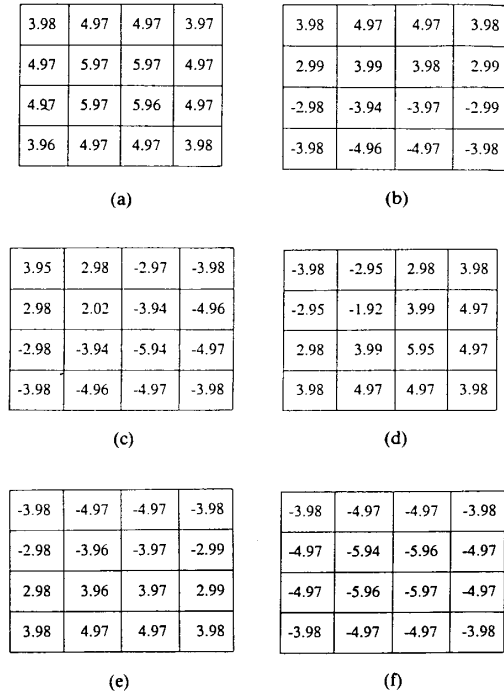| -3.98	-4.97	-4.97	-3.98
-4.97	-5.94	-5.96	-4.97
-4.97	-5.96	-5.97	-4.97
-3.98	-4.97	-4.97	-3.98

(f)

Fig. 11. The final states corresponding to the initial conditions in Fig. 10.

six distinct set of initial conditions in Fig. 10. In all the cases, the initial state of cell $C(2,2)$ are the same, that is, $v_{x22}(0) = 1.0$. After the transient has decayed sufficiently at $t = 5$ μS, the states corresponding to these initial conditions are shown in Fig. 11. Observe that the states of cell $C(2,2)$ at $t = 5$ μS are given, respectively, by 5.97, 3.99, 2.02, -1.92, -3.96, and -5.94. It is clear that all six distinct stable cell equilibrium states of $C(2,2)$ will be attained in the steady state. The transient behaviors of the cell $C(2,2)$ for the six initial conditions in Fig. 10 are shown in Fig. 12. Observe that even though they all start from the same initial point, the steady states approach different points via different routes. Observed also that the transient response is not necessarily monotonic; the response in Fig. 12(c) is a case in point.

Another interesting phenomenon can be observed by choosing the 4 distinct sets of initial conditions in Fig. 13. Observe that even though these initial conditions are *very different*, their corresponding final states at $t = 5$ μS in Fig. 14 are virtually identical. The corresponding outputs in Fig. 15(a) are exactly the same, as expected.

Let us now focus on the *global* dynamic behavior.

Definition 5:

A *stable system equilibrium point* of a cellular neural network is defined to be the *state vector* with all its components consisting of *stable* cell equilibrium states.

It follows from the above definition that a cellular neural network is always at one of its stable system equi-
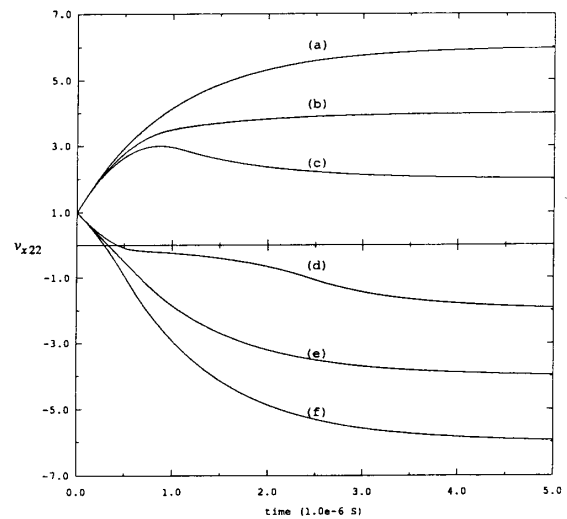

Fig. 12. The transient behaviors of cell $C(2,2)$ corresponding to the initial conditions in Fig. 10.

librium point after the transient has decayed to zero. From the dynamic system theory point of view, the transient of a cellular neural network is simply the *trajectory* starting from some initial state and ending at an equilibrium point of the system. Since any stable system equilibrium point, as defined in Definition 5, of a cellular neural network is a *limit point* of a set of trajectories of the corresponding

0.8	1.0	1.0	0.6
1.0	1.0	1.0	1.0
-1.0	0.9	-1.0	-0.8
-0.9	-1.0	-0.7	-0.8

(a)

0.7	1.0	0.8	0.2
0.9	-0.5	1.0	1.0
-1.0	0.9	-1.0	-0.8
-0.8	-1.0	-0.7	0.2

(b)

-0.4	0.8	1.0	0.6
0.9	0.7	0.7	1.0
-0.8	-0.9	-0.7	-0.8
-0.9	0.2	-0.8	-0.4

(c)

0.5	1.0	0.8	-0.2
0.9	0.5	0.4	1.0
-0.9	-0.9	-0.4	-0.8
-0.9	0.0	-0.7	0.2

(d)

Fig. 13.   Four different sets of initial conditions.

3.98	4.97	4.97	3.98
2.99	3.99	3.98	2.99
-2.98	-3.94	-3.97	-2.99
-3.98	-4.96	-4.97	-3.98

(a)

3.98	4.97	4.97	3.97
2.98	3.97	3.98	2.99
-2.98	-3.97	-3.97	-2.98
-3.98	-4.97	-4.97	-3.97

(b)

3.96	4.97	4.97	3.98
2.98	3.98	3.98	2.99
-2.99	-3.98	-3.98	-2.98
-3.98	-4.96	-4.97	-3.97

(c)

3.98	4.97	4.97	3.97
2.99	3.98	3.97	2.98
-2.99	-3.98	-3.97	-2.98
-3.98	-4.96	-4.97	-3.97

(d)

Fig. 14.   The final states corresponding to the initial conditions in Fig. 13.

1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0
-1.0	-1.0	-1.0	-1.0
-1.0	-1.0	-1.0	-1.0

(a)

4.0	5.0	5.0	4.0
3.0	4.0	4.0	3.0
-3.0	-4.0	-4.0	-3.0
-4.0	-5.0	-5.0	-4.0

(b)

Fig. 15.   (a) The outputs corresponding to the initial conditions in Fig. 13. (b) The stable system equilibrium point corresponding to the initial conditions in Fig. 13.

0.8	0.8	0.8	-0.8
0.8	0.8	0.8	-0.8
0.8	0.8	0.8	-0.8
-0.8	-0.8	-0.8	-0.8

Fig. 16.   Another initial condition.

3.98	4.97	2.99	-1.99
4.97	5.96	3.98	-2.98
2.99	3.98	1.99	-2.99
-1.99	-2.98	-2.99	-3.98

(a)

1.50	1.72	1.74	-1.99
1.72	-2.81	1.97	-1.74
1.74	1.97	1.99	-1.74
-1.99	-1.74	-1.74	-1.49

(b)

Fig. 17.   The outputs of a cellular neural network with its dynamic rule as prescribed by the cloning template in Fig. 8 (a) and (b), respectively, and with its initial condition given by Fig. 16.

1.0	1.0	1.0	-1.0
1.0	1.0	1.0	-1.0
1.0	1.0	1.0	-1.0
-1.0	-1.0	-1.0	-1.0

(a)

1.0	1.0	1.0	-1.0
1.0	-1.0	1.0	-1.0
1.0	1.0	1.0	-1.0
-1.0	-1.0	-1.0	-1.0

(b)

Fig. 18.   The final states corresponding to the outputs in Fig. 17.

differential equations (2), such an attracting limit point has a *basin of attraction*; namely, the union of all trajectories converging to this point. Therefore, the state space of a cellular neural network can be partitioned into a set of basins centered at the stable system equilibrium points.

It follows that all four initial states in Fig. 13 are located within the basin of the same stable system equilibrium point, which is shown in Fig. 15(b).

Our final goal in this section is to take a glimpse at the effects on the choice of a dynamic rule for the cellular neural network. Let us consider the initial condition shown in Fig. 16. First, let us use the same dynamic rule described by the cloning template in Fig. 8(a). The final state at $t = 5$ μS and its corresponding output of the cellular neural network starting from this initial state are shown in Figs. 17(a) and 18(a), respectively. Next, let us change the dynamic rule by using the new cloning template shown in Fig. 8(b). The final state at $t = 5$ μS and its corresponding output starting from the same initial state of Fig. 16 are shown in Figs. 17(b) and 18(b), respectively. Although the only difference between the two outputs shown in Fig. 18 occurs in cell $C(2,2)$, which have opposite values, we will see in [1] that these two dynamic rules perform very different functions when applied to image processing.

Generally speaking, a cellular neural network processes signals by mapping them from one signal space into another one. In our example, the cellular neural network can be used to map an initial state of a system into one of many distinct stable system equilibrium points. If we consider the initial state space as $[-1.0, 1.0]^{M \times N}$ and the output space as $\{-1, 1\}^{M \times N}$, then the dynamical map F, can be defined as

$$F: [-1.0, 1.0]^{M \times N} \rightarrow \{-1, 1\}^{M \times N}. \qquad (34)$$

This means that the map F can be used to partition a continuous signal space into various basins of attractions of the stable system equilibrium points via a *dynamic*

process. This property can be exploited in the design of associative memories, error correcting codes, and fault-tolerant systems.

In general, the limit set of a complex nonlinear system is very difficult, if not impossible, to determine, either analytically or numerically. Although, for piecewise-linear circuit, it is possible to find all dc solutions by using either a brute force algorithm [19] or some more efficient ones [20], [21], it is nevertheless very time consuming for large systems. For our cellular neural networks, in view of the nearest neighbor interactive property, we can solve for all system equilibrium points by first determining the stable cell equilibrium states, and then using the neighbor interactive rules to find the corresponding system equilibrium points.

As presented above, the dynamic behavior of a cellular neural network with zero control operators $(B(i, j; k, l) = 0)$ and nonzero feedback operators $(A(i, j; k, l) \neq 0)$ is reminiscent of a two-dimensional cellular automaton [22]–[25]. Both of them have the parallel signal processing capability and are based on the nearest neighbor interactive dynamic rules. The main difference between a cellular neural network and a cellular automata machine is in their dynamic behaviors. The former is a continuous time while the latter is a discrete-time dynamical system. Because the two systems have many similarities, we can use cellular automata theory to study the steady state behavior of cellular neural networks. Another remarkable distinction between them is that while the cellular neural networks will always settle to stable equilibrium points in the steady state, a cellular automata machine is usually imbued with a much richer dynamical behavior, such as periodic, chaotic and even more complex phenomena. Of course, we have tamed our cellular neural networks by choosing a sigmoid nonlinearity. If we choose some other nonlinearity for the nonlinear elements, many more complex phenomena will also occur in cellular neural networks. These two models will be compared in more detail in Section VII.

VI. MULTILAYER CELLULAR NEURAL NETWORKS

We can generalize the single-layer cellular neural network introduced in Section II to a multilayer cellular neural network. Instead of only one state variable in the single-layer case there may be several state variables in each cell of a multilayer cellular neural network. The concept of multilayering emphasizes the interactions of the state variables on the same layer. To avoid clutter, it is convenient to use the convolution operator *, as defined in Definition 3, in the following. Using the convolution operator, we can rewrite (2a) as

$$C\frac{dv_{xij}(t)}{dt} = \frac{-1}{R_x}v_{xij}(t) + A * v_{yij}(t) + B * v_{uij} + I,$$

$$1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N. \quad (35)$$

Then, for multilayer cellular neural networks, the cell dynamic equations can be expressed in the following com-

pact vector form:

$$C\frac{dv_{xij}(t)}{dt} = -R^{-1}v_{xij}(t) + A * v_{yij}(t) + B * v_{uij} + I,$$

$$1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N \quad (36)$$

where

$$C = \begin{pmatrix} C_1 & & 0 & 0 \\ & \ddots & & 0 \\ 0 & & \ddots & \\ 0 & & 0 & C_m \end{pmatrix} \quad R = \begin{pmatrix} R_{1x} & & 0 & 0 \\ & \ddots & & 0 \\ 0 & & \ddots & \\ 0 & & 0 & R_{mx} \end{pmatrix}$$

$$(37a)$$

$$A = \begin{pmatrix} A_{11} & 0 & 0 & 0 & 0 \\ \cdot & \cdot & 0 & & 0 \\ \cdot & & \cdot & 0 & 0 \\ \cdot & & & \cdot & 0 \\ A_{m1} & \cdot & \cdot & \cdot & A_{mm} \end{pmatrix}$$

$$B = \begin{pmatrix} B_{11} & 0 & 0 & 0 & 0 \\ \cdot & \cdot & 0 & & 0 \\ \cdot & & \cdot & 0 & 0 \\ \cdot & & & \cdot & 0 \\ B_{m1} & \cdot & \cdot & \cdot & B_{mm} \end{pmatrix} \quad (37b)$$

$$v_{xij} = \begin{pmatrix} v_{1xij} \\ \vdots \\ v_{mxij} \end{pmatrix} \quad v_{yij} = \begin{pmatrix} v_{1yij} \\ \vdots \\ v_{myij} \end{pmatrix} \quad v_{uij} = \begin{pmatrix} v_{1uij} \\ \vdots \\ v_{muij} \end{pmatrix} \quad (37c)$$

$$I = \begin{pmatrix} I_1 \\ \vdots \\ I_m \end{pmatrix} \quad (37d)$$

and where m denotes the number of the variables in the multilayer cell circuit. Here, the convolution operator * between a matrix and a vector is to be decoded like matrix multiplication but with the operator * inserted between each entry of the matrix and of the vector.

Observe that C and R are diagonal matrices, whereas A and B are block triangular matrices.

Remarks:

(a) For multilayer cellular neural networks, all of the results presented in the previous sections still hold with some minor modifications. The stability can be proved from the bottom layer (layer 1) to the upper ones by noting the block triangular structures of the A and B matrices.

(b) Since there are several state variables in a cell circuit, we can choose multiple dynamic rules concurrently for the different state variables. This property makes the network extremely flexible and allows us to deal with more complicated image processing problems.

(c) In addition to using multiple dynamic rules as mentioned in (b), we can choose different time constants for the different state variables of the cell circuits. As a limiting case, we can choose $C_q = 0$ for some state variable v_{qxij}, thereby obtaining a set of differential and algebraic equations. This property gives

TABLE I
COMPARISON OF THREE MATHEMATICAL MODELS HAVING
SPATIAL REGULARITIES

Model	Cellular neural network	Partial differential equation	2-D cellular automata
time	continuous	continuous	discrete
space	discrete	continuous	discrete
state value	real	real	binary number
dynamics	nonlinear	linear (for (38))	nonlinear

us even more flexibility in the design of cellular neural networks for practical problems.

VII. RELATION TO PARTIAL DIFFERENTIAL EQUATIONS AND CELLULAR AUTOMATA

In general, cellular neural networks can be characterized by a *large* system of ordinary differential equations. Since all of the cells are arranged in a regular array, we can exploit many spatial properties, such as regularity, sparsity and symmetry in studying the dynamics of cellular neural networks.

There are two mathematical models which can characterize dynamical systems having these spatial properties. One is *partial differential equation*, and the other is *cellular automata*. Partial differential equation, cellular automata, and our cellular neural networks share a common property; namely, their dynamic behavior depend only on their spatial local interactions. Our objective in this section is to identify the relationships between our cellular neural networks and these two mathematical models.

Consider a partial differential equation first. The well-known heat equation from Physics is

$$\frac{\partial^2 u(x, y, t)}{\partial x^2} + \frac{\partial^2 u(x, y, t)}{\partial y^2} = \frac{1}{\kappa} \frac{du(x, y, t)}{dt} \quad (38)$$

where κ is a constant, called the thermal conductivity. The solution, $u(x, y, t)$ of the heat equation is a continuous function of the time, t, and the space variables, x and y. If the function $u(x, y, t)$ is approximated by a set of functions $u_{ij}(t)$, which is defined as

$$u_{ij}(t) = u(ih_x, jh_y, t) \quad (39)$$

where h_x and h_y are the space interval in the x and y coordinates, then, the partial derivatives of $u(x, y, t)$ with respect to x and y can be replaced approximately by

$$\frac{\partial^2 u(x, y, t)}{\partial x^2} + \frac{\partial^2 u(x, y, t)}{\partial y^2}$$
$$\approx \frac{1}{4}\left[u_{ij-1}(t) + u_{ij+1}(t) + u_{i-1j}(t) + u_{i+1j}(t)\right]$$
$$- u_{ij}(t), \quad \text{for all } i, j. \quad (40)$$

Thus the heat equation can be approximately by a set of equations

$$\frac{1}{\kappa} \frac{du_{ij}(t)}{dt} = \frac{1}{4}\left[u_{ij-1}(t) + u_{ij+1}(t) + u_{i-1j}(t) + u_{i+1j}(t)\right]$$
$$- u_{ij}(t), \quad \text{for all } i, j. \quad (41)$$

Comparing (41) with (2), we see a remarkable similarity between the two equations. They are both ordinary differential equations with the nearest neighbor variables involved in the dynamic rules. The important difference between these two equations is that our cell equation (2) is a nonlinear (piecewise linear) ordinary differential equation whereas (41) is a linear ordinary differential equation.

Consider next the relationship between our cellular neural network model and the cellular automaton model. The two-dimensional cellular automaton is defined by [22]

$$a_{ij}(n+1) = \phi\left[a_{kl}(n) \text{ for all } C(k, l) \in N_r(i, j)\right]. \quad (42)$$

If we discretize the time, t, in the cell equation (2) and let $B(i, j; k, l) = 0$ for all i, j, k and l, we would obtain

$$\frac{C}{h}\left[v_{xij}((n+1)h) - v_{xij}(nh)\right]$$
$$= -\frac{1}{R_x}v_{xij}(nh)$$
$$+ \sum_{C(k, l) \in N_r(i, j)} A(i, j; k, l)v_{ykl}(nh) + I,$$
$$1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N \quad (43a)$$

and

$$v_{yij}(nh) = 0.5R_y\left(\left|v_{xij}(nh) + 1\right| - \left|v_{xij}(nh) - 1\right|\right), \quad 1 \leqslant i \leqslant M; 1 \leqslant j \leqslant N. \quad (43b)$$

After rearranging (43a) and substituting the resulting expression for $v_{xij}((n+1)/h)$ into (43b), we obtain

$$v_{yij}(n+1) = \phi'\left[v_{xij}(n), v_{ykl}(n) \text{ for all } C(k, l) \in N_r(i, j)\right] \quad (44)$$

where

$$v_{yij}(n) \triangleq v_{yij}(nh), \quad v_{xij}(n) \triangleq v_{xij}(nh) \quad (45a)$$

$$\phi'\left[v_{xij}(n), v_{ykl} \text{ for all } C(k, l) \in N_r(i, j)\right]$$
$$= g\left[\left(\frac{h}{CR_x} + 1\right)v_{xij}(n)\right.$$
$$\left. + \frac{h}{C}\sum_{(k, l) \in N_r(i, j)} A(i, j; k, l)v_{ykl}(n) + \frac{hI}{C}\right] \quad (45b)$$

and

$$g[z] = \frac{1}{2}(|z + 1| - |z - 1|). \quad (45c)$$

Comparing (42) and (44), we can once again see a remarkable similarity between them. The main difference is that for cellular automata, the state variables are binary value variables and the dynamic function is a logic function of the previous states of the neighbor cells, whereas
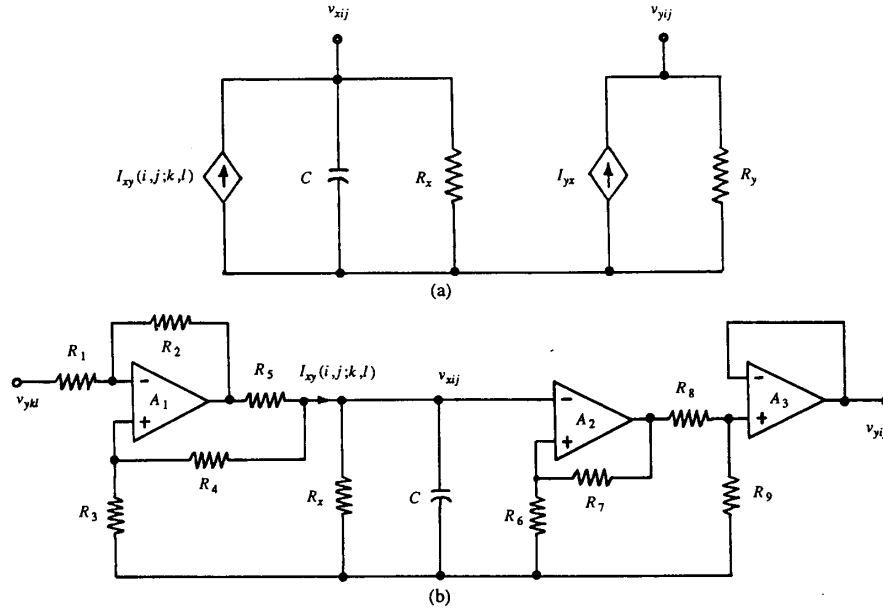
Fig. 19.   An op amp implementation of a simplified cell circuit of cellular neural networks.

for cellular neural networks, the state variables are real-valued variables and the dynamic function is a nonlinear real function of the previous states of the neighbor cells.

Our comparisons of the above three mathematical models are summarized in Table I.

VIII. Concluding Remarks

In this paper, we have proposed a *new* circuit architecture, called a cellular neural network, which can perform *parallel* signal processing in *real* time. We have proved some theorems concerning the dynamic range and the steady states of cellular neural networks. We have also used computer simulation to illustrate some dynamic properties of simple cellular neural networks. In view of the *nearest neighbor* interactive property of cellular neural networks, they are much more amenable to VLSI implementation than general neural networks. In spite of the "local nature" of the nearest neighbor interconnections, cellular networks are nevertheless imbued with some global properties because of the propagation effects of the local interactions during the transient regime. In fact, in a companion paper [1], this *transient* regime will be exploited to show how our ability to analyze the "local" dynamics (via the *dynamic route approach* in [13]) will allow us to *steer* the system trajectories into a configuration of stable equilibria corresponding to some global pattern we seek to recognize. Indeed, our ability to *control the local circuit dynamics* will stand out as one of the most desirable features of cellular neural networks. Furthermore, cellular neural networks have practical dynamic ranges, whereas, general neural networks often suffer from a severe dynamic range restrictions in the circuit implementation stage. Since this paper represents the first study

on cellular neural networks, there are clearly many theoretical and practical problems yet to be solved in our future research on this subject. Nevertheless, some rather impressive and promising applications of cellular neural networks to pattern recognition have already been achieved and will be reported in a companion paper [1].

Appendix

The circuit in Fig. 19(a) is a simplified cell circuit of cellular neural networks. It consists of the basic circuit elements, namely, a linear capacitor C; a linear resistor R_x; a linear voltage controlled current source $I_{xy}(i, j; k, l) = A(i, j; k, l)v_{ykl}$; a subcircuit with the piecewise-linear function $v_{yij} = 0.5 \ (|v_{xij} + 1| - |v_{xij} - 1|)$.

One possible op amp implementations of the above circuit is shown in Fig. 19(b). The voltage controlled current source $I_{xy}(i, j; k, l)$ is realized by op amp A_1 and resistors $R_1 - R_5$. It can be shown that

$$I_{xy}(i, j; k, l) = \frac{-R_2}{R_1 R_5} v_{ykl} \tag{46}$$

under the condition that

$$\frac{R_2}{R_1} = \frac{R_4 + R_5}{R_3}. \tag{47}$$

The output resistance of $I_{xy}(i, j; k, l)$ is infinite under this condition (47). The piecewise-linear function $v_{yij}(v_{xij})$ is realized by op amps A_2, A_3 and resistors $R_6 - R_9$ with the constraint that

$$\frac{R_6 + R_7}{R_6} = \frac{R_8 + R_9}{R_9} + |V_{cc}| \tag{48}$$

where V_{cc} is the voltage of the power supply.

The two circuits in Fig. 19 are equivalent in the sense that they have the same state and output equations.

REFERENCES

[1] L. O. Chua and L. Yang, "Cellular neural networks: Applications," *IEEE Trans. Circuits Syst.*, pp. 1273–1290, this issue.

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

[3] J. J. Hopfield and D. W. Tank "'Neural' computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141–152, 1985.

[4] J. J. Hopfield and D. W. Tank, "Computing with neural circuits: a model," *Science (USA)*, vol. 233, no. 4764, pp. 625–633, 1986.

[5] D. W. Tank and J. J. Hopfield, "Simple 'neuron' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Circuits Syst.*, vol. CAS-33, pp. 533–541, 1986.

[6] M. P. Kennedy and L. O. Chua, "Unifying the Tank and Hopfield linear programming circuit and the canonical nonlinear programming circuit of Chua and Lin," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 210–214, 1987.

[7] T. Kohonen, *Self-Organization and Associative Memory*. New York: Springer-Verlag, 1984.

[8] C. Koch J. Marroquin, and A. Yuille, "Analog 'neuronal' networks in early vision," *Proc. Natl. Acad. Sci. USA*, vol. 83, p. 4263–4267, 1986.

[9] I. Kanter and H. Sompolinsky, "Associative recall of memory without errors," *Phys. Rev. A*, vol. 35, no. 1, pp. 380–392, 1987.

[10] J. Buhmann and K. Schulten, "Associative recognition and storage in a model network of physiological neurons," *Biol. Cybern.*, vol. 54, pp. 319–335, 1986.

[11] M. P. Kennedy and L. O. Chua, "Neural networks for nonlinear programming," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 554–562, May 1988.

[12] L. O. Chua and T. Lin, "A neural network approach to transform image coding," *Int. J. Circuit Theory Appl.*, to be published.

[13] L. O. Chua, C. A. Desoer, and E. S. Kuh, *Linear and Nonlinear Circuits*. New York: McGraw-Hill, 1987.

[14] Leon O. Chua, *Introduction to Nonlinear Network Theory*. New York: McGraw-Hill, 1969.

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

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

[17] L. O. Chua and Y. Liao, "PWLSPICE: SPICE for piecewise-linear circuits," in preparation.

[18] T. Quarles, A. R. Newton, D. O. Pederson, and A. Sangiovanni-Vincentelli, "SPICE.3A7 User's Guide," Dep. Elect. Eng. and Comp. Sci., Univ. of California, Berkeley, June 13, 1986.

[19] L. O. Chua and P. M. Lin, *Computer Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[20] L. O. Chua and R. L. P. Ying, "Finding all solutions of piecewise-linear circuits," *J. Circuit Theory Appl.*, vol. 10, pp. 201–229, 1982.

[21] L. O. Chua and A. C. Deng, "Canonical piecewise-linear analysis: Generalized breakpoint hopping algorithm," *Int. J. Circuit Theory Appl.*, vol. 14, pp. 35–52, Jan. 1986.

[22] S. Wolfram (Eds.), *Theory and Applications of Cellular Automata*. New York: World Scientific, 1986.

[23] N. Packard and S. Wolfram, "Two-dimensional cellular automata," *J. Stat. Phys.*, vol. 38, nos. 5/6, pp. 901–946, 1985.

[24] T. Toffoli and N. Margolus, *Cellular Automata Machines — A New Environment for Modeling*. Cambridge, MA: M.I.T. Press, 1986.

[25] K. Preston, Jr. and M. J. B. Duff, *Modern Cellular Automata: Theory and Applications*. New York, Plenum, 1984.

[26] 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.

[27] M. P. Kennedy and L. O. Chua, "Circuit theoretic solutions for neural networks—An old approach to a new problem," in *Proc. First IEEE Conf. on Neural Networks*, vol. 2, pp. 169 - 176, San Diego, CA, June 21 - 24, 1987.

L. O. Chua (S'60–M'62–SM'70–F'74), for a photograph and biography please see page 880 of the July 1988 issue of this TRANSACTIONS.

✳

Lin Yang (S'87) was born in Shenyang, China, on April 15, 1953. He received the B.Sc. degree in electronic engineering from Fudan University, Shanghai, China, in 1982, and the M.S. degree in information engineering from Tsinghua University, Beijing, China, in 1985.

Currently, he is with the Department of Electrical Engineering and Computer Sciences of the University of California, Berkeley, where he is working towards the Ph.D. degree.

His research interests include nonlinear dynamical systems, image processing, neural networks, and computer-aided design of VLSI circuits.