2010 International Symposium on Nonlinear Theory and its Applications
NOLTA2010, Krakow, Poland, September 5-8, 2010

Can you achieve any function with a 2-neuron CNN?

Mireia Vinyoles-Serra† and Xavier Vilasís-Cardona†

†LIFAELS, laSalle, Universitat Ramon Llull
Pg. Bonanova 8, 08022 Barcelona, Spain
Email: mireiav@salle.url.edu, xvilasis@salle.url.edu

Abstract—We recover the analysis of the response of the 2-neuron CNN to its external inputs in the stable symmetric case. From this we study which binary functions can be implemented either directly, either by composing templates. The results show that some particular combinations can not be achieved. One of this functions is found when implementing a universal Turing machine header.

1. Introduction

Cellular Neural Networks (CNNs), as introduced in [1], [2] are nonlinear dynamical systems completely stable for certain parameter range. Their behavior is defined by the CNN parameters usually called cloning template yet sometimes a single cloning template is not enough to solve a particular problem. In this case, a template combination can be used. But the necessary templates in order to reproduce some input-output functional relations is not clear yet. So we study the simplest but rich case: the two neuron CNN.

Taking a symmetric set of weights, and the self-feedback coefficient larger than one, the state variables always converge to ± 1. This stability results allow to establish relations between the CNN parameters and the final outputs [3], and so aboard the template design and template composition problems. From this relations we set which combinations are possible with a single template, composing templates and which are unreachable. For instance, we shall see how one of the state transitions of the 4-symbol, 7-state universal Turing machine [4] is impossible to be implemented with a two neuron CNN.

2. Convergence map

Focusing our study in the two neuron Cellular Neural Network, we first define our notation for the piecewise linear CNN system as:

$$\begin{cases} \dot{x_0} = -x_0 + sy_0 + p_+ y_1 + b_0 u_0 + b_+ u_1 + I, \\ \dot{x_1} = -x_1 + sy_1 + p_- y_0 + b_- u_0 + b_0 u_1 + I, \end{cases} \tag{1}$$

where x_i are the internal states of the neuron and are taken in $[-1, 1]$. Variables y_i are the external states defined by the piecewise linear function, $f(x_i) = \frac{1}{2}(|x_i(t) + 1| - |x_i(t) - 1|)$, $i = 0, 1$. The external inputs are u_i and they shall be constant in time. Our analysis uses u_i's as the input variables and the final stable state y_i's as output. The other parameters $(s, p_+, p_-, b_0, b_+, b_-, I)$ configure the network cloning template. Along the paper, we will use the notation,

$$\begin{pmatrix} u'_0 \\ u'_1 \end{pmatrix} = \begin{pmatrix} b_0 & b_+ \\ b_- & b_0 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \end{pmatrix} + \begin{pmatrix} I \\ I \end{pmatrix} \tag{2}$$

so that, the action of B on to (u_0, u_1) to obtain (u'_0, u'_1) will be called *B-transformation*.

In order to study the template influence on the CNN function, we must work in a parameter range where the system converges to a fixed-point. From Lyapunov theory, it is known that for $s > 1$ and $p_+ = p_- = p$, the system converges to one of the four corner points $\mathcal{S} = \{(\pm 1, \pm 1)\}$. This particular convergence set allows to aboard classification problems using the Lyapunov function defined as

$$L(y_0, y_1) = -p y_0 y_1 - \frac{s-1}{2}(y_0^2 + y_1^2) - u'_0 y_0 - u'_1 y_1. \tag{3}$$

$L(y_0, y_1)$ is a monotone decreasing function and bounded from below, so the CNN system converges to the point where $L(\pm 1, \pm 1)$ is minimum [1]. The comparison of this four values gives us the necessary convergence conditions and let's choose the adequate CNN parameters in order to guarantee some desired input-output relation. To do it, we first fix the initial conditions at $(0, 0)$ and then compare the four possible output values of $L(\pm 1, \pm 1)$ in order to find in which L takes lower value.

The Lyapunov function (3) takes the minimum value at $(+1, +1)$, this is $L(+1, +1) \leq L(i, j) \quad i, j = 1, -1$ if and only if parameters fulfill equations: $u'_0 \geq -p$, $u'_1 \geq -p$ and $u'_0 + u'_1 \geq 0$. Plotting this region in the (u'_0, u'_1)-plane we obtain a convergence map and then applying the B-transformation (2) in order to find the correspondent region in the (u_0, u_1)-plane, we obtain a convergence map in Figure 1.

From these maps, we may fix the parameters of the CNN to obtain the relation between the external inputs and the final outputs. Taking inputs u_i inside a convergence region where $L(i, j)$ is minimum, the system will converge the output value (i, j) in $\mathcal{S}$. The shape of regions $L(i, j)$ depends only on four parameters: two slopes and two intersection points, instead of the six apparent free parameters in equation (1). Knowing for example the slopes of the external lines $m_0 = -b_-/b_0$, $m_1 = -b_0/b_+$, the slope of the line

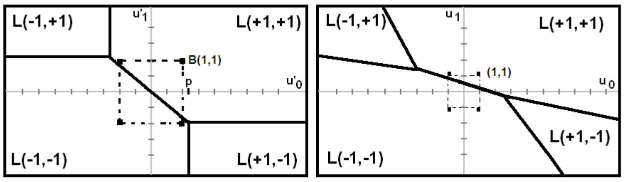

Figure 1: Convergence maps in the $\{u'_0, u'_1\}$-plane and in the (u_0, u_1)-plane for $p > 0$.

connecting them is already determined and, knowing the two intersection points which depend on m_0, m_1, I and p, we complete the construction of the convergence map. For instance, on Table 1 we can see the line equations defining the boundaries of the different convergence maps for a positive value of parameter p.

$\{u_0, u_1\}$ plane	$\{u'_0, u'_1\}$ plane
$(x_{-p}, y_p) = \left(\frac{(p-I)-m_1(I+p)}{b_0(m_1-m_0)}, \frac{m_1((p-I)-m_0(p+I))}{b_0(m_1-m_0)}\right)$	$(-p, p)$
$(x_p, y_{-p}) = \left(\frac{m_1(p-I)-(I+p)}{b_0(m_1-m_0)}, \frac{m_1(m_0(p-I)-(I+p))}{b_0(m_1-m_0)}\right)$	$(p, -p)$
$(u_1 - y_p) = m_1(u_0 - x_{-p})$	$u'_1 = p$
$(u_1 - y_p) = m_0(u_0 - x_{-p})$	$u'_0 = -p$
$(u_1 - y_{-p}) = m_1(u_0 - x_p)$	$u'_1 = -p$
$(u_1 - y_{-p}) = m_0(u_0 - x_p)$	$u'_0 = p$
$(u_1 - y_p) = m_1\left(\frac{1-m_0}{1-m_1}\right)(u_0 - x_{-p})$	$u'_1 + u'_0 = 0$

Table 1: Intersection points and boundary lines of the convergence regions for $p > 0$.

From all this study, we also note that we can make the system converge where we want if we compose different templates. So for example, composing two templates, the first one will drive the system to one of the four points $y_i \in \mathcal{S}$. Using this points as the external inputs $u_i = y_i \in \mathcal{S}$ for the second template, we construct another convergence map to make each of this new inputs correspond to a new output.

3. Fixing the CNN parameters

The particular shape of the convergence map seems to limit the kind of problems which can be solved using a two neuron CNN. In order to determine whether this limitation is apparent or real we shall examine the repeated action of the cloning template. For this we can restrict the input choice (u_0, u_1) to ± 1 without loss of generality. Using the B-transformation (2) we shall find their images $B(\pm 1, \pm 1)$, and the necessary parameter conditions to place them into a pre-established convergence region in the $\{u'_0, u'_1\}$-plane (Figure 1). This may done by studying if each one of the image points $B(i, j)$, $i, j = \pm 1$, are equal to one of the four possible outputs $\mathcal{S}$ located in each of the four different convergence regions. If the output points are located on a boundary line dividing different convergence regions, we shall translate the input point $(i, j) + (\varepsilon, \varepsilon)$, $i, j = -1, 1$, $\varepsilon \neq 0$, and proceed as we have explained before. To simplify the notation, let's rename $\mathcal{S}$-points with the correspondence shown in Table 2, and the convergence regions $L(i, j)$ will then be $R(k)$, for $k = 1, 2, 3, 4$.

$(1, 1) \equiv 1$	$(1, -1) \equiv 2$	$(-1, -1) \equiv 3$	$(-1, 1) \equiv 4$

Table 2: $\mathcal{S}$-points correspondence.

For example, let us study the convergence of input 1 for a positive parameter p. If we take 1 coverging to itself, $B(1) = (1, 1)$, this condition implies that for $p > 0$, B-parameters fulfill:

$$\{b_0 + b_+ + I = 1, \quad b_- + b_0 + I = 1\}. \tag{4}$$

Next we consider the different outputs where input 3 can converge. If $B(3) = (-1, 1)$, we have

$$\{-b_0 - b_+ + I = -1, \quad -b_- - b_0 + I = 1\}. \tag{5}$$

Solving the system equations (4) and (5), parameter I must be equal to 0 and 1. Therefore, such an association can not be achieved by one single 2-neuron CNN.

However, if $B(3) = (1, 1)$, we have

$$\{-b_0 - b_+ + I = 1, \quad -b_- - b_0 + I = 1\}. \tag{6}$$

Solving the system equations (4) and (6), we find parameters $I = 1, b_+ = b_- = -b_0$. This relation is compatible with a 2-neuron CNN. Now we study the four possible outputs for inputs 2 and 4 where $B(2) = (1 - 2b_0, 1 + 2b_0)$ and $B(4) = (1 + 2b_0, 1 - 2b_0)$, summarized in Table 3.

$(1, -1)$	$(-1, 1)$	parameter conditions
$R(1)$	$R(1)$	$p > \max\{-1 - 2b_0, -1 + 2b_0\}$
$R(2)$	$R(4)$	$p < -1 + 2b_0$
$R(3)$	$R(3)$	$\times$
$R(4)$	$R(2)$	$p < -1 - 2b_0$

Table 3: Convergence study for points $(-1, 1)$ and $(1, -1)$.

Let us note that the parameter conditions are incompatible for certain values of b_0. Using the parameter conditions shown in Table 3, we obtain that input point 2 can converge to outputs $(1, 1), (1, -1)$ depending on parameters p, b_0 (7). Otherwise for $0 < b_0 < 1/2$ and $p > 0$, input point 2 can only converge to $(1, 1)$.

$$\begin{cases} B(2) = (1, 1) \Leftrightarrow p > -1 + 2b_0, b_0 > 1/2, \\ B(2) = (1, -1) \Leftrightarrow p < -1 + 2b_0, b_0 > 1/2. \end{cases} \tag{7}$$

Like in the first case, input 3 can not converge to $(1, -1)$ because there is no solution for the system equations obtained from (4) and $B(3) = (1, -1)$. Finally, input 3 converges to itself for parameters, $I = 0$ and $b_+ = b_- = 1 - b_0$. The other square point B-images are $B(2) = (-1 + 2b_0, 1 -$

$2b_0) = -B(4)$ and lay on a boundary line of the convergence map. In this case, we apply a translation to the image points in order to solve $B(\pm1,\pm1) = (\pm1+\varepsilon, \pm1+\varepsilon)$, $\varepsilon \in \mathbb{R} - \{0\}$.

From the equations obtained, we find parameters $I = \varepsilon, b_+ = b_- = 1 - b_0$. The other square point B-images are then $B(2) = (-1+2b_0+\varepsilon, 1-2b_0+\varepsilon)$ and $B(4) = (1-2b_0+\varepsilon, -1+2b_0+\varepsilon)$. Using the convergence map, $B(2) = (1,1)$ if and only if conditions in (8) are fulfilled.

$$\begin{cases} \pm(-1+2b_0)+\varepsilon > -p \\ 1-2b_0+\varepsilon \geq 1-2b_0-\varepsilon \Rightarrow \varepsilon \geq 0 \end{cases} \tag{8}$$

Parameter conditions are then $p > \max\{\pm(-1+2b_0) - \varepsilon\}$. Doing a similar study for the other convergence regions we obtain the rest.

Using similar arguments we find all possible output values for the case where the first input point 1 converges to itself and $p > 0$ in Table 4. We use the two row notation in order to describe the rearrangement of the input-output relations obtained.

input-output	parameter conditions
$\begin{pmatrix} \mathbf{1} & 2 & 3 & 4 \\ \mathbf{1} & 1 & 1 & 1 \end{pmatrix}$	$p > \max\{-1-2b_0, -1+2b_0\}$
$\begin{pmatrix} \mathbf{1} & 2 & 3 & 4 \\ \mathbf{1} & 2 & 1 & 4 \end{pmatrix}$	$0 < p < -1+2b_0$
$\begin{pmatrix} \mathbf{1} & 2 & 3 & 4 \\ \mathbf{1} & 4 & 1 & 2 \end{pmatrix}$	$0 < p < -1-2b_0$
$\begin{pmatrix} \mathbf{1} & 2 & 3 & 4 \\ \mathbf{1} & 4 & 3 & 2 \end{pmatrix}$	$0 < p < \min\{-1+2b_0 \pm \varepsilon\}$
$\begin{pmatrix} \mathbf{1} & 2 & 3 & 4 \\ \mathbf{1} & 2 & 3 & 4 \end{pmatrix}$	$0 < p < \min\{1-2b_0 \pm \varepsilon\}$
$\begin{pmatrix} \mathbf{1} & 2 & 3 & 4 \\ \mathbf{1} & 1 & 3 & 1 \end{pmatrix}$	$p > \max\{\pm(-1+2b_0)-\varepsilon\}, \varepsilon > 0$
$\begin{pmatrix} \mathbf{1} & 2 & 3 & 4 \\ \mathbf{1} & 3 & 3 & 3 \end{pmatrix}$	$p > \max\{\pm(-1+2b_0)-\varepsilon\}, \varepsilon < 0$

Table 4: Possible outputs for the case $B(1) = (1,1)$, $p > 0$.

Now, from the study of all the direct input-output relations, we have found 25 possible convergence options with their correspondent templates. Let's note that the parameter conditions in order to reproduce a desired input-output relation, are determined by p and b_0. The rest, b_+ and b_-, depend in each particular case, on b_0. From the different system equations, parameter I gives us the key point in order to discuss the existence of a solution.

Moreover, composing this different templates, we obtain all the possible relation between the four points in $\mathcal{S}$ using a two neuron CNN. This relation can be classified with those converging to one, two, three or four different outputs.

For example, there are four elements T_i, $i = 1, \ldots, 4$ converging to a single output. A template T_1 making the system converge to output 1, can be defined for parameters $I = 1$, $b_+ = b_- = -b_0$, $p > \max\{-1 \pm 2b_0\}$ and $s > 1$. Choosing $b_0 = 2$, $p = 4 > 3$ and $s = 3$ we find $(s, p, b_0, b_+, b_-, I) = (3, 4, 2, -2, -2, 1)$. Composing this template with another one T_j where input 1 converges to 2 we obtain $T_2 = T_j \circ T_1$.

$$T_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{pmatrix}\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 2 & 2 & 2 \end{pmatrix}$$

Let's note that using these results, a two neuron CNN can also realize Boolean functions $F : \mathbb{R}^2 \to \mathbb{R}$ defined for example as $F(u_0, u_1) = y_0(\infty)$, $(u_0, u_1) \in \mathcal{S}$ like in [6]. Using a single template, linearly separable Boolean funcions can be solved while for the rest, template composition must be used.

4. Impossible relations

In the remaining of the paper we shall focus on the only but bijective input-output relations summarized in Table 5. From p_1 to p_8, they are obtained by the action of a single template except p_3 and p_8 which come from the composition of two templates.

inputs	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
1	1	1	2	2	3	3	4	4
2	2	4	1	3	2	4	1	3
3	3	3	4	4	1	1	2	2
4	4	2	3	1	4	2	3	1

Table 5: Cases where the CNN converges to four different outputs.

In this particular case, p_i can be written as permutations of four different objects: the input points in $\mathcal{S}$.

Remark that we have found only eight bijective relations, while using four elements $\mathcal{S}$, we should find the set of all possible permutations, the symmetric group S_4 of $4! = 24$ elements. To shed light in the number of different templates which perform a functional relation between all the four elements, we compose the eight ones described in Table 5.

Let's first rewrite the eight permutation templates p_i using the cycle notation, and compose them.

$$\begin{array}{llll} p_1 = Id & p_3 = (12)(34) & p_5 = (13) & p_7 = (1432) \\ p_2 = (24) & p_4 = (1234) & p_6 = (13)(24) & p_8 = (14)(23) \end{array} \tag{9}$$

The result of all the composition templates represented by product permutations is shown in Table 6.

Let's note that we have found a special subset of group S_4 that fulfill the group properties, this is a subgroup.

With this results we set a *Convergence Lemma*. Let's consider a two neuron CNN defined by equations (1) where parameters fulfill $s > 1$ and $p_+ = p_- = p$. Let's name $\mathcal{S} = \{(\pm1, \pm1)\}$ the four possible output values set where the CNN can converge. There exist only eight different cases where the CNN system converges to the four different outputs $\mathcal{S}$ summarized in Table 5.

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
p_1	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
p_2	p_2	p_1	p_7	p_8	p_6	p_5	p_3	p_4
p_3	p_3	p_4	p_1	p_2	p_7	p_8	p_5	p_6
p_4	p_4	p_3	p_5	p_6	p_8	p_7	p_1	p_2
p_5	p_5	p_6	p_4	p_3	p_1	p_2	p_8	p_7
p_6	p_6	p_5	p_8	p_7	p_2	p_1	p_4	p_3
p_7	p_7	p_8	p_2	p_1	p_3	p_4	p_6	p_5
p_8	p_8	p_7	p_6	p_5	p_4	p_3	p_2	p_1

Table 6: Template composition for all the permutations p_i founded in a two neuron CNN.

One example where we can see this restrictions is trying to implement Minsky's 7-state 4-color universal Turing machine illustrated in Figure 2.

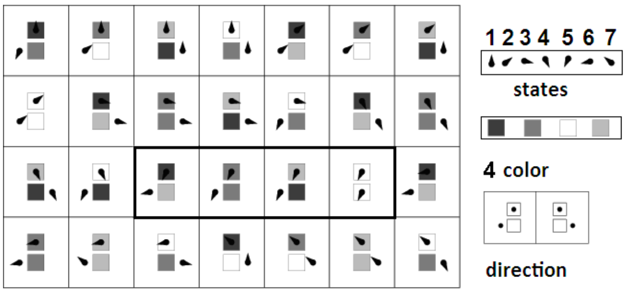

Figure 2: Generalization of Minsky's 7-state 4-color universal Turing machine made by Macura [4].

We want the two neuron CNN to represent the header action of the Turing machine on the tape so we shall use the values ± 1 to code the colors. Since a symmetric two neuron CNN has four possible outputs for $s > 1$, it can be used to modify the color of an active cell of the Minsky's universal Turing machine. The input color will be coded on the u_i's while the output color will be obtained from the final state y_i's of the neurons (Figure 3).

Figure 3: One correspondence between the Turing machine colors and the four possible states of the 2-neuron CNN.

In this way, each state of the machine corresponds to a template or to a combination of templates relating the four possible input symbols to their correspondent output ones. To design these templates we shall use the convergence map studied before. Taking the fifth state (Figure 2) which can be written as $s_5 = (34)$, we see from the convergence lemma that this case can not be performed using a two neuron CNN because it is not one of the eight possible permutations found (9). Of course, this result seem to depend on the choice between colors and $\mathcal{S}$-points. State s_5 is the only permutation of colors so one may think on choosing a diferent relation between $\mathcal{S}$-points and the Turing machine colors to implement the state. However, in all the cases, the associations fulfilling s_5, do no fulfill some other state s_i, $i \neq 5$, $i = 1, \ldots, 7$, where the system converges to three different outpus. So, a 2-neuron CNN can not be used to reproduce this header action of the universal Turing machine.

5. Conclusions

We have seen that a 2-neuron CNN can not perform all logical bijective function but only a subgroup. The particular geometry of the convergence map limits the kind of problems which can be solved yet, allows to classify them into those converging to one, two, three or four different outputs. In this way, we may know if a specific problem, can be solved using a two neuron CNN. In the line of [6], we have checked that all single-output Boolean functions can be reproduced. Linearly separable ones, just need one template while non linearly separable ones require the composition of two. We have also seen as an example the problem to reproduce the header action of the universal Turing machine.

From [5] we know that a CNN is a universal Turing machine in higher dimensions. So this leads to the discussion on which is the minimal CNN being a universal Turing machine. Moreover, using $y_0(\infty)$ as output, we can reproduce any Boolean function just like a universal CNN cell but, using $(y_0(\infty), y_1(\infty))$ as output we loose universality.

Acknowledgments

This work is financially supported by FUNITEC.

References

[1] L. O. Chua, L. Yang "Cellular Neural Networks: Theory," *IEEE transactions on Circuits and systems*, vol 35, 1998.

[2] L. O. Chua, "CNN: a Paradigm for Complexity," *World Scientific*, 1998.

[3] X. Vilasis, M. Vinyoles "On cellular neural network learning," *Proceedings of ECCTD*, Cork, Ireland, 2005.

[4] Weisstein, Eric W. "Turing Machine." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/TuringMachine.html

[5] L. O. Chua, T. Roska, P. Venetianer "*The CNN is universal as the Turing machine*, IEEE Trans. Circuits Syst. I, vol 40, no. 4, pp 289-291, 1993.

[6] Dogaru, R., Chua, L. "Universal CNN cells",*Int. Journal of Bifurcations and Chaos*, vol 9, No 1, pp 1-48 1999.