Cellular neural network (CNN)

(Cellular Neural Network Approach. Three simple CNN classes.

Discrete-time CNN.)

Cellular Neural Network Approach.

The cellular neural network (CNN) model was introduced by Chua and Yang (1988) and consists of a recurrent nonlinear network in which neurons are locally connected and dynamics is identical for each node. These neurons are commonly called cells. The connection with the cells outside the *r*-neighborhood is enabled by the propagation effects of network dynamics.

Cellular neural networks are very suited for high-speed parallel signal processing like image or other two-dimensional signals processing. In the same time CNN is used for solving partial differential equations (PDEs).

The CNN dynamics is described by a set of differential equations.

Each cell in CNN has an input, an internal state and an output. Any one cell is connected only to its neighboring cells. The cell located in the position $(i,\ j)$ of the two-dimensional $M \times N$ area is denoted as C_{ij}, and its *r*-neighborhood N_{ij}^r is defined by

$$N_{ij}^r = \left\{ C_{kl}\ , \max\left(|k-i|, |l-j|\right) \le r, 1 \le k \le M; 1 \le l \le N \right\},$$

where the size of the neighborhood *r* is a positive integer number.

Set N_{ij}^r is sometimes referred as the $(2r+1)\times(2r+1)$ neighborhood. For the 3×3 neighborhood depicted in fig. 4.1, *r* should be 1. Thus, the parameter *r* controls the connectivity of a cell, i.e. the number of active synapses that connects the cell with its immediate neighbors.

CNN is entirely characterized by the set of nonlinear differential equations associated with cells in the network.

The mathematical model for the state equation of the single cell C_{ij} is given by the following relation:

$$\frac{\partial x_{ij}(t)}{\partial t} = -x_{ij}(t) + \sum_{kl \in N^r} A_{ij,kl} y_{kl}(t) + \sum_{kl \in N^r} B_{ij,kl} u_{kl}(t) + I_{ij}, \tag{4.1}$$

where $x_{ij}(t)$ denotes the state of the cell C_{ij}; $y_{kl}(t)$, $u_{ij}(t)$ denote the output and input respectively of cells C_{kl} located in the sphere of influence with radius *r*; $A_{ij,kl}$ and $B_{ij,kl}$ are the feedback and feedforward templates respectively; I_{ij} is the bias term.

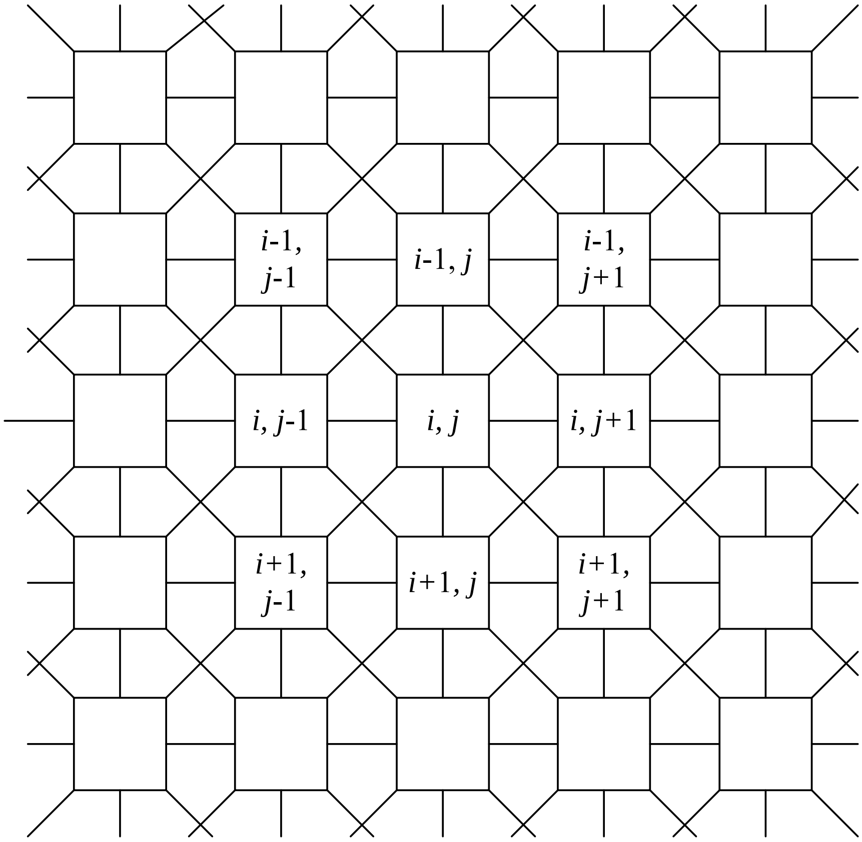

Fig. 4.1. The 3×3 neighborhood

In many applications CNN is isotropic, that is space-invariant. Isotropic network is characterized by parameters in equation (4.1) which are fixed for the whole neural network. In the case of isotropic CNN, for example under $r = 1$, the terms in state equation (4.1) are represented below.

Contributions from the feedback synaptic weights $A_{ij,kl}$ in (4.1). In view of space-invariance, we can write

$$\sum_{kl\in N_{ij}^r} A_{ij,kl}y_{kl}(t) = \sum_{|k-i|\le 1}\ \sum_{|l-j|\le 1} A(i-k,\, j-l)\, y_{kl}(t) =$$

$$= a_{-1,-1}y_{i-1,\,j-1} + a_{-1,0}y_{i-1,\,j} + a_{-1,1}y_{i-1,\,j+1} +$$

$$+a_{0,-1}y_{i,\,j-1} + a_{0,0}y_{i,\,j} + a_{0,1}y_{i,\,j+1} +$$

$$+a_{1,-1}y_{i+1,\,j-1} + a_{1,0}y_{i+1,\,j} + a_{1,1}y_{i+1,\,j+1} =$$

$$= \sum_{k=-1}^{1}\ \sum_{l=-1}^{1} a_{k,l}\, y_{i+k,\,j+l} \triangleq$$

$$\triangleq \begin{bmatrix} a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ a_{0,-1} & a_{0,0} & a_{0,1} \\ a_{1,-1} & a_{1,0} & a_{1,1} \end{bmatrix} \otimes \begin{bmatrix} y_{i-1,\,j-1} & y_{i-1,\,j} & y_{i-1,\,j+1} \\ y_{i,\,j-1} & y_{i,\,j} & y_{i,\,j+1} \\ y_{i+1,\,j-1} & y_{i+1,\,j} & y_{i+1,\,j+1} \end{bmatrix} = \mathbf{A} \otimes \mathbf{Y}_{ij}\,, \qquad (4.2)$$

where the matrix $\mathbf{A}$ of the 3×3 dimension is called the feedback cloning template, the symbol $\otimes$ denotes the summation of dot products, henceforth called a template dot product. In discrete mathematics, this operation is called “spatial convolution.” The 3×3 matrix $\mathbf{Y}_{ij}$ in (4.2) can be obtained by moving an opaque mask with the 3×3 window to the position $(i,\ j)$ of the $M\times N$ output image $\mathbf{Y}$, henceforth called the output image at $C(i,\ j)$.

An element a_{kl} is called the center (respectively, surround) element, weight or coefficient, of the feedback template $\mathbf{A}$, if and only if $(k, l) = (0, 0)$ (respectively, $(k, l) \ne (0, 0)$).

It is sometimes convenient to decompose the $\mathbf{A}$ template as follows

$$\mathbf{A} = \mathbf{A}^0 + \overline{\mathbf{A}}\,, \qquad (4.3)$$

where $\mathbf{A}^0$ and $\overline{\mathbf{A}}$ are called the center and surround component templates, respectively,

$$\mathbf{A}^0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & a_{0,0} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \ \bar{\mathbf{A}} = \begin{bmatrix} a_{-1,-1} & a_{-1,0} & a_{-1,1} \\ a_{0,-1} & 0 & a_{0,1} \\ a_{1,-1} & a_{1,0} & a_{1,1} \end{bmatrix}.$$

Contributions from the input synaptic weights $B_{ij,kl}$ in (4.1). Following the above notes, we can write

$$\sum_{kl \in N_{ij}^r} B_{ij,kl} u_{kl}(t) = \sum_{|k-i| \le 1} \sum_{|l-j| \le 1} B(i-k, j-l) u_{kl}(t) =$$

$$= \sum_{k=-1}^{1} \sum_{l=-1}^{1} b_{k,l}\, u_{i+k,j+l} \triangleq$$

$$\triangleq \begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{bmatrix} \otimes \begin{bmatrix} u_{i-1,j-1} & u_{i-1,j} & u_{i-1,j+1} \\ u_{i,j-1} & u_{i,j} & u_{i,j+1} \\ u_{i+1,j-1} & u_{i+1,j} & u_{i+1,j+1} \end{bmatrix} = \mathbf{B} \otimes \mathbf{U}_{ij}, \quad (4.4)$$

where the 3×3 matrix $\mathbf{B}$ is called the feedforward or input cloning template, and $\mathbf{U}_{ij}$ is the translated masked input image.

Contribution from the threshold term in (4.1). In view of space-invariance, denote $I_{ij} = z$.

The output signal of the cell C_{ij} is given by the following equation

$$y_{ij}(t) = f(x_{ij}(t)), \quad (4.5)$$

where $y_{ij}(t)$ denotes the output value of the cell C_{ij}, $f(\bullet)$ is the non-linear function.

Generally, $f(\bullet)$ may be any continuous function. Non-linear function is specified as:

– the unity gain piecewise linear saturation function described by equation

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

Graphically this function is shown in fig. 4.2;

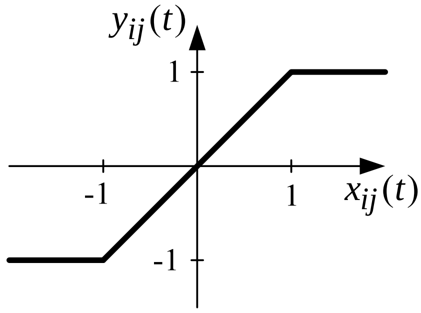

Fig. 4.2. Piecewise linear saturation function

– the signum function (a so-called hard limiter)

$$y_{ij}(t) = f\left(x_{ij}(t)\right) = \mathrm{sgn}\left(x_{ij}(t)\right) = \begin{cases} +1 & \text{для } x_{ij}(t) \geq 0, \\ -1 & \text{для } x_{ij}(t) < 0, \end{cases}$$

– the step function;

– the sigmoid function obeying conditions:

$$\left| f\left(x_{ij}(t)\right) \right| \leq \mathrm{Const},$$

$$\frac{df\left(x_{ij}(t)\right)}{dx_{ij}(t)} \geq 0 .$$

In the literature, the most frequently used CNNs are using the two types of $f(\bullet)$ just introduced (being either the hard limiter or the unity gain piecewise linear saturation function).

A significant CNN feature is that CNN has two independent input capabilities: the generic input and the initial state of cells. They are normally bounded by

$$\left|u_{ij}(t)\right| \leq 1 \quad \text{and} \quad \left|x_{ij}(0)\right| \leq 1 .$$

Similarly, if $\left| f(\cdot) \right| \leq 1$ then $\left|y_{ij}(t)\right| \leq 1$.

In fig. 4.3 it is shown how two-dimensional signals are processed with the standard cellular neural network having templates of 3x3 dimensions. Applying the image *U* on the CNN input and having at state an initial image *X*, the CNN output image *Y* is obtained by using operators *A*, *B*, *z*, when that equilibrium point is reached.

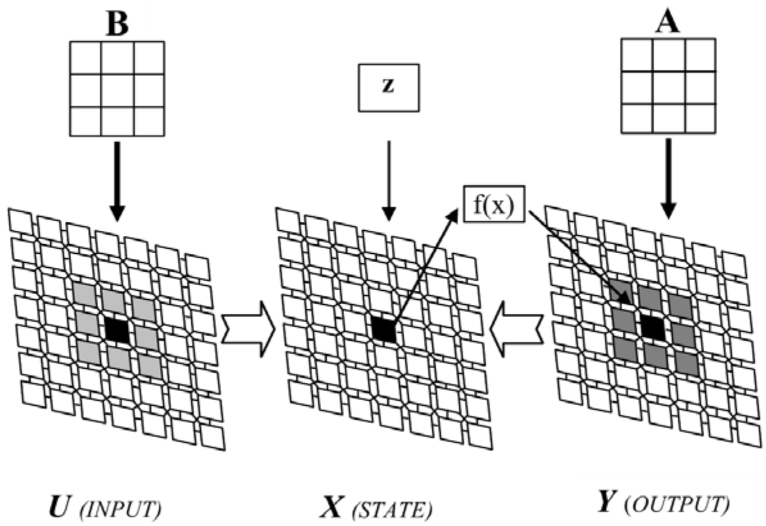

Fig. 4.3. Signals processing with a standard cellular neural network having templates of 3x3 dimensions

Using the above notations in (4.2)–(4.4), **the space-invariant CNN is completely described with state equation**

$$\dot{x}_{ij} = -x_{ij} + \mathbf{A} \otimes \mathbf{Y}_{ij} + \mathbf{B} \otimes \mathbf{U}_{ij} + z, \tag{4.6}$$

where $\otimes$ is the sign of vector product.

We will usually decompose equation (4.6) as follows

$$\dot{x}_{ij} = -x_{ij} + a_{00} f(x_{ij}) + \overline{\mathbf{A}} \otimes \mathbf{Y}_{ij} + \mathbf{B} \otimes \mathbf{U}_{ij} + z.$$

Let's denote

$$g(x_{ij}) = -x_{ij} + a_{00} f(x_{ij}),$$

$$w_{ij}(x_{ij}, t) = \overline{\mathbf{A}} \otimes \mathbf{Y}_{ij} + \mathbf{B} \otimes \mathbf{U}_{ij} + z,$$

$$h_{ij}(x_{ij}, w_{ij}) = g_{ij}(x_{ij}) + w_{ij}(x_{ij}, t),$$

where $h_{ij}(x_{ij}, w_{ij})$ is called the rate function, $g(x_{ij})$ is called the driving-point (DP) component because it is closely related to a central concept from non-linear circuit theory, $w_{ij}(x_{ij}, t)$ is called the offset level.

Four CNN classes.

Each CNN is uniquely defined by three terms of the cloning templates $\{\mathbf{A}, \mathbf{B}, z\}$, which consist of 19 real numbers for a 3×3 neighborhood ($r=1$). Since real numbers are uncountable, there are infinitely many distinct CNN templates, of which the following three subclasses are the simplest and hence mathematically tractable.

Definition: Excitatory and Inhibitory synaptic weights (fig. 4.4). A feedback synaptic weight a_{kl} is said to be excitatory (respectively, inhibitory) if and only if it is positive (respectively, negative).

A synaptic weight is "excitatory" (respectively, "inhibitory") because it makes the rate function $h_{ij}(x_{ij}, w_{ij})$ more positive (less positive) for a positive input, and hence increases (respectively, decreases) $\dot{x}_{ij}$, namely the rate of growth of $x_{ij}(t)$.

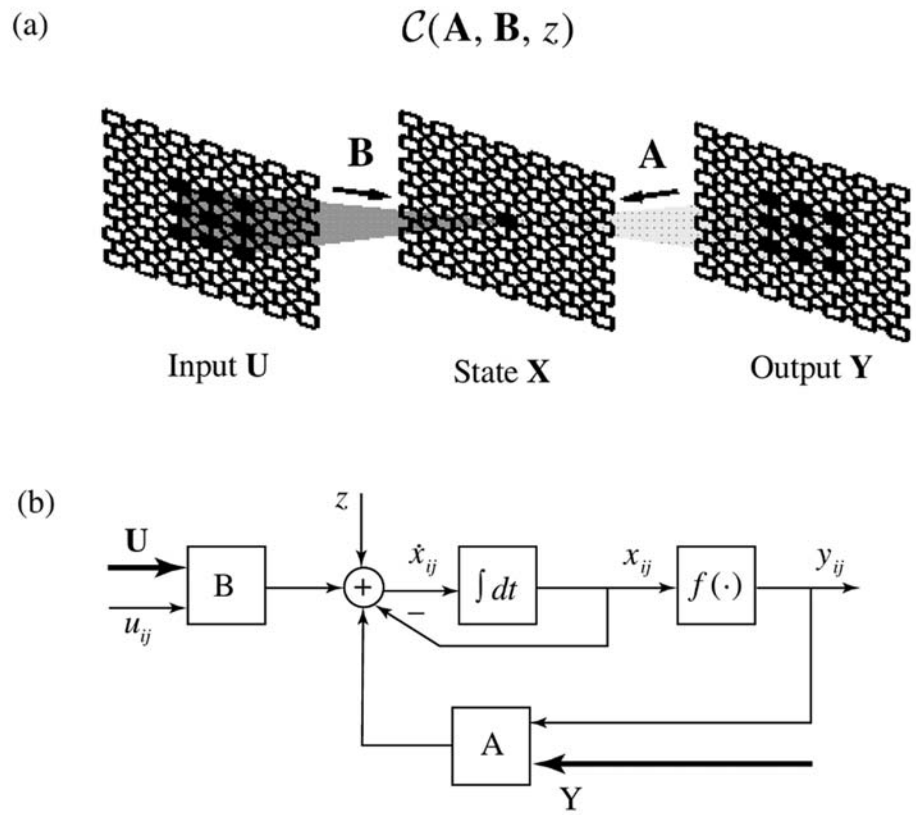

Fig. 4.4. A space-invariant CNN $C=(\mathbf{A}, \mathbf{B}, z)$ with a 3×3 neighborhood N_{ij}^{r}.
(a) Signal flow structure of the CNN with a 3×3 neighborhood.
Two shaded cones symbolize the weighted contributions of input

and output voltages of cell $C_{kl} \in N_{ij}^1$ to the state voltage of the center cell C_{ij}.
(b) System structure of the cell C_{ij}. Arrows printed in bold mark parallel data paths from the input and the output of the surround cells u_{kl} and y_{kl}, respectively. Arrows with thinner lines denote the threshold, input, state and output, z, u_{ij}, x_{ij} and y_{ij}, respectively.

Definition: Zero-feedback (feedforward) class $C = (0, \mathbf{B}, z)$ (fig. 4.5). A CNN belongs to the zero-feedback class $C = (0, \mathbf{B}, z)$ if and only if all feedback template elements are zero, i.e., $\mathbf{A} \equiv 0$.

Each cell of a zero-feedback CNN is described by

$$\dot{x}_{ij} = -x_{ij} + \mathbf{B} \otimes \mathbf{U}_{ij} + z.$$

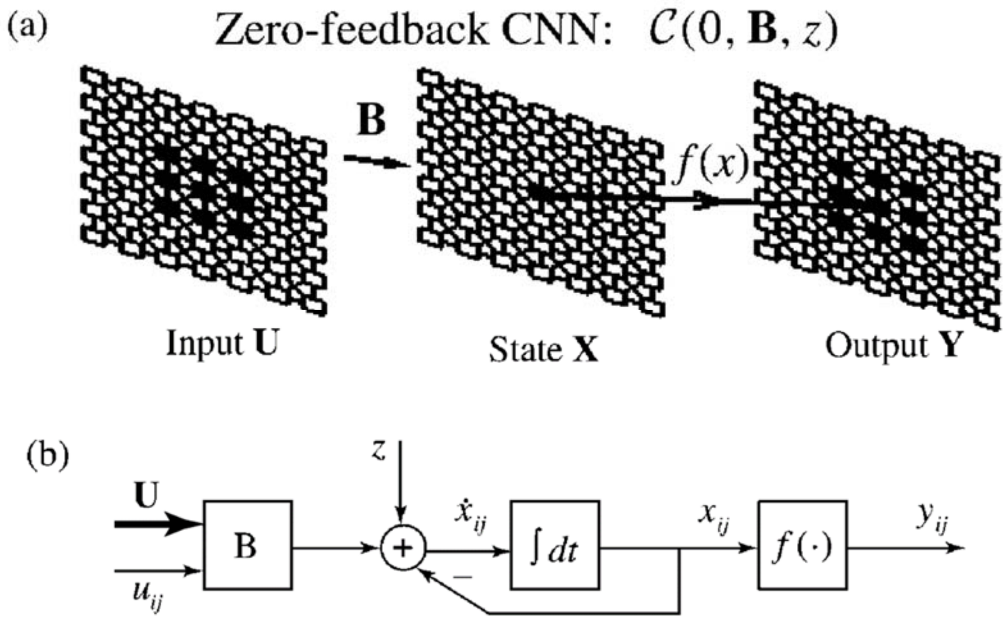

Fig. 4.5. Zero-feedback (feedforward) CNN $C = (0, \mathbf{B}, z)$.
(a) Signal flow structure of a zero-feedback CNN with a 3×3 neighborhood. The cone symbolizes the weighted contributions of input voltages of cells $C_{kl} \in N_{ij}^1$ to the center cell C_{ij}.
(b) System structure of the cell C_{ij}. Arrows printed in bold denotes the input signal from the surround cells. In this case, there is no self-feedback, and no couplings from the outputs of the surround cells.

Definition: Zero-input (Autonomous) class $C = (\mathbf{A}, 0, z)$ **(fig. 4.6).** A CNN belongs to the zero-input class $C = (\mathbf{A}, 0, z)$ if and only if all feedforward template elements are zero, i.e., $\mathbf{B} \equiv 0$.

Each cell of a zero-input CNN is described by

$$\dot{x}_{ij} = -x_{ij} + \mathbf{A} \otimes \mathbf{Y}_{ij} + z.$$

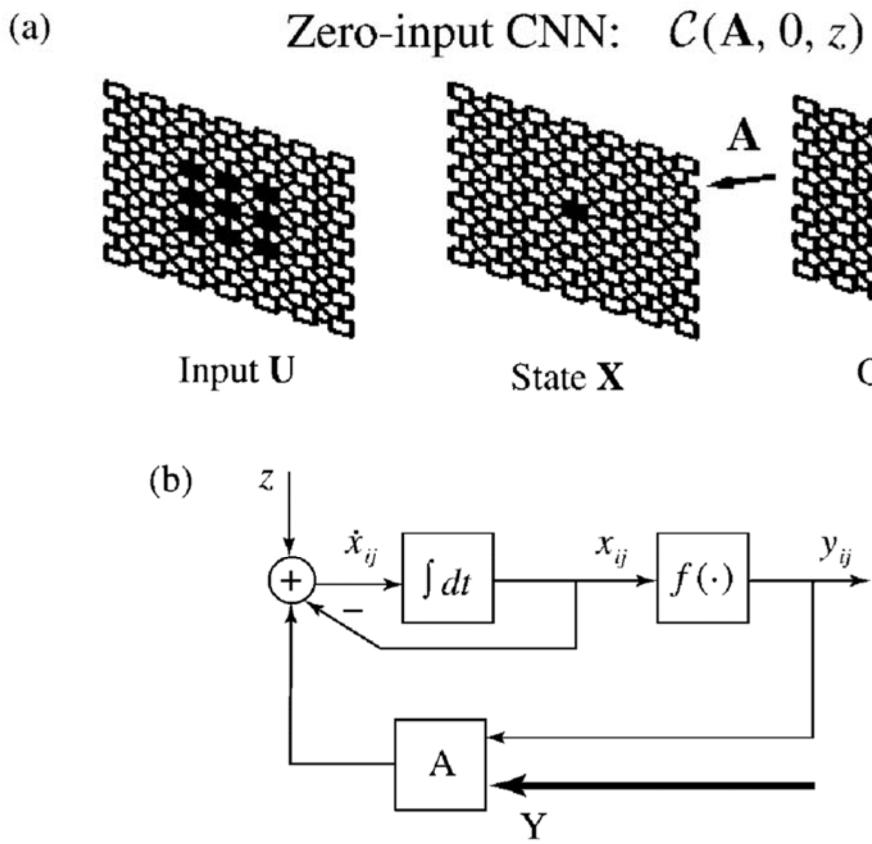

Fig. 4.6. Zero-input (Autonomous) CNN $C = (\mathbf{A}, 0, z)$.
(a) Signal flow structure of a zero-input CNN with a 3×3 neighborhood.
The cone symbolizes the weighted contributions of the output
voltage of cells $C_{kl} \in N_{ij}^{1}$ to the center cell C_{ij}.
(b) System structure of the center cell C_{ij}.
Arrow printed in bold denotes the signal fed-back from the outputs
of the surround cells. In this case, there are no input signals.

Definition: Uncoupled (scalar) class $C = (\mathbf{A}^{0}, \mathbf{B}, z)$ **(fig. 4.7).** CNN belongs to the uncoupled class $C = (\mathbf{A}^{0}, \mathbf{B}, z)$ if and only if $a_{ij} = 0$ except $i = j$, i.e., $\overline{\mathbf{A}} \equiv 0$.

Each cell of an uncoupled CNN is described by a scalar nonlinear ODE which is not coupled to its neighbors:

$$\dot{x}_{ij} = -x_{ij} + a_{00} f(x_{ij}) + \mathbf{B} \otimes \mathbf{U}_{ij} + z .$$

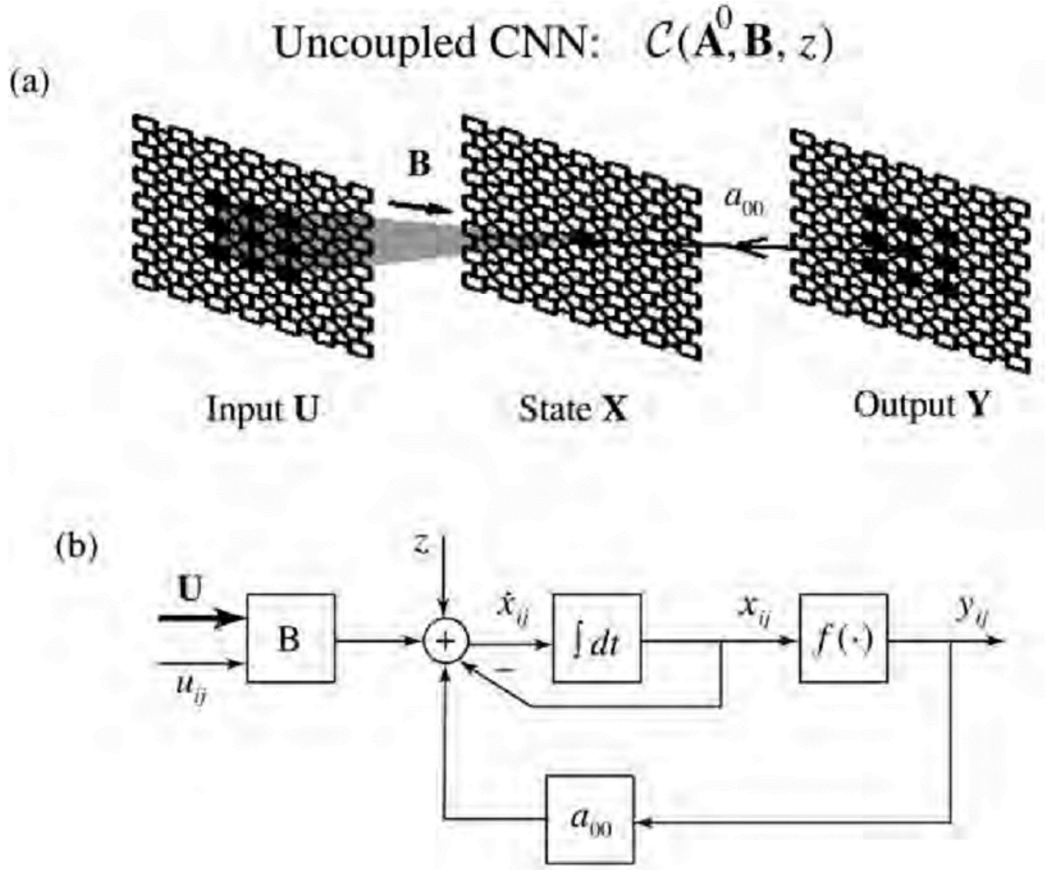

Fig. 4.7. Uncoupled CNN $C = (\mathbf{A}^0, \mathbf{B}, z)$.
(a) Signal flow structure of an uncoupled CNN with a 3×3 neighborhood.
The cone symbolizes the weighted contributions of the input voltages
of cells $C_{kl} \in N^1_{ij}$ to the center cell C_{ij}.
(b) System structure of the center cell C_{ij}.
Arrow printed in bold denotes the input signals from the surround cells.
In this case, the data streams simplified into simple streams marked
by thinner arrows, indicating only a “scalar” self-feedback,
but no couplings from the outputs of the surround cells.

Next we show the possible electronic circuit model of cell. In fig. 4.8, voltage-controlled current sources are used to implement various coupling terms. These trans-conductances can be easily constructed on CMOS integrated circuits.

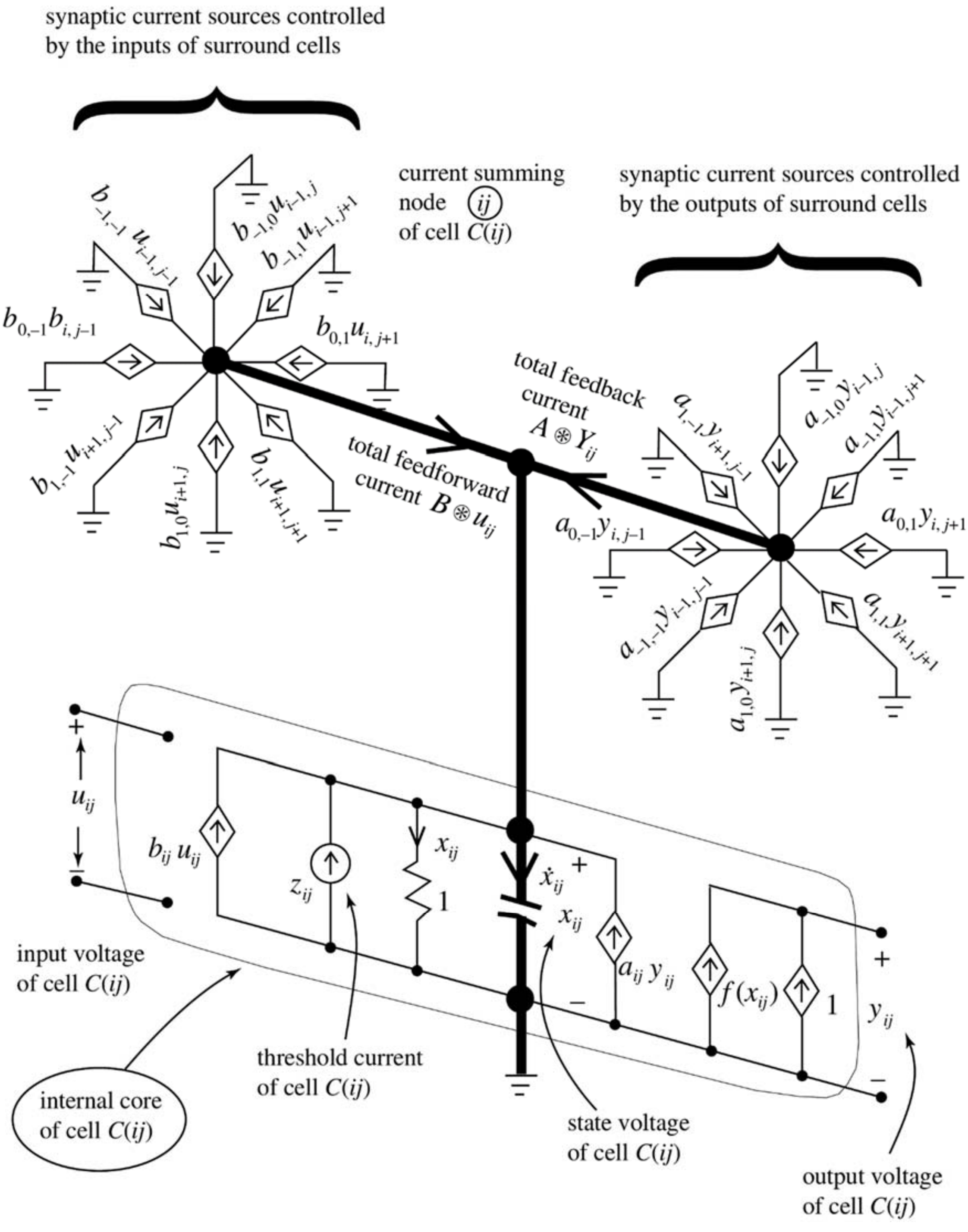

Fig. 4.8. Cell realization of a standard CNN cell C_{ij}.

All diamond-shape symbols denote a voltage-controlled current source which injects a current proportional to the indicated controlling voltage u_{kl} or y_{kl}, weighted by b_{kl} or a_{kl}, respectively, except for the rightmost diamond $f(x_{ij})$ in the internal core which is a nonlinear voltage controlled current source, resulting in the output voltage $y_{ij} = f(x_{ij})$.

Discrete-time CNN

The CNN description in discrete time domain is formed from expression (4.1) as a result of the following transformations:

– the approximation of derivative

$$\frac{dx_{ij}(t)}{dt} \approx \frac{x_{ij}(t) - x_{ij}(t-\Delta t)}{\Delta t} = x_{ij}(n) - x_{ij}(n-1),$$

where n is the normalized discrete time. Let us suppose, that $\Delta t = 1$;

– the transition from differential equation (4.1) to recursive difference equation

$$x_{ij}(n) - x_{ij}(n-1) = -x_{ij}(n-1) + \sum_{kl \in N_{ij}^{r}} A_{ij,kl} y_{kl}(n-1) +$$

$$+ \sum_{kl \in N_{ij}^{r}} B_{ij,kl} u_{kl}(n-1) + I_{ij}. \qquad (4.7)$$

Eventually, on the bases of (4.5) and (4.7) the model of cell C_{ij} in discrete-time CNN (DTCNN) is described as

$$x_{ij}(n) = \sum_{kl \in N_{ij}^{r}} A_{ij,kl} y_{kl}(n-1) + \sum_{kl \in N_{ij}^{r}} B_{ij,kl} u_{kl}(n-1) + I_{ij},$$

$$(4.8)$$

$$y_{ij}(n) = f\left(x_{ij}(n)\right).$$

The structure of DTCNN cell C_{ij} is depicted in fig. 4.9, here T is the delay element. This structure corresponds to model (4.8).

Expressions (4.8) are transformed into the isotropic model of DTCNN cell (the parameters of isotropic model are fixed for the whole neural network, $z = I_{ij}$):

$$x_{ij}(n) = \sum_{kl \in N_{ij}^{r}} A_{ij,kl} y_{kl}(n-1) + \sum_{kl \in N_{ij}^{r}} B_{ij,kl} u_{kl}(n-1) + z,$$

$$y_{ij}(n) = f\left(x_{ij}(n)\right).$$

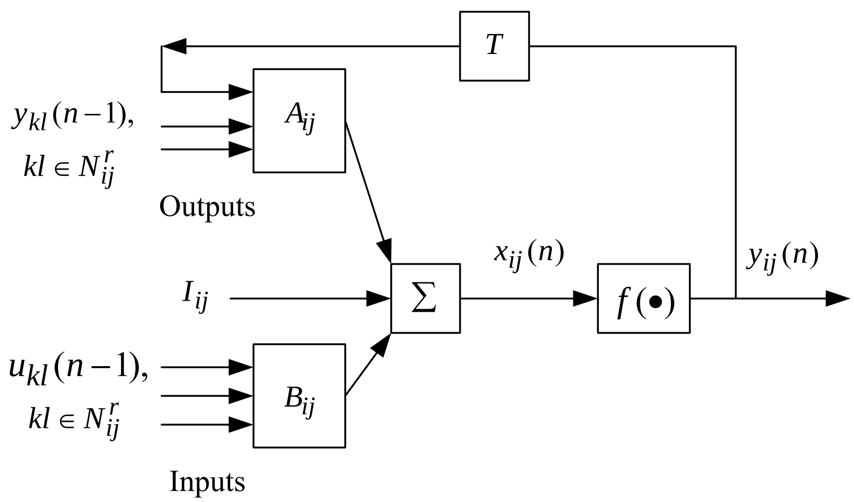

Fig. 4.9. The structure of DTCNN cell C_{ij}