AN ANALYTIC METHOD FOR DESIGNING SIMPLE CELLULAR NEURAL NETWORKS

by

Leon O. Chua and Patrick Thiran

Memorandum No. UCB/ERL M91/3

8 January 1991

AN ANALYTIC METHOD FOR DESIGNING SIMPLE CELLULAR NEURAL NETWORKS

by

Leon O. Chua and Patrick Thiran

Memorandum No. UCB/ERL M91/3

8 January 1991

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

AN ANALYTIC METHOD FOR DESIGNING SIMPLE CELLULAR NEURAL NETWORKS

by

Leon O. Chua and Patrick Thiran

Memorandum No. UCB/ERL M91/3

8 January 1991

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

An Analytic Method for Designing Simple Cellular Neural Networks

Leon O. Chua and Patrick Thiran *

January 8, 1991

Abstract

A method is proposed for synthetizing Cellular Neural Networks designed for simple applications. Based on the comparison principle for ordinary differential equations, our method leads to a set of inequalities which must be satisfied by the parameters of the cloning template defining the Cellular Neural Network, in order to guarantee a correct operation of the network.

*Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720, USA

1 Introduction

A Cellular Neural Network (abreviated as CNN) is a large scale nonlinear circuit, made of only locally connected cells [1], [2]. Numerous applications of these CNNs are quite encouraging. Nevertheless, the design of a cloning template is a non-trivial problem, whose solution is often based only on intuition and some computer simulations. Recently, some algorithms have been proposed to design CNNs (e.g. [3], [4]). In this paper, we introduce an analytic method to synthetize a CNN for solving a given problem.

Our method relies on a simple application of the so-called comparison principle, which will provide bounds on the state and output waveforms of an analog processing cell circuit. We will then be able to find conditions on the elements of the CNN, ensuring a correct functionning of the CNN for a particular application. This comparison principle has been used for other applications than Neural Networks, such as time-analysis of MOS VLSI circuits.

First, we will review briefly the architecture of CNNs in Section 2. In Section 3, we will compute the bounds of the state and output of a cell, and we will illustrate how to use this technique to design CNNs for shadowing (Section 4), motion detection (Section 5) and hole filling (Section 6).

2 Architecture of Cellular Neural Networks

The CNN architecture is described in [1], here we will recall only the main results. The cell located in the (i, j) position of a two-dimensional $M \times N$ array is denoted by C_{ij}, and its r-neighborhood $\mathcal{N}_{ij}$ is defined by

$$\mathcal{N}_{ij} = \{C_{kl} \mid \max\{|k-i|, |l-j|\} \leq r;\ 1 \leq k \leq M, 1 \leq l \leq N\}$$

where r is a positive integer number. The input u_{ij} of a cell C_{ij} is assumed to be a constant with magnitude less than or equal to 1. The state of the cell C_{ij} at time t is denoted by $x_{ij}(t)$, while the output at time t is denoted by $y_{ij}(t)$. The initial state $x_{ij}(0)$ is assumed to have a magnitude less than or equal to 1. The equations of a cell $C_{ij}(1 \leq i \leq M, 1 \leq j \leq N)$ are the state equation

$$\dot{x}_{ij}(t) = -x_{ij}(t) + \sum_{C_{kl} \in \mathcal{N}_{ij}} A_{i,j,k,l}\, y_{kl}(t) + \sum_{C_{kl} \in \mathcal{N}_{ij}} B_{i,j,k,l}\, u_{kl} + I \qquad (1)$$

and the output equation

$$y_{ij}(t) = f(x_{ij}(t)), \tag{2}$$

where the piecewise-linear function $f(\cdot)$, represented in figure 1, is given by

$$f(v) = \frac{1}{2}(|v+1| - |v-1|). \tag{3}$$

Note that $|y_{ij}(t)| \leq 1$ for all $t \geq 0$. The piecewise-linear function $f(\cdot)$ can be approximated to within any precision by a smooth (C^1) strictly increasing function. The templates A and B are assumed to be space invariant, which implies that $A_{i,j,k,l}$ can be expressed as $A_{k-i,l-j}$, and similarly that $B_{i,j,k,l}$ can be expressed as $B_{k-i,l-j}$. Finally, if we define

$$g_{ij}(t) = \sum_{C_{kl} \in \mathcal{N}_{ij} \setminus \{C_{ij}\}} A_{k-i,l-j}\, y_{kl}(t) + \sum_{C_{kl} \in \mathcal{N}_{ij}} B_{k-i,l-j}\, u_{kl} + I, \tag{4}$$

we can then restate the state equation (1) as

$$\dot{x}_{ij}(t) = -x_{ij}(t) + A_{0,0}\, y_{ij}(t) + g_{ij}(t). \tag{5}$$

The notation $\mathcal{N}_{ij} \setminus \{C_{ij}\}$ stands for the set of all cells belonging to the neighborhood $\mathcal{N}_{ij}$ of the cell C_{ij}, except the cell C_{ij} itself.

The condition

$$A_{0,0} > 1 \tag{6}$$

must always be satisfied, so that the magnitude of all stable steady states is greater than or equal to 1. It has been shown ([1],[5]) that the complete stability of the network is assured for some very important classes of templates (such as symmetric, positive/negative and opposite-sign templates). Nevertheless, no assumption (other than space-invariance and (6)) will be made on the parameters of the network in this paper.

3 Upper and lower bounds

3.1 Circuit theoretic motivation

The equivalent circuit of a cell, represented in figure 2, contains three elements : a linear capacitor (which we assumed equal to 1F without any loss of generality), a piecewise-linear voltage controlled resistor $\mathcal{R}$ whose driving point characteristic is

$$i_R = v_R - A_{0,0}\, f(v_R)$$

and a time-varying independent current source $i_s(t) = g_{ij}(t)$. The voltage v_C across the capacitor is equal to x_{ij}.

Now, replace the time-varying current source $i_s(t)$ by a dc independent current source I_s^+ and let v_C^+ denote the resulting voltage across the capacitor. Repeat the same operation with another dc independent current source I_s^-, and let v_C^- be the voltage across the capacitor in this latter case.

If $v_C^-(t_0) = v_C(t_0) = v_C^+(t_0)$ at some time $t_0 \geq 0$, and if $I_s^- \leq i_s(t) \leq I_s^+$ for $t \geq t_0$, we will see that the solution v_C of the first circuit is bounded by the solutions v_C^+ and v_C^- of the second and third circuits: $v_C^-(t) \leq v_C(t) \leq v_C^+(t)$ for all $t \geq t_0$.

This property of a cell circuit illustrates the guideline of the method presented in this paper. We will find an upper and a lower bound of $i_s(t) = g_{ij}(t)$ for $t \geq t_0$. We can then easily compute closed-form upper and lower bounds of the function $v_c(\cdot) = x_{ij}(\cdot)$, and by doing so, estimate the state of the cell.

3.2 Preliminaries

Before beginning our analysis, it is useful to introduce first some notations. Throughout this section, we consider the cell C_{ij} $(1 \leq i \leq N,\, 1 \leq j \leq M)$.

Let $t_0 \geq 0$, and let $g_{ij}^+(t_0)$ (respectively $g_{ij}^-(t_0)$) denote an upper (respectively lower) bound of $g_{ij}(t)$ for $t \geq t_0$. (So $g_{ij}^+(t_0)$ is I_s^+ and $g_{ij}^-(t_0)$ is I_s^- in the previous subsection). Note the dependence of these bounds on t_0.

Consider then the functions $\xi_{ij}^+(t, t_0)$ and $\eta_{ij}^+(t, t_0)$ defined for $t \geq t_0$ by the initial condition

$$\xi_{ij}^+(t_0, t_0) = x_{ij}(t_0), \tag{7}$$

the state equation

$$\dot{\xi}^+_{ij}(t,t_0) = -\xi^+_{ij}(t,t_0) + A_{0,0}\,\eta^+_{ij}(t,t_0) + g^+_{ij}(t_0) \tag{8}$$

and the output equation

$$\eta^+_{ij}(t,t_0) = f(\xi^+_{ij}(t,t_0)). \tag{9}$$

So $\xi^+_{ij}(t,t_0)$ corresponds to the voltage $v^+_C(t)$ defined in the previous subsection.

Note that if $t_0 = 0$, the equations (7), (8) and (9) define a CNN array in the linear threshold class [6]. By a similar reasonning as in [6], we can state that the steady state output of this CNN is

$$\eta^+_{ij}(\infty,t_0) = \lim_{t\to+\infty} \eta^+_{ij}(t,t_0) = sgn[(A_{0,0}-1)\,y_{ij}(t_0) + g^+_{ij}(t_0)]. \tag{10}$$

The solution of the equations (7), (8) and (9) can be explicitly obtained (see [7], chapter 6). Consider the three possible cases

(1) $y_{ij}(t_0) = \eta^+_{ij}(\infty,t_0)$. Then, for all $t \geq t_0$,

$$\xi^+_{ij}(t,t_0) = [x_{ij}(t_0) - A_{0,0}\,y_{ij}(t_0) - g^+_{ij}(t_0)]\,e^{-(t-t_0)} + A_{0,0}\,y_{ij}(t_0) + g^+_{ij}(t_0)$$

and

$$\eta^+_{ij}(t,t_0) = y_{ij}(t_0). \tag{11}$$

(2) $|x_{ij}(t_0)| < 1$. Let

$$t_1 = t_0 + \frac{1}{A_{0,0}-1}\ln\frac{(A_{0,0}-1)\,\eta^+_{ij}(\infty,t_0) + g^+_{ij}(t_0)}{(A_{0,0}-1)\,x_{ij}(t_0) + g^+_{ij}(t_0)}. \tag{12}$$

Then, for $t_0 \leq t < t_1$,

$$\xi^+_{ij}(t,t_0) = \eta^+_{ij}(t,t_0) = [x_{ij}(t_0) + \frac{g^+_{ij}(t_0)}{A_{0,0}-1}]\,e^{(A_{0,0}-1)(t-t_0)} - \frac{g^+_{ij}(t_0)}{A_{0,0}-1}$$

while for all $t \geq t_1$

$$\xi^+_{ij}(t,t_0) = [(1-A_{0,0})\,\eta^+_{ij}(\infty,t_0) - g^+_{ij}(t_0)]\,e^{-(t-t_1)} + A_{0,0}\,\eta^+_{ij}(\infty,t_0) + g^+_{ij}(t_0)$$

and

$$\eta^+_{ij}(t,t_0) = \eta^+_{ij}(\infty,t_0). \tag{13}$$

(3) $y_{ij}(t_0) = -\eta^+_{ij}(\infty, t_0)$. Let

$$t_2 = t_0 + \ln \frac{x_{ij}(t_0) - A_{0,0}\, y_{ij}(t_0) - g^+_{ij}(t_0)}{(1 - A_{0,0})\, y_{ij}(t_0) - g^+_{ij}(t_0)} \tag{14}$$

and

$$t_3 = t_2 + \frac{1}{A_{0,0} - 1} \ln \frac{(1 - A_{0,0})\, y_{ij}(t_0) + g^+_{ij}(t_0)}{(A_{0,0} - 1)\, y_{ij}(t_0) + g^+_{ij}(t_0)}. \tag{15}$$

Then, for $t_0 \le t < t_2$,

$$\xi^+_{ij}(t, t_0) = [x_{ij}(t_0) - A_{0,0}\, y_{ij}(t_0) - g^+_{ij}(t_0)]\, \mathrm{e}^{-(t-t_0)} + A_{0,0}\, y_{ij}(t_0) + g^+_{ij}(t_0),$$

$$\eta^+_{ij}(t, t_0) = y_{ij}(t_0),$$

while for $t_2 \le t < t_3$

$$\xi^+_{ij}(t, t_0) = \eta^+_{ij}(t, t_0) = [y_{ij}(t_0) + \frac{g^+_{ij}(t_0)}{A_{0,0} - 1}]\, \mathrm{e}^{(A_{0,0}-1)(t-t_2)} - \frac{g^+_{ij}(t_0)}{A_{0,0} - 1}$$

and for all $t \ge t_3$

$$\xi^+_{ij}(t, t_0) = [(A_{0,0} - 1)\, y_{ij}(t_0) - g^+_{ij}(t_0)]\, \mathrm{e}^{-(t-t_3)} - A_{0,0}\, y_{ij}(t_0) + g^+_{ij}(t_0)$$

$$\eta^+_{ij}(t, t_0) = -y_{ij}(t_0). \tag{16}$$

We define $\xi^-_{ij}(t, t_0)$ and $\eta^-_{ij}(t, t_0)$ by the equations (7), (8) and (9) where all the superscripts "+" are replaced by "−". All the the formulae stated above apply to $\xi^-_{ij}(t, t_0)$ (which corresponds thus to the voltage v^-_C defined in the previous subsection) and to $\eta^-_{ij}(t, t_0)$ if we replace all the superscripts "+" by "−".

3.3 Theorem 1

The function $\xi^+_{ij}(t, t_0)$ (respectively $\xi^-_{ij}(t, t_0)$), defined in the previous subsection, is an upper (respectively lower) bound of $x_{ij}(t)$ for all $t \ge t_0$:

$$\xi^-_{ij}(t, t_0) \le x_{ij}(t) \le \xi^+_{ij}(t, t_0) \tag{17}$$

for $t \ge t_0$.

Proof: (i) By (5) and with the definition of $g_{ij}^{+}(t_0)$ we have that for $t \geq t_0$

$\dot{x}_{ij}(t) = -x_{ij}(t) + A_{0,0}\, f(x_{ij}(t)) + g_{ij}(t) \leq -x_{ij}(t) + A_{0,0}\, f(x_{ij}(t)) + g_{ij}^{+}(t_0).$

On the other hand, $\xi_{ij}^{+}(t, t_0)$ is the solution of

$$\dot{\xi}_{ij}^{+}(t, t_0) = -\xi_{ij}^{+}(t, t_0) + A_{0,0}\, f(\xi_{ij}^{+}(t, t_0)) + g_{ij}^{+}(t_0)$$

satisfying the initial condition $\xi_{ij}^{+}(t_0, t_0) = x_{ij}(t_0)$.

We may now apply a simple form of the comparison principle (see [8], [9]). Suppose that for some $t_0'' \geq t_0$, $x_{ij}(t_0'') > \xi_{ij}^{+}(t_0'', t_0)$. Let t_0' be the largest t in the interval $[t_0, t_0'']$ such that $x_{ij}(t_0') = \xi_{ij}^{+}(t_0', t_0)$. Therefore, if

$$d_{ij}(t) = x_{ij}(t) - \xi_{ij}^{+}(t, t_0)$$

we have that $d_{ij}(t_0') = 0$ and that for $t' - 0 \leq t \leq t_0''$

$$d_{ij}(t) \geq 0. \tag{18}$$

For the function $f(\cdot)$ shown in figure 1, if $v \leq w$ then $f(w) - f(v) \leq w - v$.

Hence, for $t_0' \leq t \leq t_0''$,

$$\begin{aligned} \dot{d}_{ij}(t) &= \dot{x}_{ij}(t) - \dot{\xi}_{ij}^{+}(t, t_0) \\ &\leq -[x_{ij}(t) - \xi_{ij}^{+}(t, t_0)] + A_{0,0}[f(x_{ij}(t)) - f(\xi_{ij}^{+}(t, t_0))] \\ &\leq -[x_{ij}(t) - \xi_{ij}^{+}(t, t_0)] + A_{0,0}[x_{ij}(t) - \xi_{ij}^{+}(t, t_0)] \\ &\leq (A_{0,0} - 1)\, d_{ij}(t), \end{aligned}$$

which implies that, for $t_0' \leq t \leq t_0''$,

$$d_{ij}(t) \leq d_{ij}(t_0')\, e^{(A_{0,0}-1)(t-t_0')} = 0.$$

It follows from this last inequality and from (18) that $d_{ij}(t) = 0$ for $t_0' \leq t \leq t_0''$. But then $x_{ij}(t_0'') = \xi_{ij}^{+}(t_0'', t_0)$, which contradicts our initial assumption. Therefore, $x_{ij}(t) \leq \xi_{ij}^{+}(t, t_0)$ for any $t \geq t_0$.

(ii) One proves similarly that $\xi_{ij}^{-}(t, t_0) \leq x_{ij}(t)$ for $t \geq t_0$. □

We will use this theorem to predict the steady state of the cell C_{ij}. Indeed, if we can pick some time $\bar{t}_0$ and find an upper bound $\xi_{ij}^{+}(t, \bar{t}_0)$ such that $\eta_{ij}^{+}(\infty, \bar{t}_0) = -1$, then we know by the previous theorem that $y_{ij}(\infty) =$

$\lim_{t\to\infty} y_{ij}(t) = -1$. Similarly, if can find some $\hat{t}_0$ and a lower bound $\xi_{ij}^-(t,\hat{t}_0)$ such that $\eta_{ij}^-(\infty,\hat{t}_0) = 1$, then we can conclude that $y_{ij}(\infty) = 1$. But very often, $\eta_{ij}^+(\infty,t_0) = 1$ and $\eta_{ij}^-(\infty,t_0) = -1$, and we cannot "predict" the value of $y_{ij}(\infty)$, as shown in figure 3. (Nevertheless, we may find some other $\bar{t}_0 \geq t_0$ such that a prediction is possible). It is therefore important to find an upper or lower bound as close as possible to the waveform $x_{ij}(t)$. This in turn leads us to choose the upper and lower bounds of $g_{ij}(t)$ as sharply as we can.

3.4 Corollary 1

The function $\eta_{ij}^+(t,t_0)$ (respectively $\eta_{ij}^-(t,t_0)$), defined in the preliminaries, is an upper (respectively lower) bound of $y_{ij}(t)$ for all $t \geq t_0$:

$$\eta_{ij}^-(t,t_0) \leq y_{ij}(t) \leq \eta_{ij}^+(t,t_0) \tag{19}$$

for $t \geq t_0$.

Proof: This corollary follows immediately from (17) and from the fact that the function $f(\cdot)$ is strictly increasing. □

3.5 Corollary 2

(i) If there is some $t_0 \geq 0$ such that

$$(A_{0,0} - 1)\, y_{ij}(t_0) + g_{ij}^+(t_0) < 0 \tag{20}$$

then $y_{ij}(t) = -1$ for all $t \geq T$ with $T = t_0$ if $y_{ij}(t_0) = -1$, $T = t_1$ (where t_1 is given by (12)) if $|y_{ij}(t_0)| < 1$ and $T = t_3$ (where t_3 is given by (15)) if $y_{ij}(t_0) = 1$.

(ii) If there is some $t_0 \geq 0$ such that

$$(A_{0,0} - 1)\, y_{ij}(t_0) + g_{ij}^-(t_0) > 0 \tag{21}$$

then $y_{ij}(t) = 1$ for all $t \geq T$ with $T = t_0$ if $y_{ij}(t_0) = 1$, $T = t_1$ (where t_1 is given by (12) with the superscripts "+" replaced by "−") if $|y_{ij}(t_0)| < 1$ and $T = t_3$ (where t_3 is given by (15) with the superscripts "+" replaced by "−") if $y_{ij}(t_0) = 1$.

Proof : (i) The condition (20) implies that $\eta_{ij}^{+}(\infty, t_0) = -1$, and by corollary 1, $y_{ij}(t) \leq \eta_{ij}^{+}(t, t_0)$ for $t \geq t_0$. The claim follows then immediately from (11), (12), (13), (14), (15) and (16).

(ii) Similar to (i). □

4 Application 1 : Shadowing template

In order to illustrate how corollary 2 can help us in designing a template, let us first take the very simple example of a "shadowing" CNN. Matsumoto, Chua and Suzuki [10] proposed the cloning template

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2.0 & 2.0 \\ 0 & 0 & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2.0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad I = 0.$$

creating the "shadow" of an object in a bipolar image, as illustrated in figure 4. The image is fed into the input while all the initial states are equal to 1. Is it possible to find CNN templates performing the same operation without having to store the image in the input (i.e. the image will be stored in the initial states only) ? Suppose that we want to create the shadow of an object with the "light source" coming from the right. Note that

(i) Any black pixel in the initial image, or in the image at any temporary stage, will remain black thereafter.

(ii) Any white pixel which has a black neighbor on its right will become black itself. By applying this fact recursively, and because of (i), all the pixels located on the left of a black pixel will eventually become black.

(iii) Finally, a white pixel which has never a black pixel on its right will remain white.

These three observations are the *rules* to be implemented by our shadowing templates. Since this is a one dimensional problem, only the elements of the central row of the template A will be different than zero. Since we do not want to store anything in the input, all the elements of B will be zeros. Finally, since the only external influence on a pixel is due to its right neighbor according to rules (ii) and (iii), the templates which could solve this problem have the form

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & A_{0,0} & A_{0,1} \\ 0 & 0 & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad I$$

Consider a cell C_{ij} ($1 \le i \le M, 1 \le j \le N$). Let us find the conditions on $A_{0,0}$, $A_{0,1}$ and I imposed by the three rules stated above. First, note that

$$g_{ij}(t) = A_{0,1}\, y_{ij+1}(t) + I. \tag{22}$$

(i) The first rule implies that if $y_{ij}(t_0) = 1$ at some time $t_0 \ge 0$, then $y_{ij}(t) = 1$ for $t \ge t_0$. By part (ii) of corollary 2, if we pick $g_{ij}^-(t_0) = -|A_{0,1}| + I$ (which is clearly a lower bound of $g_{ij}(t)$ for any $t \ge 0$), the first rule will always be satisfied if

$$A_{0,0} - 1 - |A_{0,1}| + I > 0. \tag{23}$$

(ii) The second rule will apply when $y_{ij+1}(t_0) = 1$ at some time $t_0 \ge 0$. The first rule implies then that $y_{ij+1}(t) = 1$ for all $t \ge t_0$. Therefore we may take this time $g_{ij}^-(t_0) = A_{0,1} + I$ (In this case, $g_{ij}^-(t_0) = g_{ij}(t)$ for $t \ge t_0$), and corollary 2 (ii) leads to the condition

$$(A_{0,0} - 1)\, y_{ij}(t_0) + g_{ij}^-(t_0) \ge 1 - A_{0,0} + A_{0,1} + I > 0. \tag{24}$$

(iii) The third rule takes care of all other cases, i.e $y_{ij}(0) = -1$ and $y_{ij+1}(t) < 1$ for $t \ge 0$. Note that because of (ii), a pixel will never become black if and only if all the pixels on its right are initially white. Hence $y_{il}(0) = -1$ for $j \le l \le N$. First, consider the edge cell C_{iN}, located on the right edge of the array. For this cell, $g_{iN}(t) = I$. Therefore $y_{iN}(t) = -1$ if

$$1 - A_{0,0} + I < 0. \tag{25}$$

because of corollary 2(i). Next, consider C_{iN-1}. Since $y_{iN}(t) = -1$ for all $t \ge 0$, $g_{iN-1}(t) = -A_{0,1} + I$, and the condition

$$1 - A_{0,0} - A_{0,1} + I < 0. \tag{26}$$

implies that $y_{iN-1}(t) = -1$ for $t \ge 0$. Recursive application of inequality (26) yields that $y_{il}(t) = -1$ for all $t \ge 0$ if $j \le l \le N$. Therefore $y_{ij}(t) = -1$

for all $t \geq 0$ if (25) and (26) are both satisfied. Note that by combining (24) and (26), we deduce that $A_{0,1} > 0$. Hence if (25) is satisfied, so is (26).

To summarize the previous results, if we choose $A_{0,0}$, $A_{0,1}$ and I so that

$$\begin{aligned} A_{0,0} - 1 - A_{0,1} + I &> 0 \\ 1 - A_{0,0} + A_{0,1} + I &> 0 \\ A_{0,0} - 1 - I &> 0 \\ A_{0,0} - 1 &> 0 \end{aligned} \tag{27}$$

we have the *guarantee* that the shadowing templates will work properly. Obviously, the solution of these inequalities is not unique, and many templates will perform the same operation. For instance, we may take

$$A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2.0 & 1.0 \\ 0 & 0 & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad I = 0.5.$$

5 Application 2 : Motion detection

5.1 Hubel and Wiesel's experiment

Hubel and Wiesel have conducted important experiments on the cat's visual cortex. In particular, they found some specific cortical cells which respond only to movement across the retina in a certain direction [11]. Figure 5 shows actual records of selected cells when a bar is moving across a particular region of the cat's visual field. Hubel and Wiesel noted that a slow downward movement of the bar gave a strong response, while upward movement gave only weak response and horizontal movements no responses (and so would oblique movements do as well) ([11], p. 120). They further observed that "discharges of highest frequency were evoked by relatively slow rates of downward movement; rapid movement in either direction gave only very weak responses. If the movement was halted, the cell continued to fire, but less vigorously".

In other words, if the bar is moving in a certain direction at a speed lying within a certain range, some specific cells are firing and the motion of the bar is detected. Our aim is to implement these experiments on CNNs.

5.2 Algorithm

The motion of an object is simulated by a sequence of time discrete samples of moving bipolar images $P(k\Delta t)$ (Here k is an integer and Δt is the sampling time). The consecutive image samples $P(k\Delta t)$ are fed into the inputs while the image samples $P((k+1)\Delta t)$ are applied to the initial states of the CNN. We always suppose that the settling time of the CNN is much smaller than Δt, so that we may assume that the input is constant during the processing time.

We want to detect whether an object is moving with a given speed in a given direction, say for instance a speed lying in a small neighborhood of $\Delta x/\Delta t$ in the horizontal direction (Δx is the resolution of a pixel). Consider two successive images $P(k\Delta t)$ (applied to the input) and $P((k+1)\Delta t)$ (applied to the initial states). The problem amounts then to comparing the position of the moving object in $P(k\Delta t)$ and its position in $P((k+1)\Delta t)$. We make the assumption that the shape of the object does not change between the two snapshots $P(k\Delta t)$ and $P((k+1)\Delta t)$. In a previous approach [12], the first step of the processing was the computation of the difference between $P(k\Delta t)$ and $P((k+1)\Delta t)$. In the approach described here, we do not take the difference between $P(k\Delta t)$ and $P((k+1)\Delta t)$, but we compare directly both pictures. Figure 6 shows four bars at some time $k\Delta t$ (fig 6(a)) and at time $(k+1)\Delta t$ (fig 6(b)), and the desired output of the CNN (fig 6(c)). Consider first the case of the second bar, which moves with the correct speed of $\Delta x/\Delta t$ in the horizontal direction. Then, to every black pixel $u_{ij-1} = 1$ in $P(k\Delta t)$ will correspond a black pixel $x_{ij}(0) = 1$ in $P((k+1)\Delta t)$. In this case the output is the same as the initial state.

Now, consider the first bar (which did not move), the third (whose speed was greater than $\Delta x/\Delta t$) and the fourth (which moved in another direction than horizontal to the right). Then some black pixels $u_{ij-1} = 1$ in $P(k\Delta t)$ will not have a corresponding black pixel $x_{ij}(0) = 1$ in $P((k+1)\Delta t)$. In this case, we want to erase the object from the screen: we must have $y_{ij}(\infty) = -1$ for all $1 \leq i \leq M$, $1 \leq j \leq N$.

As in the previous example of the shadowing templates, we must find a set of rules covering all the cases which may occur. Let us consider a particular cell C_{ij}. We will call "corresponding pixels" the pair $\{u_{ij-1}, x_{ij}(0)\}$. Four cases may occur :

(i) at some time t_0, $y_{ij}(t_0) = -1$. We may then impose that $y_{ij}(t) = -1$ for all $t \geq t_0$ since a white pixel will always remain white.

(ii) $x_{ij}(0) = 1$ and $u_{ij-1} = -1$. Then we know that the object has moved with a "wrong" speed, so we must impose that $y_{ij}(T) = -1$ at some time T.

(iii) $x_{ij}(0) = 1$, $u_{ij-1} = 1$ but $y_{mn}(t_0) = -1$ and $u_{mn-1} = 1$ for at least one cell adjacent to C_{ij}, at some time $t_0 \geq 0$. This means that the object has moved with a "wrong" speed and we must again impose that $y_{ij}(T') = -1$ at some time $T' \geq t_0$. This, in turn, implies that for any cell C_{kl} adjacent to C_{ij}, $y_{kl}(T'') = -1$ for some $T'' \geq T'$, and this process will continue until the entire (connected) object is erased.

(iv) $x_{ij}(0) = 1$, $u_{ij-1} = 1$ and for every cell C_{mn} adjacent to C_{ij}, we have that if $u_{mn-1} = 1$ then $y_{mn}(t) = 1$ for all $t \geq 0$. In this case the object has moved with correct speed, and we will impose that $y_{ij}(t) = 1$ for all $t \geq 0$.

Rules (i) to (iii) apply whenever the object has moved at a different speed and/or in a different direction than 1 pixel per sampling period in the horizontal direction to the right; and rules (i) and (iv) apply when the object has moved with the correct speed in the correct direction.

5.3 Templates for motion detection

Let us now implement these rules with a CNN, as we did in the case of the shadowing templates. First note that only the immediate neighbors of a pixel of $P((k+1)\Delta t)$, and their corresponding pixels in $P(k\Delta t)$, are involved in the computations. Moreover, all these neighboring pixels have the same influence on the central pixel, so we may take all $A_{k-i,l-j}$ equal to some A_n and all $B_{k-i,l-j-1}$ equal to some B_n for $(k,l) \neq (i,j)$. Finally, because of rule (iv), the neighboring cells C_{kl} for which $x_{kl}(0) = u_{kl-1}$ must have no influence on the state of the central cell C_{ij}. Since this influence is equal to $A_{k-i,l-j}\, y_{kl}(t) + B_{k-i,l-j-1}\, u_{kl-1}$, we should take $A_{k-i,l-j} = A_n = -B_n = -B_{k-i,l-j-1}$. Therefore the templates suited for this problem will have the form

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & A_n & A_n & A_n & 0 \\ 0 & A_n & A_{0,0} & A_n & 0 \\ 0 & A_n & A_n & A_n & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix};$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -A_n & -A_n & -A_n & 0 & 0 \\ -A_n & B_{0,-1} & -A_n & 0 & 0 \\ -A_n & -A_n & -A_n & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad I.$$

Note that since A is symmetric, the CNN is completely stable and all states will have a magnitude greater than one at some time t. So the four rules (i) to (iv) encompass all possible cases. With these templates, (4) becomes

$$g_{ij}(t) = A_n \sum_{C_{kl} \in \mathcal{N}_{ij} \backslash \{C_{ij}\}} [y_{kl}(t) - u_{kl-1}] + B_{0,-1}\, u_{ij-1} + I. \tag{28}$$

Let us assume that $A_n > 0$, and let us find a set of parameters $A_{0,0}$, A_n, $B_{0,-1}$ and I such that these templates "implement" rules (i) to (iv).

(i) Rule (i) applies whenever $y_{ij}(t_0) = -1$ at some time $t \geq 0$. Clearly, $16A_n + |B_{0,-1}| + I$ is an upper bound of $g_{ij}(t)$ (given by (28)) for any t. Hence, by corollary 2(i), $y_{ij}(t) = -1$ for $t \geq t_0$ if

$$1 - A_{0,0} + 16A_n + |B_{0,-1}| + I < 0. \tag{29}$$

(ii) Rule (ii) is handled similarly. Now, we have that $y_{ij}(0) = 1$ and $u_{ij-1} = -1$, so $16A_n - B_{0,-1} + I$ is an upper bound of $g_{ij}(t)$ for $t \geq 0$. If

$$A_{0,0} - 1 + 16A_n - B_{0,-1} + I < 0 \tag{30}$$

then, by corollary 2(i), $y_{ij}(t) = -1$ for all $t \geq T$ with

$$T = \frac{1}{A_{0,0} - 1} \ln \frac{1 - A_{0,0} + 16A_n - B_{0,-1} + I}{A_{0,0} - 1 + 16A_n - B_{0,-1} + I}. \tag{31}$$

(iii) Rule (iii) applies when $u_{ij-1} = 1$, and when at least one cell in the neighborhood of C_{ij}, say C_{mn}, is such that $u_{mn-1} = 1$ and $y_{mn}(t_0) = -1$ at some time $t_0 \geq 0$. Because of rule (i), $y_{mn}(t) = -1$ for all $t \geq t_0$. On the other hand, because of rule (ii), for any cell C_{pq} such that $u_{pq-1} = -1$, we have that $y_{pq}(t) = -1$ for $t \geq T$, with T given by (31). This implies that for any cell C_{kl}

$$y_{kl}(t) - u_{kl-1} \leq 0$$

for $t \geq T$, since $u_{kl-1} = \pm 1$ and $|y_{kl}(t)| \leq 1$. Therefore, if $\bar{t} = \max\{t_0, T\}$,

$$\begin{aligned} g_{ij}(t) &= A_n \sum_{C_{kl} \in \mathcal{N}_{ij} \setminus \{C_{ij}, C_{mn}\}} [y_{kl}(t) - u_{kl-1}] + A_n[y_{mn}(t) - u_{mn-1}] + B_{0,-1}\, u_{ij-1} + I \\ &\leq 0 + A_n\,[-1-1] + B_{0,-1} + I = -2A_n + B_{0,-1} + I \end{aligned}$$

for $t \geq \bar{t}$. So $g_{ij}^{+}(\bar{t}) = -2A_n + B_{0,-1} + I$ is an upper bound of $g_{ij}(t)$ for $t \geq \bar{t}$, and if

$$(A_{0,0} - 1)y_{ij}(\bar{t}) + g_{ij}^{+}(\bar{t}) \leq A_{0,0} - 1 - 2A_n + B_{0,-1} + I < 0, \tag{32}$$

corollary 2 (i) implies that $y_{ij}(t) = -1$ for all $t \geq T'$, where T' can be computed by (12) or (15).

(iv) Finally, rule (iv) applies when $x_{ij}(0) = u_{ij-1} = 1$, and when for all cells $C_{pq} \in \mathcal{N}_{ij} \setminus \{C_{ij}\}$ such that $u_{pq-1} = 1$, we have that $y_{pq}(t) = 1$ for all $t \geq 0$. Consequently, for any cell $C_{kl} \in \mathcal{N}_{ij} \setminus \{C_{ij}\}$,

$$y_{kl}(t) - u_{kl-1} \geq 0$$

for all $t \geq 0$, and since $A_n > 0$,

$$g_{ij}^{-}(0) = A_n \sum_{C_{kl} \in \mathcal{N}_{ij} \setminus \{C_{ij}\}} (0) + B_{0,-1}\, u_{ij-1} + I = B_{0,-1} + I$$

is a lower bound of $g_{ij}(t)$ for all $t \geq 0$ in this case. It follows from corollary 2, part (ii) that $y_{ij}(t) = -1$ for all $t \geq 0$ if

$$(A_{0,0} - 1)y_{ij}(0) + g_{ij}^{-}(0) = A_{0,0} - 1 + B_{0,-1} + I > 0. \tag{33}$$

Note that the combination of (30) and (33) yields that $B_{0,-1} > 0$ since $A_n > 0$.

Consequently, if we choose $A_{0,0}$, A_n, $B_{0,-1}$ and I so that

$$\begin{aligned} A_{0,0} - 1 - 16A_n - B_{0,-1} - I &> 0 \\ 1 - A_{0,0} - 16A_n + B_{0,-1} - I &> 0 \\ 1 - A_{0,0} + 2A_n - B_{0,-1} - I &> 0 \\ A_{0,0} - 1 + B_{0,-1} + I &> 0 \\ A_{0,0} - 1 &> 0 \\ A_n &> 0 \end{aligned} \tag{34}$$

the CNN will solve the problem of motion detection stated in subsection 5.1, in all cases. Such a set of parameters is $A_{0,0} = 4.2$, $A_n = 0.3$, $B_{0,-1} = 3.1$ and $I = -6$, or the templates

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.3 & 0.3 & 0.3 & 0 \\ 0 & 0.3 & 4.2 & 0.3 & 0 \\ 0 & 0.3 & 0.3 & 0.3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix};$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -0.3 & -0.3 & -0.3 & 0 & 0 \\ -0.3 & 3.1 & -0.3 & 0 & 0 \\ -0.3 & -0.3 & -0.3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}; \quad I = -6.0.$$

Figure 7 shows a computer simulation of this cloning template in the case of a movement with the correct speed and direction (figure 7(a)), in the correct direction but with a greater speed (figure 7(b)) and in the wrong direction (figure 7(c)). Note finally that if the given speed was different than $\Delta x/\Delta t$ or if the direction was different from the horizontal direction to the right, we can modify the template B accordingly.

6 Application 3 : Hole filler

6.1 Statement of the problem

Our next example is the "Hole-Filler" CNN. We will distinguish here two different problems :

1. A given bipolar image, representing a black object on a white background, is fed into the input while all initial states are equal to 1. Then the CNN must fill up any hole in the object, as shown in figure 8(a). (We will consider here the so-called 8-connected neighborhood. Namely, two pixels diagonally adjacent are considered to be connected. So the object in figure 8(a) is connected.)

2. The given bipolar image which is fed into the input represents now either a black object (on a white background) or a white object (on a black background). All the states of the CNN are initialized at the values of the pixels forming the object (i.e. $x_{ij}(0) = 1$ if the object is black and $x_{ij}(0) = -1$ if the object is white). The CNN must fill up any hole in the object in both cases, as shown in figures 8(a) and (b).

A CNN cloning template solving the first problem is reported in [13]. The second problem is more restrictive than the first one, and we will of course have more constraints on the choice of the templates. We will see, thanks to the following theorem, that the additional constraint is to impose $I = 0$.

6.2 Theorem 2

For all $1 \leq i \leq M$, $1 \leq j \leq N$, let $x_{ij}(t)$ be the trajectory of the state corresponding to the initial condition $x_{ij}(0)$ and input u_{ij}. Denote by $x'_{ij}(t)$ the trajectory of the state corresponding to the initial condition $x'_{ij}(0) = -x_{ij}(0)$ and input $u'_{ij} = -u_{ij}$, for all $1 \leq i \leq M$, $1 \leq j \leq N$. Then $x'_{ij}(t) = -x_{ij}(t)$ for all $t \geq 0$ if and only if $I = 0$.

Proof: First, let us repack the state variables x_{ij} into an $MN \times 1$ vector $\mathbf{x}$ and the input variables u_{ij} into $\mathbf{u}$ (using the same ordering as that for $\mathbf{x}$). Then the state equation can be rewritten as

$$\dot{\mathbf{x}}(t) = -\mathbf{x}(t) + A_m\, \mathbf{f}(\mathbf{x}(t)) + B_m\, \mathbf{u} + \mathbf{I}.$$

Here $f_i(\mathbf{x}) = f_i(x_i) = f(x_i)$. The diagonal elements of the $MN \times MN$ matrices A_m and B_m are respectively $A_{0,0}$ and $B_{0,0}$, and their off-diagonal elements are either zeros or off-diagonal elements of the templates A and B. All the components of the vector $\mathbf{I}$ are equal to I. Similarly,

$$\dot{\mathbf{x}}'(t) = -\mathbf{x}'(t) + A_m\, \mathbf{f}(\mathbf{x}'(t)) + B_m\, \mathbf{u}' + \mathbf{I}.$$

Define

$$\mathbf{s}(t) = \mathbf{x}(t) + \mathbf{x}'(t).$$

Since $\mathbf{u} = -\mathbf{u}'$,

$$\dot{\mathbf{s}}(t) = \dot{\mathbf{x}}(t) + \dot{\mathbf{x}}'(t) = -(\mathbf{x}(t) + \mathbf{x}'(t)) + A_m\,(\mathbf{f}(\mathbf{x}(t)) + \mathbf{f}(\mathbf{x}'(t))) + 2\mathbf{I}. \quad (35)$$

($\Rightarrow$) Suppose first that for all $1 \leq i \leq M, 1 \leq j \leq N$ and all $t \geq 0$, $x'_{ij}(t) = -x_{ij}(t)$, or equivalently that $\mathbf{x}'(\cdot) = -\mathbf{x}(\cdot)$. Then $\mathbf{s}(\cdot) = \mathbf{0}$, and since $\mathbf{f}(\mathbf{x}') = \mathbf{f}(-\mathbf{x}) = -\mathbf{f}(\mathbf{x})$, it follows from (35) that $\mathbf{I} = \mathbf{0}$, and thus that $I = 0$.

($\Leftarrow$) Next, suppose that $\mathbf{I} = \mathbf{0}$. Since $f(x_i) \leq x_i$ with $f(\cdot)$ given in figure 1, it follows then from (35) that

$$\|\dot{\mathbf{s}}(t)\| \leq \|(A_m - U)(\mathbf{x}(t) + \mathbf{x}'(t))\| \leq \|A_m - U\|\ \|\mathbf{s}(t)\|$$

where U is the identity matrix. Noting that $\mathbf{s}(0) = \mathbf{0}$ for $\mathbf{x}'(0) = -\mathbf{x}(0)$, this inequality implies that

$$\|\mathbf{s}(t)\| \leq \|\mathbf{s}(0)\|\, e^{\|A_m - U\|\, t} = 0.$$

Hence $\mathbf{x}(\cdot) = -\mathbf{x}'(\cdot)$, and $x_{ij}(\cdot) = -x'_{ij}(\cdot)$ for all $1 \leq i \leq M$, $1 \leq j \leq N$. □

6.3 Cloning template for a Hole-Filler CNN

It follows from the previous theorem that templates solving the first problem stated in subsection 6.1 will solve the second problem as well if $I = 0$. We can restate problem 1 as a set of local rules applying to a particular cell C_{ij} ($1 \leq i \leq M$, $1 \leq j \leq N$) and to the cells horizontally and vertically adjacent to C_{ij}. By a similar reasonning as the two previous examples, we come up with templates assuming the form

$$A = \begin{bmatrix} 0 & A_n & 0 \\ A_n & A_{0,0} & A_n \\ 0 & A_n & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & B_{0,0} & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad I = 0$$

where the parameters $A_{0,0}$, A_n and $B_{0,0}$ satisfy the inequalities

$$\begin{aligned} A_{0,0} - 1 - 4A_n + B_{0,0} &> 0 \\ A_{0,0} - 1 + 4A_n - B_{0,0} &> 0 \\ 1 - A_{0,0} - 3A_n + B_{0,0} &> 0 \\ A_{0,0} - 1 &> 0 \\ A_n &> 0. \end{aligned} \tag{36}$$

For instance, we can take $A_{0,0} = 2.5$, $A_n = 1$ and $B_{0,0} = 5$, that is, the templates

$$A = \begin{bmatrix} 0 & 1.0 & 0 \\ 1.0 & 2.5 & 1.0 \\ 0 & 1.0 & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 5.0 & 0 \\ 0 & 0 & 0 \end{bmatrix}; \quad I = 0.$$

7 Conclusion

We have presented a method for synthetizing a CNN to solve a given problem. The first step consists in the translation of the problem into a set of local rules, involving a cell and its nearest neighbors only. The second step is the implementation of these rules with a cloning template. By applying corollary 2, we get then a system of inequalities which must be satisfied by the parameters of the templates to ensure a correct operation of the CNN.

This method can also provide us with some insight on the sensitivity of the templates to small variations around their nominal value. Indeed, we will try to choose the parameters defining the templates so that they always satisfy the set of inequalities derived in the second step, even if their values are slightly perturbed. This point is relevant because of the unavoidable lack of accuracy of the VLSI realization of a CNN.

The method described in this paper can be extended to multiple layers CNNs, and can therefore be applied to a broader class of problems. Nevertheless, its application is restricted to relatively simple problems, which can be formulated within a reasonable number of local "rules".

Acknowlegments

This work was supported in part by the Belgian American Educational Foundation, by the office of Naval Research under Grant number N000-14-86K-

0351 and by the National Science Foundation under Grant MIP-86-14000.

References

[1] L.O. Chua and L. Yang, 'Cellular Neural Networks: Theory,' *IEEE Transactions on Circuits and Systems*, CAS-35, 1257–1272 (1988).

[2] L.O. Chua and L. Yang, 'Cellular Neural Networks: Applications,' *IEEE Transactions on Circuits and Systems*, CAS-35, 1272–1290 (1988).

[3] F. Zou, S. Schwartz and J. Nossek, 'Cellular Neural Network Design Using a Learning Algorithm', *Proceedings of the IEEE International Workshop on Cellular Neural Networks and their Applications*, Budapest, 73–81 (1990).

[4] K. Ślot, 'Determination of Cellular Neural Network Parameters for Feature Detection of Two-dimensional Images', *Proc. IEEE Int. Workshop on Cellular Neural Networks and their Applications*, Budapest, 82–91 (1990).

[5] L.O. Chua and T. Roska, 'Stability of a class of Nonreciprocal Cellular Neural Networks', *IEEE Transactions on Circuits and Systems*, CAS-37, 1520–1527 (1990).

[6] L.O. Chua and B. Shi, 'Multiple Layer Cellular Neural Networks : A Tutorial', Memo. UCB/ERL M90/113, Univ of California at Berkeley, Electronic Research Laboratory (1990).

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

[8] V. Lakshmikantham and S. Leela, '*Differential and Integral Inequalities*', Academic Press, vol. 1, New York, 1969.

[9] G. Birkhoff and G.C. Rota, '*Ordinary Differential Equations*', Wiley and sons, 4th edition, New York, 1989.

[10] T. Matsumoto, L.O. Chua and H. Suzuki, 'CNN Cloning Template: Shadow Detector', *IEEE Transactions on Circuits and Systems*, CAS-37, 1070–1573 (1990).

[11] D. H. Hubel and T.N. Wiesel, 'Receptive fields, binocular interaction and functional architecture in the cat's visual cortex', *Journal of Physiology*, 160, 106–154 (1962).

[12] T. Roska, T. Boros, P. Thiran and L.O. Chua, 'Detecting Simple Motion using Cellular Neural Networks', *Proc. IEEE Int. Workshop on Cellular Neural Networks and their Applications*, Budapest, 127–138 (1990).

[13] T. Matsumoto, L.O. Chua and R. Furukawa, 'CNN Cloning Template: Hole-Filler', *IEEE Transactions on Circuits and Systems*, CAS-37, 635–638 (1990).

List of figures

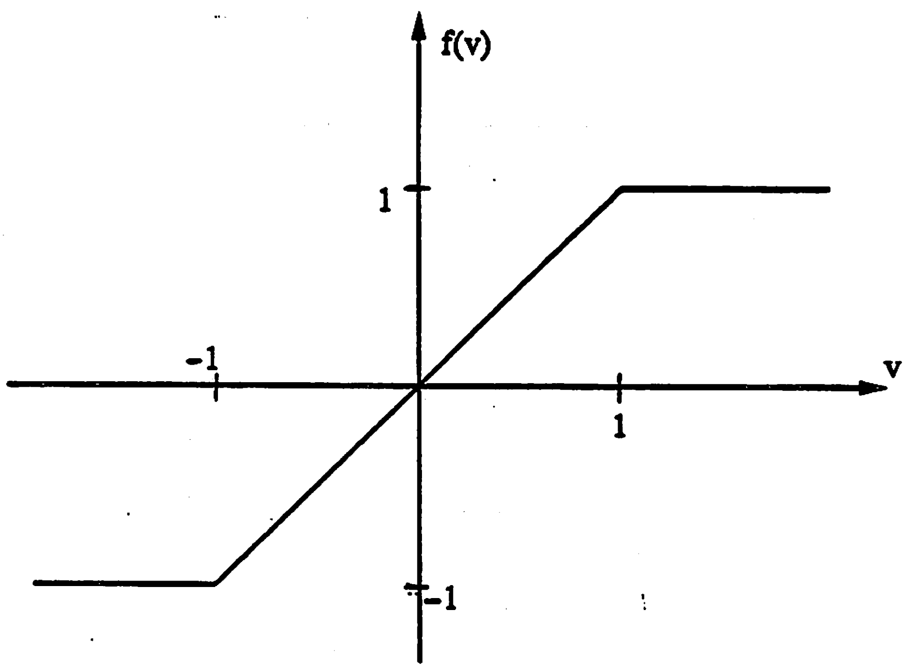


Figure 1: The piecewise-linear sigmoid function $f(\cdot)$.

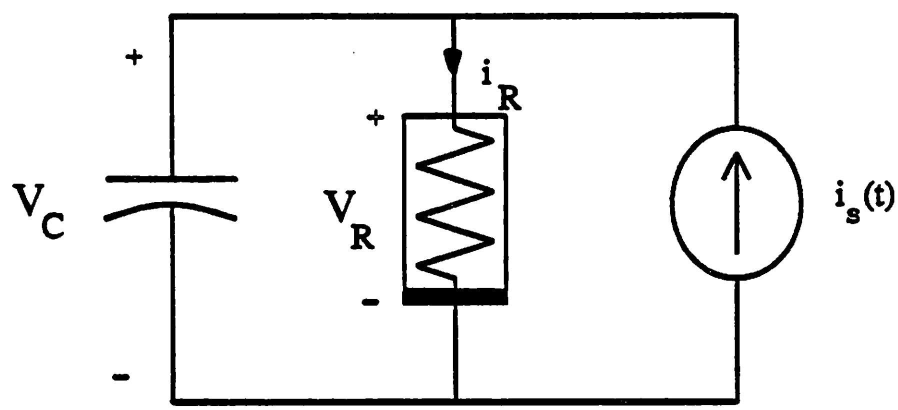


Figure 2: The equivalent circuit of a cell.

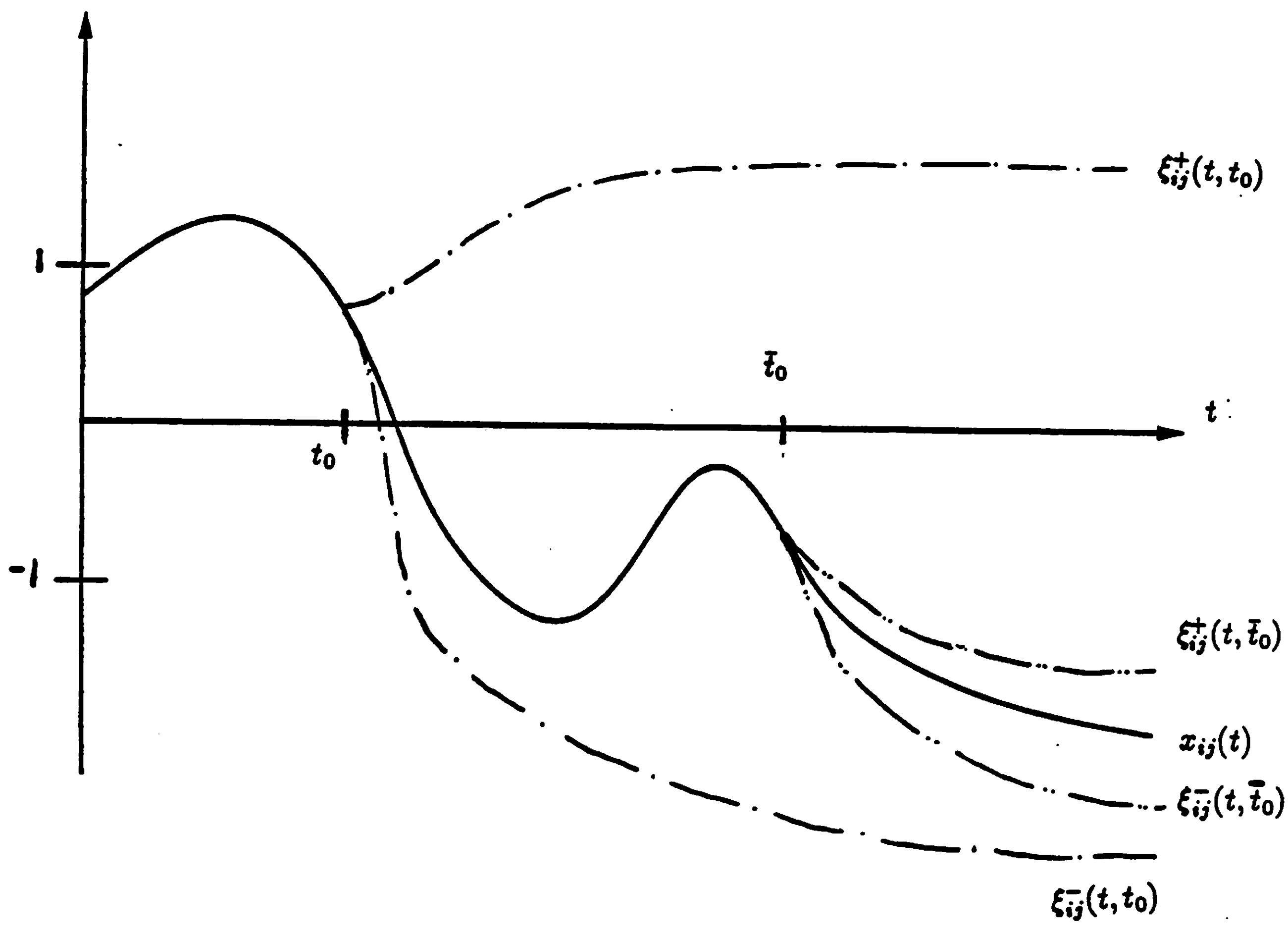


Figure 3: The waveform of the state of a cell, and two sets of upper and lower bounds. We cannot predict the steady state from the bounds evaluated at $t = t_0$, but we can conclude in $\bar{t}_0$ that $y_{ij}(\infty) = -1$.

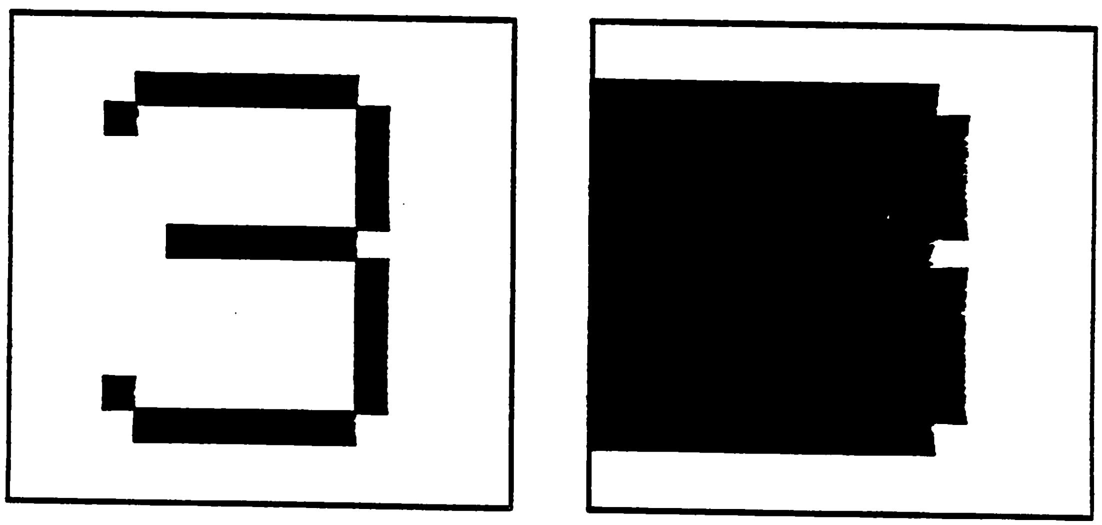


Figure 4: Creation of the shadow of an object: the given image (left) and the output of the CNN (right).

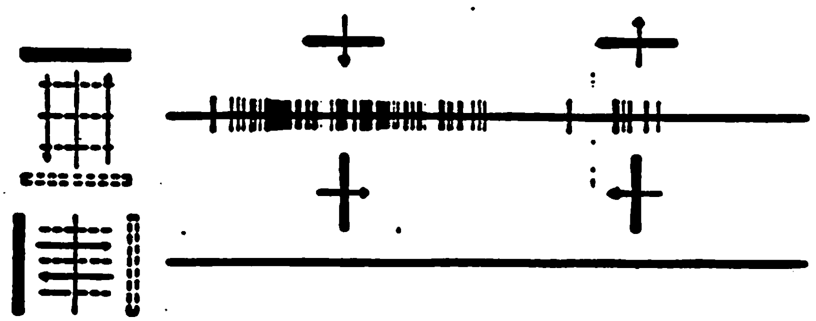
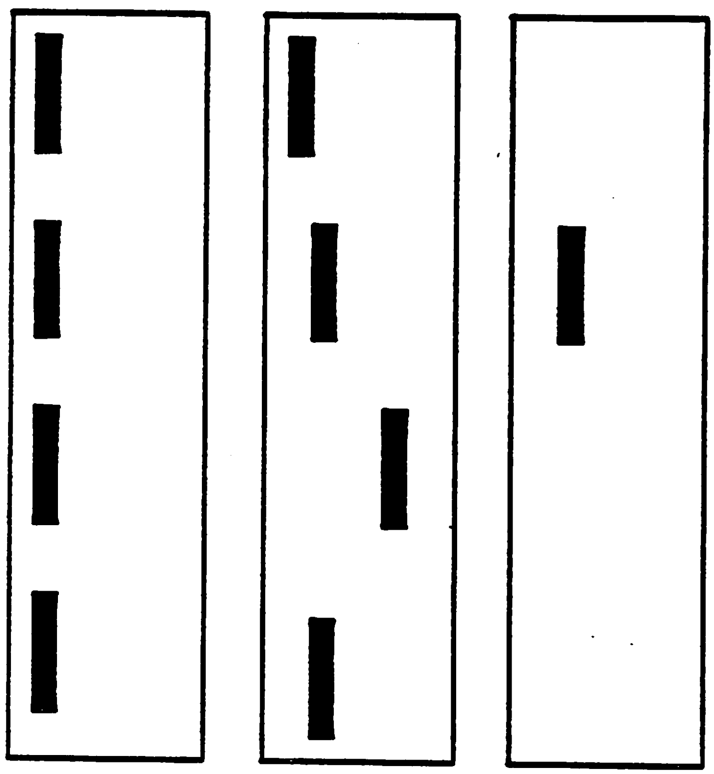


Figure 5: Responses of a cat's cortical cell sensitive to downward movement only (from [7], p. 121).



(a) (b) (c)

Figure 6: Motion detection: the images $P(k\Delta t)$ (a) and $P((k+1)\Delta t)$ (b), applied respectively to the input and the initial states, and the output of the CNN (c).

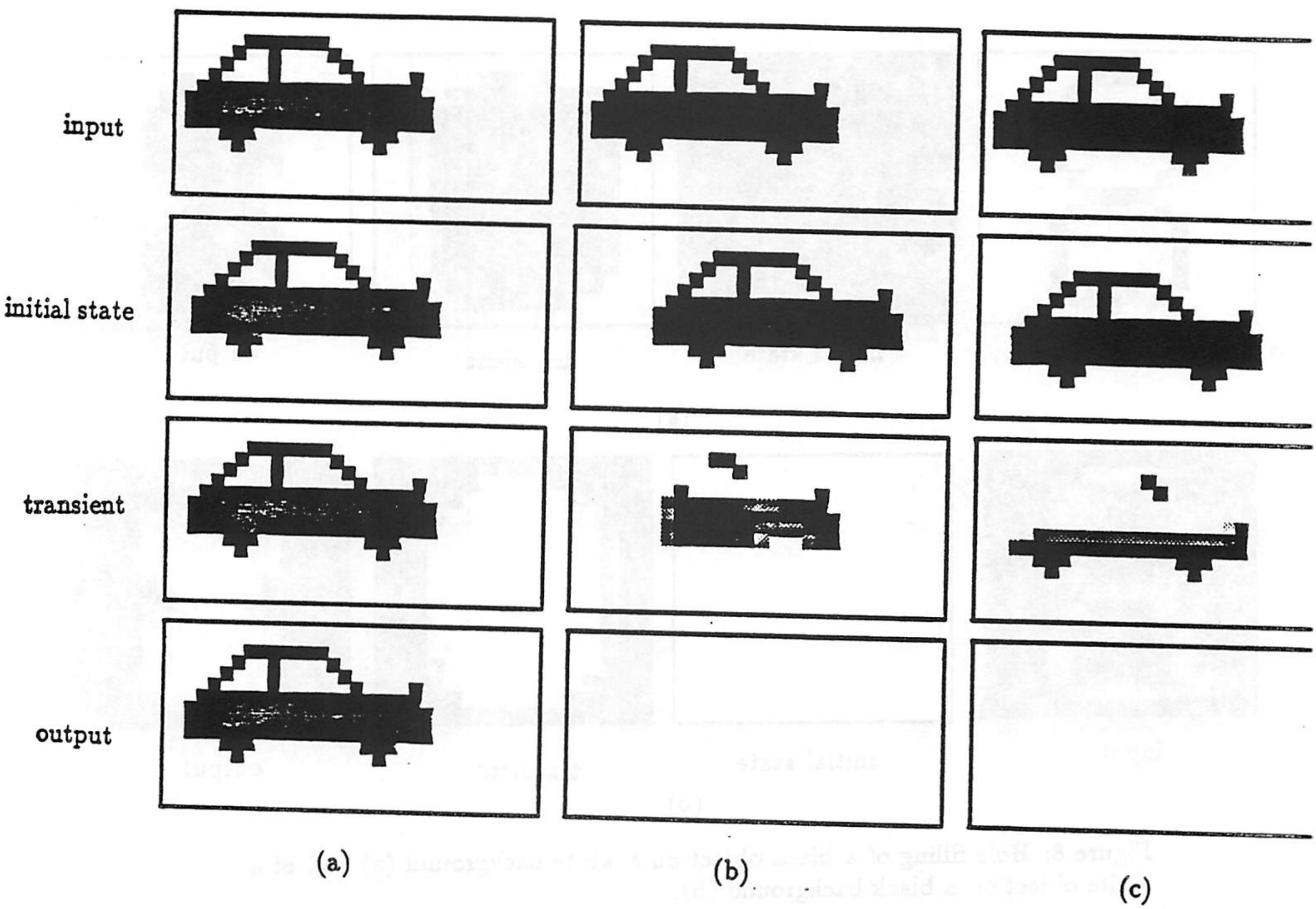


Figure 7: Motion detection: the moving car is erased from the screen if it moved with a "wrong" speed (b) or in a "wrong" direction (c), and remains only in the case of a movement et the right speed in the right direction (a).

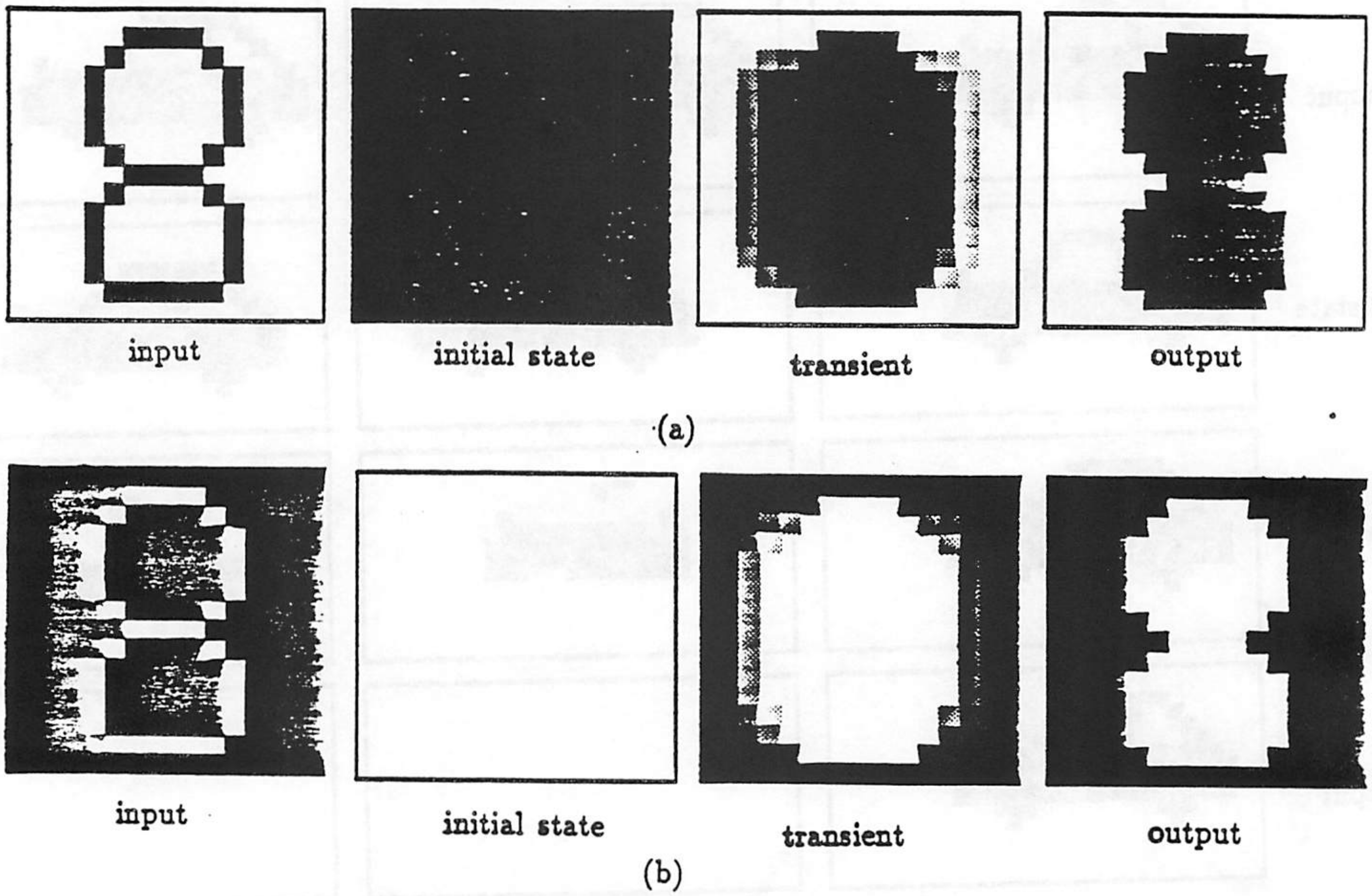


Figure 8: Hole filling of a black object on a white background (a) and of a white object on a black background (b).