Universitat Ramon Llull

TESI DOCTORAL

Títol

Dynamics of Two Neuron Cellular Neural Networks

Realitzada per Mireia Viñoles Serra

en el Centre La Salle

i en el Departament Electrònica

Dirigida per Xavier Vilasís Cardona

C. Claravall, 1-3
08022 Barcelona
Tel. 936 022 200
Fax 936 022 249
E-mail: urlsc@sec.url.es
www.url.es

Acknowledgments

Vull donar les gràcies al meu marit per la seva ∞ paciència, per estar allà, per donar-me suport, per cuidar-me cada dia i per un conjunt gairebé no numerable de coses més.
A la meva família, pare, mare, germà i l'Estel. A la Sara. A la meva família política, la Carme, l'Eusebio, la Maika i en Gabi. A la Sònia. Finalment al meu director de tesi, en Xavier Vilasís.

Summary

In this dissertation we review the two neuron Cellular Neural Network stability using the Lyapunov theory, and using the different local dynamic behavior derived from the piecewise linear function. We study then a geometrical way to understand the system dynamics. The Lyapunov stability, gives us the key point to tackle the different convergence problems that can be studied when the CNN system converges to a fixed-point. The geometric stability shed light on the convergence to limit cycles. This work is basically organized based on these two convergence classes.

We try to make an exhaustive study about Cellular Neural Networks in order to find its intrinsic difficulties, and the possible uses of a CNN. Understanding the CNN system in a lower dimension, give us some of the main keys in order to understand the general case. That's why we will focus our study in the one dimensional CNN case with only two neurons.

Based on the Lyapunov function study, we propose some methods to avoid the dependence on initial conditions problem. Its intrinsic characteristics as a quadratic form of the output values gives us the key points to find parameters where the final outputs do not depend on initial conditions. At this point, we are able to study different CNN applications for parameter range where the system converges to a fixed-point. We start by using CNNs to reproduce Bernoulli probability distributions, based on the Lyapunov function geometry. Secondly, we reproduce linear functions while working inside the unit square.

The existence of the Lyapunov function allows us to construct a map, called convergence map, depending on the CNN parameters, which relates the CNN inputs with the final outputs. This map gives us a recipe to design templates performing some desired input-output associations, and drive us into the template composition problem. We study the way different templates can be applied in sequence. From the template design analysis, we may think on finding a functional relation between the external inputs and the final outputs. This correspondence can be thought as a classification problem, because the set of final states is discrete thanks to the piecewise

linear function. Each one of the different classes is defined by the different final states which, will depend on the CNN parameters.

Next, we study which classifications problems can be solved by a two neuron CNN, and relate them with weight parameters. In this case, we also find a recipe to design templates performing these classification problems. The results obtained allow us to tackle the problem to realize Boolean functions using CNNs, and show us some CNN limits trying to reproduce the header of a universal Turing machine.

Based on a particular limit cycle example extracted from Chua's book, we start this study with anti symmetric connections between cells. The results obtained can be generalized for CNNs with opposite sign parameters. We have seen in the stability study that limit cycles have the possibility to exist for this parameter range. Periodic behavior of these curves is computed in a particular case. The limit cycle period can be expressed as a function of the CNN parameters, and can be used to generate clock signals.

Finally, we compare the CNN dynamic behavior using different output functions, hyperbolic tangent, and piecewise linear function. Many times in the literature, hyperbolic tangent is used instead of piecewise linear function because of its differentiability along the plane. Nevertheless, in some particular regions in the parameter space, they exhibit a different number of equilibrium points. Then, for theoretical results, hyperbolic tangent should not be used instead of piecewise linear function.

Contents

1 Introduction **11**
1.1 Framework 11
1.2 CNN dynamics 14
1.3 Outline 18

I Convergence **21**

2 CNN stability **23**
2.1 Two neuron CNN 24
2.2 Bounded states 24
2.3 Lyapunov stability 26
2.4 Geometric stability 30
2.4.1 Symmetric parameter range 31
2.4.2 Non symmetric parameter range 39
2.5 Conclusions 42

II Convergence to a fixed-point **45**

3 Dependence on initial conditions **47**
3.1 Lyapunov function 48
3.2 Dependence problem 51
3.3 Example 53
3.4 Conclusions 56

4 On CNN applications **57**
4.1 Probability distributions 58
4.2 Reproducing linear functions 60
4.3 Template design 61
4.4 Classification problems 73

4.5 Realizing Boolean functions 83
4.6 The header of a universal Turing machine 84
4.7 Conclusions 87

III Convergence to a limit cycle 91

5 Limit Cycles: Antisymmetric Case 93
5.1 Geometry on the CNN. 94
5.2 Limit cycles 95
5.2.1 Searching limit cycles 95
5.2.2 Boundary problem 97
5.2.3 Towards limit cycles 98
5.2.4 Example 99
5.3 Antisymmetric case. 104
5.4 Conclusions 108

6 Limit Cycles: General Case 109
6.1 General case 109
6.2 Conclusions 118

7 Limit cycle period 123
7.1 Computing the period function 123
7.2 Generating clock signals using CNNs 127
7.3 Conclusions 130

IV Convergence comparison 133

8 Topological equivalence 135
8.1 The Chua-Yang and $\tanh x$ models 135
8.2 Topological equivalence 137
8.3 Conclusions 144

V Conclusions 145

9 Conclusions and Future work 147
9.1 Conclusions 147
9.2 Final discussion: The present and the future 150

VI Appendix **153**

A Convergence map **155**

B Input-output relations **165**

C Wolfram's 7-4 Universal Turing machine **189**

D Limit cycles: Antisymmetric Case **191**

E Limit cycles: General case **199**

Bibliography **223**

Chapter 1

Introduction

1.1 Framework

Artificial Neural Networks (ANN) emerged after the introduction of simplified neurons by McCulloch and Pitts in 1943 [45]. These neurons were presented as models of biological neurons and as conceptual components for circuits that could perform computational tasks. Usually, a model artificial neuron as can be seen in Figure 1.1, receives input from a number of other units or external sources, weights every input and adds them up. If the total input is above a threshold, the output of the unit is one; otherwise it is zero. Therefore, the output changes from 0 to 1 when the total weighted sum of inputs is compared to the threshold.

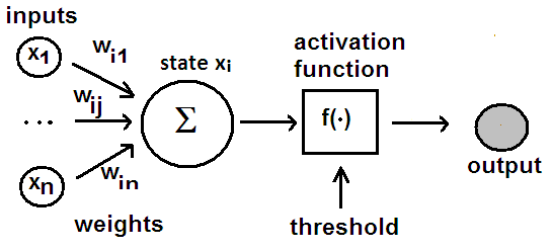


Figure 1.1: Graphical representation of an artificial neuron model.

Mathematically, an Artificial Neural Network is any computational model with the following elements: a state, an activation function, a threshold and the topology defining its neuron connections. The state $x_i(t)$ is an internal variable of the neuron. Its value is defined by the states of neighboring neurons. In the simplest model, a linear combination of these states defines their influence. The coefficients of this linear combination are called weights,

since they weigh the influence of a neuron over another. Such influence can be continuous or discrete, bounded or unbounded.

$$x_i \sim \sum_i w_{ij} x_j$$

An activation function $f(\cdot)$ controls the neuron output amplitude. It must be chosen according to the nature of the neuron state. Usually, an acceptable output range lies between 0 and 1 or -1 and 1. Typical activation functions are step functions, sign functions or logical functions. For continuous valued neurons, linear, hyperbolic tangent or sigmoid functions can be used.

The threshold μ defines the minimal excitation level needed to fire a bi-valuated neuron. It is understood as a term added to the contribution of the neighboring neurons to the activation of a cell. The update rule of the neuron, having n neighbors, is generally given by

$$x_i = f\left(\sum_{j=1}^{n} w_{ij} x_j - \mu\right)$$

The topology describes the pattern connections between the different neurons using the appropriate weights w_{ij}. It has direct implications on the dynamical structure of the neural network.

Artificial Neural Networks have been applied successfully to many fields such as: pattern and sequence recognition, image analysis, function approximation and adaptive control. These are used to construct software agents or autonomous robots. The possibility of learning is what has attracted the most interest in ANN. Given a specific task to solve, learning means using a set of observations to find an algorithm which changes the neuron weights in order to find a solution for the task in some optimal sense. This is one of the greatest advantages of ANNs, and it is particularly useful in applications where the data or task complexity makes the design of such a function by hand highly impractical.

Cellular Automaton (CA) were developed by John von Neumann and Arthur Burks [72] in the 1940s. A cellular automaton is a collection of cells on a grid of a specific shape which evolve through a number of discrete time steps according to a set of rules based on the states of neighboring cells. Formally, a cellular automaton is represented by a lattice, a set of states S, a neighborhood and a local transition function. The lattice is either a finite or infinite discrete regular grid of cells on a finite number of dimensions. Each cell is defined by its discrete position and by its discrete value. The state of a cell at certain discrete time t, is a function of the present state and of the finite neighboring cells N. From the cells surrounding the observed one, the

local transition function $f : S^N \rightarrow S$ calculates the value of a single future cell.

The simplest type cellular automaton is a binary, nearest-neighbor, one-dimensional automaton. Such automaton were named elementary cellular automaton by S. Wolfram [64], who studied their properties. Each cell has rules that depend only on nearest neighbor values, and it can have two possible states 0 or 1. As a result, the evolution of an elementary cellular automaton can be completely described by a table with the state of a cell based on the value of the cell to its left, the cell to its right and its own value (Table 1.1). In the one-dimensional case, there are $2^3 = 8$ possible binary states for the three cells neighboring a given one. There are then $2^8 = 256$ elementary cellular automaton, each of which can be indexed by a unique binary number. The decimal representation of which is known as the rule for the particular automaton. The evolution of a one-dimensional cellular automaton can be illustrated by starting with the initial state in the first row named generation zero, the first generation on the second row and so on. Depending on where the evolution leads (homogeneous configuration, periodic patterns, chaotic patterns or complex structures) rules can be divided into four different classes.

For example, Wolfram's rule 110, like the Game of Life, exhibits a Class 4 behavior, evolving to complex long-lived structures. The Game of Life was invented by John Conway, and popularized by Martin Gardner in a Scientific American article [29]. Table 1.1 describes the different states of a cell for rule 110. The image below (Figure 1.2) shows its history evolution where each pixel is colored white for 0 and black for 1. Matthew Cook proved that some of these structures were rich enough to support universality. In fact, rule 110 has been the basis over which some of the smallest universal Turing machines have been built [16].

current pattern	111	110	101	100	011	010	001	000
new state for center cell	0	1	1	0	1	1	1	0

Table 1.1: Rule 110 of an elementary cellular automaton.

Cellular Neural Networks (CNN) were proposed by Leon O.Chua and L.Yang [5] in 1987. Cellular Neural Networks share the best features of two worlds, adjacent cells are locally connected and so they have a similar structure to Cellular Automaton. On the other hand, they have an asynchronous parallel processing, continuous-time dynamics and global interaction of network elements just like Artificial Neural Networks. Locality allows to overcome a well known drawback for the hardware implementation of artificial

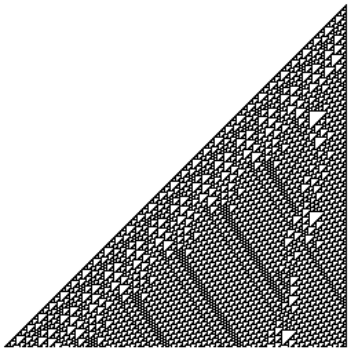


Figure 1.2: History evolution for rule 110 of an elementary cellular automaton.

neural networks. Since electronic units are planar by construction, the number of allowed connections is heavily reduced. The structure of a CNN is particularly suited to be implemented in hardware.

1.2 CNN dynamics

A CNN is any spatial arrangement of locally-coupled cells, where each cell is a dynamical system which has an input, an output, and a state evolving according to some prescribed dynamical laws [12].

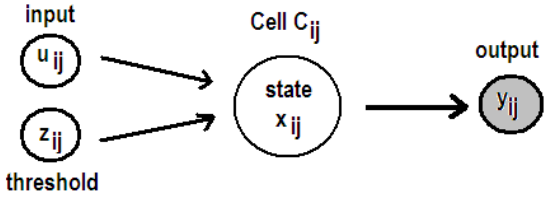


Figure 1.3: Isolated cell with the input, threshold, state and output.

Mathematically, each cell can be modeled by a nonlinear dynamical system where information is encoded via the initial state x_{ij}, the inputs u_{ij}, a threshold z_{ij} and the output values y_{ij} (Figure 1.3). The state of standard isolated cell C_{ij} is weighted by the sum of the inputs, the outputs and the threshold . The corresponding state equation is

$$\frac{dx_{ij}}{dt} = -x_{ij} + a_{ij} f(x_{ij}) + b_{ij} u_{ij} + z_{ij} \tag{1.1}$$

where a_{ij} and b_{ij} are weighting coefficients, and the output values are defined by the piecewise linear function (Figure 1.4).

$$y_{ij} = f(x_{ij}) = \frac{1}{2}\left(|x_{ij}(t)+1| - |x_{ij}(t)+1|\right) = \begin{cases} 1 & x_{ij} \geq 1 \\ x_{ij} & |x_{ij}| < 1 \\ -1 & x_{ij} \leq -1 \end{cases} \tag{1.2}$$

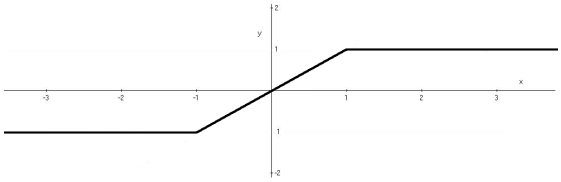


Figure 1.4: Piecewise linear function.

Each CNN cell is coupled locally only to those neighbor cells which lie inside a sphere of influence $S(i,j)_r$ (Figure 1.5).

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

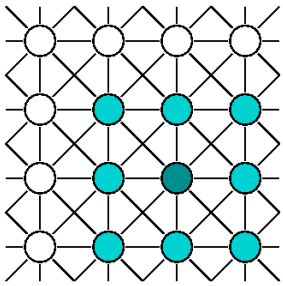


Figure 1.5: A CNN cell with a 3×3 sphere of influence.

The input u_{ij} of each neighbor cell injects a weighted contribution to the cell C_{ij} called B template

$$B(u_{ij}) = \sum_{kl \in S(i,j)_r, kl \neq ij} b_{kl} u_{kl}$$

The output contributes with A template in a similar way,

$$A(u_{ij}) = \sum_{kl \in S(i,j)_r, kl \neq ij} a_{kl} y_{kl}.$$

Templates (A, B, z) configure the CNN parameters and are called cloning template. The standard CNN equation is then

$$\frac{dx_{ij}}{dt} = -x_{ij} + A(y_{ij}) + B(u_{ij}) + z_{ij} = \tag{1.3}$$

$$= -x_{ij} + \sum_{kl \in S(i,j)_r} a_{kl} y_{kl} + \sum_{kl \in S(i,j)_r} b_{kl} u_{kl} + z_{ij}.$$

Observe that the state equation is not completely defined for boundary cells. Therefore, additional boundary conditions must be specified. The three most commonly chosen boundary conditions are: the Dirichlet boundary conditions, Newmann boundary conditions, and Toroidal boundary conditions. The former, takes x_{kl} at the boundary as fixed constant values. Newmann boundary conditions takes x_{kl} of neighboring cells perpendicular to the boundaries equal to each other. The latter, identifies first last rows (columns) of the array thereby forming a torus.

Cellular Neural, Nanoscale, Nonlinear Networks usually called CNNs, have been an interesting subject of study since their introduction back in 1988. The idea of CNN processors was introduced by Leon O. Chua and Lin Yang's in [5] and [6]. There, they outlined the mathematics behind CNN processors and studied the CNN system stability. They suggested some of the possible applications of CNNs like image processing and pattern recognition. A useful summary of definitions, CNN types, dynamics, implementations and applications of Cellular Neural Networks is provided in [4].

The first algorithmically analog CNN processor was introduced in [7]. This article proved that CNN could be produced and provided a physical platform to test the different theories on this CNN universal processor. Other universal properties like being a universal Turing machine have been also studied using the Game of Life algorithm. In [8] its shown that the Game of Life can be reproduced by 3-D Cellular Neural Network, thus the 3-D CNN is a universal machine in the Turing sense.

Another important feature of CNNs is that, in spite of their simplicity, its dynamics range from the convergence to single equilibrium points to chaotic behavior. The CNN stability has been widely studied in the literature. In their original paper [5], the authors studied the complete stability (convergence to a fixed-point) for symmetric templates. The first complete analysis of the global stability properties of a two neuron CNN was made in [25]. In [26] they extended the stability analysis providing simple criteria for checking different stability types: the complete one, the asymptotic one, and stability almost everywhere. Their results allow to check the stability via simple conditions on the template elements. Nevertheless, the complete

stability problem, looking for parameters conditions to make the system converge to a fixed-point, has been studied many times in the literature. For instance in [61], they found necessary and sufficient conditions for two-cell CNNs to be globally stable. Or in [62] where the stability for a two neuron CNN was studied under the assumptions that biases are set to zero and the self-coupling coefficients take the same value greater than one.

Many stability studies have been carried out for a particular parameter range like for opposite-sign templates or for non symmetric templates. For instance, in [13] where authors scrutinize the stability problem form a different point of view using the Gauss-Seidel method. We can find another example in [75], where it was found that the complete stability for opposite-sign templates is not always preserved, due to the existence of a limit cycle for certain parameter conditions.

Although a large number of papers concerning the global stability of CNNs have been published so far, a complete analysis of the stability process, non only those where the system converge to a fixed-point but also the convergence to different limit cycles is still missing.

Apart from stability, the CNN template design problem is another important theme where researchers have devoted their efforts to. In [10] they study the universe of stable CNN templates. In [9], a method based on a set of inequalities that must be satisfied by the CNN parameters allows them to design templates for simple applications. In [23], the stability criteria turn out to be useful in the template design problem. The "Art of CNN Template Design" in [74] describes useful templates to perform some of the common CNN uses for binary inputs and outputs. Sometimes, templates are classified into groups based on their properties. The template design for CNNs with 1-Bit weights studied in [42], would be an example.

Again, there are many other references taking on this challenge, but a simple and general algorithm to design templates has not been found yet. Usually to perform a given operation, template libraries are used. However, depending on the concrete problem to be solved, a single one may not be enough. In this case, different templates are applied in sequence. The template composition problem is another key point in the designing task that must be studied in a deeply way.

Another problem which arises from the stability studies is the existence of limit cycles. In [77], the two cell autonomous system is investigated finding a Hopf-like bifurcation at which the dynamic behavior changes from asymptotically stable to periodic. In [54], they study how and when a global propagation of information is possible through a 1-D CNN for certain connections between the neurons. For a particular case, a periodic solution is computed analytically and also the period function. The existence of limit cycles allows

a comparison between the dynamic behavior of different CNN models such as the original Chua-Yang model and the full-range model. It was proved in [24] that these models are not topologically equivalent via the existence of periodic solutions. Some applications using the existence of limit cycles for example to store individual patterns are shown in [17]. Nevertheless, a complete study of the existence of limit cycle and a classification of them depending on the CNN weights has not been done yet.

1.3 Outline

In this dissertation we will review the two neuron CNN stability in Chapter 2 using the Lyapunov theory and using the different local dynamic behavior derived from the use of the piecewise linear function which gives us a geometrical way to understand the system dynamics. The Lyapunov stability establishes the fundamentals to adress the different convergence problems that can be studied when the CNN system converges to a fixed-point. The geometric stability, shed light on the convergence to limit cycles. This work is basically organized based on these two convergence classes.

The different system stability, the template design problem, the dependence on initial conditions or the composition of the different templates needed in order to make possible a concrete association between an input and an output are some of the problems which has to be dealt in order to understand completely the CNN system. So we try to make an exhaustive study about Cellular Neural Networks in order to find the intrinsic difficulties and the possible uses of a CNN. Understanding the CNN system in a lower dimension can give some of the main keys in order to understand the general case. That's why we will focus our study in the one dimensional CNN case with only two neurons. Furthermore, this simple CNN case has the invaluable advantage of allowing the possibility to understand the existence and kind of limit cycles.

From the results obtained using the Lyapunov function, we propose in Chapter 3 some methods to avoid the problem of the dependence on initial conditions. Its intrinsic characteristics as a quadratic form of the output values, gives us the key points to find a parameters where the final outputs do not depend on initial conditions.

At this point we are able to study different CNN applications for the parameter range where the system converges to a fixed-point. Each section in Chapter 4 describes these different problems. We start by using CNNs to reproduce Bernoulli probability distributions based on the geometry of the Lyapunov function. Secondly, we reproduce linear functions while working

inside the unit square. The template design problem is studied in the third section. The existence of the Lyapunov function allows to construct a map, which we call convergence map, depending on the parameters of the CNN template. It relates the CNN inputs with the final outputs. This map gives us a recipe to design templates performing some desired input-output associations. The results obtained drive us into the template composition problem. We will study the way different templates can be applied in sequence.

From the results obtained in the template design problem, we may think on finding a functional relation between the external inputs and the final outputs. Due to the piecewise linear function, the set of final states is discrete. The input-output correspondence can be thought then as a classification problem. Each one of the different classes is defined by the different final states which, will depend on the parameters of the CNN system. Fourth section study which classifications problems can be solved by a two neuron CNN, and relate them with the weight parameters. In this case, we also find a recipe to design templates performing these classification problems. The results obtained drive us into the problem to realize Boolean functions using CNNs in section five, and show us some of the CNN limits, trying to reproduce the header of a universal Turing machine.

Chapters 5 and 6, study the existence and kind of limit cycles. Based on a particular limit cycle example extracted from Chua's book [12], we start this study with antisymmetric connections between cells. The results obtained can be generalized for CNNs with opposite sign parameters. From the stability study, limit cycles has the possibility to exist for this parameter range. The periodic behavior of these curves, is computed in a particular case in Chapter 7. The limit cycle period can be expressed as a function of the CNN parameters, and can be used to generate clock signals. Finally, conclusions and ideas for the future work are sketched.

Appendix A deals with the intrinsic difficulties of the convergence map construction. In Appendix B, we list the different input-output relations that can be obtained while solving classification problems and resume the correspondent templates. The problem to reproduce the header action of a universal Turing machine problem is treated in Appendix C. At last, different limit cycles cases for an antisymmetric template and for a general template are summarized in the final Appendices D and E.

Part I

Convergence

Chapter 2

CNN stability

Many living brains and intelligent machines are made of elementary units which exhibit two stable states. CNNs wanted initially to mimic brain functions [45] and this is one of the reasons why this bistable character must be studied. Moreover, this is the main reason why the output of each cell of the CNN is defined by the piecewise linear function. This bi-stability is achieved if all trajectories converge to an equilibrium point where the magnitude of each state is bigger than one. The stability of a dynamical system is one of the main tools of the differential equations theory. In the CNN case, it is necessary to study it mainly because CNNs were initially designed to perform image processing [6]. Specifically, the original application of CNN processors was to perform real-time ultra-high frame-rate processing unachievable by digital processors. For some concrete CNN implementation, the CNN system needs to have a one to one correspondence between an initial state and a final output. Usually, the initial state of a CNN for image processing problems is taken as the external one (u_0, u_1) with constant values. The final output is taken as (y_0, y_1) at the end of the process.

In this section, we will see mainly that the states of each neuron are always bounded, the system converge to a fixed-point for some parameter range, and the final output can take values $+1, -1$ under certain parameter conditions. There are basically two methods by which to study the system stability. One way is using the Lyapunov theory. The existence of a Lyapunov a function implies that the system is not chaotic, and it is a basic tool in order to demonstrate that the system will converge to a fixed-point as final output.

Another way to study the system stability is the geometric one. Thanks to the piecewise linear function, the input space $\{(u_0, u_1), u_i \in \mathbb{R}\}$ can be divided into nine different regions where the CNN system locally behaves like a linear one. Studying the CNN system in each of these nine regions, we find the local equilibrium points, and the trajectories in the phase plane.

These information let us assure that the system will converge to a fixed-point as final output. Still, there exist the possibility to obtain periodic solutions for certain parameter range.

2.1 Two neuron CNN

Along this dissertation, our notation for the two neuron CNN piecewise linear system will be

$$\begin{cases} \dot{x_0} = -x_0 + sy_0 + p_+y_1 + b_0u_0 + b_+u_1 + I \\ \dot{x_1} = -x_1 + sy_1 + p_-y_0 + b_-u_0 + b_0u_1 + I \end{cases} \tag{2.1}$$

where variables x_i are the internal states of the neuron. They characterize intrinsically the neuron and are usually taken in $[-1, 1]$. The external states y_i are defined by the piecewise linear function (Figure 1.4), in the simple CNN model.

$$y_i = f(x_i) = \frac{1}{2}\left(|x_i(t) + 1| - |x_i(t) - 1|\right), \quad i = 0, 1. \tag{2.2}$$

External inputs are u_i and they shall be constant in time. The other parameters configure the network cloning template (A, B, z) where

$$A = (p_-, s, p_+), \quad B = (b_-, b_0, b_+), \quad z = I.$$

In order to simplify this analysis, the external influence on the CNN system is described using the notation $u' = Bu + I$

$$\begin{cases} u'_0 = b_0u_0 + b_+u_1 + I \\ u'_1 = b_-u_0 + b_0u_1 + I \end{cases} \tag{2.3}$$

We define the action on (u_0, u_1) to obtain (u'_0, u'_1) as B-transformation.

2.2 Bounded states

To tackle the system stability problem, we first study the state-boundedness criterion like in [5].

Theorem 1. All states x_i for $i = 0, 1$ in a Cellular Neural Network are bounded for all time $t > 0$ and the bound x_{max} is :

$$x_{max} = 1 + |s| + \max\{|p_+|, |p_-|\} + \sum_{j=0}^{1} |b_{ij}| + |I| \tag{2.4}$$

Proof. First, let us rewrite the state equation (2.1) as

$$\dot{x_i}(t) = -x_i(t) + f_i(t) + g_i(u) + I \quad i = 0, 1 \tag{2.5}$$

where

$$f_0(t) = sy_0 + p_+y_1, \qquad f_1(t) = sy_1 + p_-y_0,$$
$$g_0(u) = b_{00}u_0 + b_{01}u_1, \qquad g_1(u) = b_{11}u_1 + b_{10}u_0.$$

Solving the equation (2.5) as a linear first-order ordinary differential equation we obtain $x_i(t)$.

$$\dot{x_i}(t) = -x_i \Longrightarrow \ln x_i = -t + C \Longrightarrow x_i(t) = Ke^{-t}$$

$$\dot{x_i}(t) = K'(t)e^{-t} - K(t)e^{-t} = -K(t)e^{-t} + f_i(t) + g_i(u) + I \Longrightarrow$$

$$\Longrightarrow K(t) = \underbrace{\int e^t \left(f_i(t) + g_i(u) + I\right) dt}_{I(t)} + C \Longrightarrow$$

$$\Longrightarrow x_i(t) = Ce^{-t} + e^{-t}I(t)$$

$$x_i(0) = C + I(0) \Longrightarrow C = x_i(0) - I(0) \Longrightarrow x_i(t) = x_i(0)e^{-t} + e^{-t}(I(t) - I(0))$$

This solution can be rewritten as:

$$x_i(t) = x_i(0)e^{-t} + \int_0^t e^{-(t-\tau)}[f_i(\tau) + g(u) + I]d\tau$$

At this point, we are able to bound $x_i(t)$.

$$|x_i(t)| \leq |x_i(0)e^{-t}| + \left|\int_0^t e^{-(t-\tau)}[f_i(\tau) + g(u) + I]\right| \leq$$

$$\leq |x_i(0)|e^{-t} + \int_0^t e^{-(t-\tau)}[|f_i(\tau)| + |g(u)| + |I|]d\tau \leq$$

$$\leq |x_i(0)|e^{-t} + [F_i + G_i + |I|] \int_0^t e^{-(t-\tau)}d\tau =$$

$$\leq |x_i(0)|e^{-t} + [F_i + G_i + |I|](1 - e^{-t}) \leq |x_i(0)| + [F_i + G_i + |I|]$$

where

$$F_i = \max_t |f_i(t)| \leq |s| + \max\{|p_+|, |p_-|\}$$

because $|y_i(t)| \leq 1$ for all t, because its defined by the piecewise linear function, and

$$G_i(u) = \max_u |g_i(u)|$$

Since $|x_i(0)|$ and $|u_i|$ satisfy conditions $|x_i(t)| \leq 1$, $\forall t$, $|u_i| \leq 1$, it follows that

$$\max_t |x_i(t)| \leq 1 + |s| + \max\{|p_+| + |p_-|\} + \sum_{j=0}^{1} |b_{ij}| + |I|$$

□

2.3 Lyapunov stability

The Lyapunov theory is based on the basic notion that some measure of energy dissipation of a system of differential equations, allows us to determine the system stability. If this energy dissipation can be measured, then it is not necessary to explicitly integrate the system equations. To scrutinize the system stability in such a way, a Lyapunov function should be found. In the two neuron CNN case, a Lyapunov function can be defined if the feedback coefficients $p_+ = p_- = p$ are symmetric.

Definition 1. *Let us define a function L(t),*

$$L(t) = -\frac{1}{2}(sy_0^2 + py_0y_1 + py_1y_0 + sy_1^2) + \frac{1}{2}(y_0^2 + y_1^2) - u_0'y_0 - u_1'y_1 =$$

$$= -py_0y_1 - \frac{S}{2}(y_0^2 + y_1^2) - u_0'y_0 - u_1'y_1 \tag{2.6}$$

where S=s-1 and $u_i' = b_{ii}u_i + b_{ij}u_j + I$ *for i=0,1.*

Based on some general stability results obtained in [5], next theorems prove that L(t) fulfills the necessary conditions to be a Lyapunov function.

Theorem 2. Function $L(t)$ defined in (2.6) is bounded from below.

Proof.

$$|L(t)| = |-py_0y_1 - \frac{S}{2}(y_0^2 + y_1^2) - u_0'y_0 - u_1'y_1| \leq$$

$$\leq |p||y_0||y_1| + \frac{S}{2}(|y_0|^2 + |y_1|^2) +$$

$$+(|b_{00}||u_0| + |b_{01}||u_1| + |I|)|y_0| + (|b_{10}||u_0| + |b_{11}||u_1| + |I|)|y_1| \leq$$

$$\leq |p| + \frac{|S|}{2}(1+1) + \sum_{i,j=0}^{1} |b_{ij}| + 2|I| = |p| + |S| + \sum_{i,j=0}^{1} |b_{ij}| + 2|I|$$

because $|u_i| \leq 1$,$|y_i| \leq 1$ for $i = 0, 1$. □

Theorem 3. Function L(t) defined in (2.6) is a monotone-decreasing function, that is

$$\dot{L(t)} \leq 0$$

Proof. In order to study the Lyapunov function derivative we first must be careful about the outputs. They are defined by the piecewise linear function, which it is not differentiable at the break points (1.2). To solve this problem, we define the derivative at the break points $x_i = \pm 1$ as 0.

$$\frac{dy_i}{dx_i} = \begin{cases} 1 & |x_i| < 1 \\ 0 & |x_i| \geq 1 \end{cases}$$

Then, we are able to derive $L(t)$ along the trajectories (Lie derivative) in order to see that $\dot{L(t)} \leq 0$.

$$\dot{L(t)} = -p(\frac{dy_0}{dx_0}\dot{x_0}y_1 + \frac{dy_1}{dx_1}\dot{x_1}y_0) - S\sum_{i=0}^{1}\frac{dy_i}{dx_i}\dot{x_i}y_i - \sum_{i=0}^{1}u_i'\frac{dy_i}{dx_i}\dot{x_i} =$$

$$= \begin{cases} 0 & |x_i| \geq 1, i = 0, 1 \\ -p(\dot{x_1}y_0) - S\dot{x_1}y_1 - u_1'\dot{x_1} & |x_0| \geq 1, |x_1| < 1 \\ -p(\dot{x_0}y_1) - S\dot{x_0}y_0 - u_0'\dot{x_0} & |x_0| < 1, |x_1| \geq 1 \\ -p(\dot{x_0}y_1 + \dot{x_1}y_0) - S\sum_{i=0}^{1}\dot{x_i}x_i - \sum_{i=0}^{1}u_i'\dot{x_i} & |x_i| < 1, i = 0, 1 \end{cases} =$$

$$= \begin{cases} 0 & |x_i| \geq 1, i = 0, 1 \\ -\dot{x_1}\left(py_0 + Sx_1 + u_1'\right) & |x_0| \geq 1, |x_1| < 1 \\ -\dot{x_0}\left(py_1 + Sx_0 + u_0'\right) & |x_0| < 1, |x_1| \geq 1 \\ -\dot{x_0}(py_1 + Sx_0 + u_0') - \dot{x_1}(py_0 + Sx_1 + u_1') & |x_i| < 1, i = 0, 1 \end{cases} =$$

Substituting the cell circuit equation (2.1) in our expression we obtain,

$$\dot{L(t)} = \begin{cases} 0 & |x_i| \geq 1, i = 0, 1 \\ -\left(\dot{x_1}\right)^2 & |x_0| \geq 1, |x_1| < 1 \\ -\left(\dot{x_0}\right)^2 & |x_0| < 1, |x_1| \geq 1 \\ -\sum_{i=0}^{1}(\dot{x_i})^2 & |x_i| < 1, i = 0, 1 \end{cases} \leq 0$$

□

Theorem 4. For any given input u_i i=0,1 and any initial state x_i i=0,1 of a Cellular Neural Network, we have

$$\lim_{t\to\infty} L(t) = constant \quad \text{and} \quad \lim_{t\to\infty} \frac{dL(t)}{dt} = 0$$

Proof. From Theorems (2) and (3), $L(t)$ is a bounded monotone decreasing function of time t. Hence L(t) converges to a constant limit and its derivative converges to 0.

$$\lim_{t\to\infty} L(t) = k \in \mathbb{R},$$

$$\lim_{t\to\infty} \frac{dL(t)}{dt} = \lim_{t\to\infty} \frac{dk}{dt} = 0.$$

□

Therefore, $L(t)$ is a Lyapunov function and so the system is stable in the sense of Lyapunov. Moreover, this study let us know the dynamic behavior of the different states at the end of the process.

$$\lim_{t\to\infty} \frac{dL}{dt} = -\lim_{t\to\infty} \sum_{i=0}^{1} \left(\frac{dx_i(t)}{dt}\right)^2 = 0$$

Corollary 1. In the limit as t tends to ∞, the CNN system output fulfills

$$\lim_{t\to\infty} y_i(t) = k \in \mathbb{R} \;\; i = 0,1 \quad \text{or} \quad \lim_{t\to\infty} \frac{dy_i(t)}{dt} = 0 \;\; i = 0,1$$

At this point, we are able to study the steady-state behavior of the CNN as t tends to infinity and see where the CNN system can converge. Three different cases describe the state of a cell.

Case 0. If the state of a cell x_i in the limit is $|x_i| < 1$, then $y_i = x_i$ and $\lim_{t\to\infty} \frac{dL}{dt} = -\lim_{t\to\infty} \sum_{i=0}^{1} (\dot{x_i})^2 = 0$. Hence, $\lim_{t\to\infty} \dot{x_i} = \lim_{t\to\infty} \dot{y_i} = 0$. In this case, the system converges to a fixed-point inside the unit square.

Case 1. If the state of a cell in the limit is $|x_i| \geq 1$ and $\lim_{t\to\infty} \frac{dL}{dt} = -\lim_{t\to\infty} (\dot{x_i})^2 = 0$, then $\lim_{t\to\infty} \dot{x_i} = 0$. In this case, the steady-state is out from the unit square. The final output is then $y_i = \pm 1$ because of the piecewise linear function.

Case 2. If the state of a cell in the limit is $|x_i| \geq 1$ and $\lim_{t\to\infty} \dot{x_i} \neq 0$, then x_i may be a periodic or aperiodic but bounded function in view of Theorem 2.4.

This three different cases: converge to an isolated point inside the unit square (Case 0), out from the unit square (Case 1) or converge to a bounded function (Case 2), can not co-exist for certain parameter range when the Cellular Neural Network is in its steady state. Only two of them can exist at the same time.

Next theorem will give us one of the main results on the convergence study. For certain parameter conditions, the system will converge to fixed-points where the final output y_i will take binary values ± 1. Two neuron CNN will then be completely stable in the sense that every trajectory converges to an attractive equilibrium point.

Definition 2. *A dynamical system $\dot{x} = f(x)$ is said to be completely stable if for each initial condition $\widetilde{x_0} \in \mathbb{R}^n$,*

$$\lim_{t\to\infty} x(t, \widetilde{x_0}) = const$$

where $x(t, \widetilde{x_0})$ is a trajectory starting from $\widetilde{x_0}$.

Theorem 5. Let us consider a completely stable CNN. If parameter s of a symmetric template satisfies $s > 1$, the output of every cell y_i is $+1$ or -1 for $i = 0, 1$ (Case 1).

Proof. Let us rewrite the state equation (2.1) as

$$\begin{cases} \dot{x_0} = -f(x_0) + g_0(t) \\ \dot{x_1} = -f(x_1) + g_1(t) \end{cases} \qquad \text{where} \qquad \begin{cases} f(x_0) = x_0 - sy_0 \\ g_0(t) = py_1 + u_0' \\ f(x_1) = x_1 - sy_1 \\ g_1(t) = py_0 + u_1' \end{cases}$$

where $g(x_i)$ describes the influence of the neighboring cells and the external inputs. Since the CNN is completely stable, $x_i(t)$ converges to a constant state. At this final state, $g_i(t)$ is constant because it depends only on the external inputs and y_i, which are constants, and can be seen as a translation of the function $f(x_i)$. The curve representing the dynamics of the CNN, is a three-segment piecewise-linear curve as can be seen in Figure 2.1 for $s > 1$.

$$\dot{x_i} = -x_i + sy_i + g_i(t) = \begin{cases} -x_i - x + g_i(t) & x_i < -1 \\ (-1+s)x_i + g_i(t) & |x_i| \leq 1 \\ -x_i + g_i(t) & x_i > 1 \end{cases}$$

Let us note that there can be one, two or three points where the system can converge depending on $g_i(t)$ sign. To study if these points are attractive or repulsive, we study the sign of $\dot{x_i}$ in each case. First let us call x_Q, x_M and x_P the points where $\dot{x_i} = 0$ for $x_i < -1$, $|x_i| \leq 1$ and $x_i > 1$ respectively (Figure 2.1).

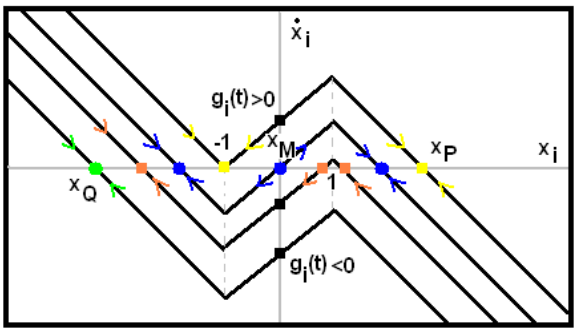


Figure 2.1: Dynamic routes for different $g_i(t)$ values.

- For $x_i < -1$, $\dot{x}_i = -x_i + x_Q$. If $x_i > x_Q$, then $\dot{x}_i < 0$. Otherwise, $\dot{x}_i > 0$. So $x_Q < -1$ is an attractive point and the system can converge to it. The final output is in this case $y_i = -1$.
- For $|x_i| \leq 1$, $\dot{x}_i = (s-1)x_i + (s-1)x_M$. If $x_i > x_M$, then $\dot{x}_i > 0$. Otherwise, $\dot{x}_i < 0$. So $|x_M| < 1$ is a repulsive point and the system can not converge to it.
- For $x_i > 1$, $\dot{x}_i = -x_i + x_P$. If $x_i > x_P$, then $\dot{x}_i < 0$. Otherwise, $\dot{x}_i > 0$. So $x_P > 1$ is an attractive point and the system can converge to it. The final output is in this case $y_i = +1$.
- For $x_Q = -1$ or $x_P = +1$, both points are attractive and so the system can converge to them with final output values ± 1.

Hence, the attractive points where the system can converge share the common property to be out from region $|x_i| \leq 1$. It means that the output y_i of every cell is $+1$ or -1 if parameter $s > 1$. □

2.4 Geometric stability

Another way to study the system stability is studying the system trajectories, its equilibrium points type and position in each of the nine regions where the CNN system is linear. The geometric representation of the trajectories in the phase plane, is an invaluable tool in studying dynamical systems. This reveals information such as whether a stable point, a repulsive point, or a limit cycle is present in the dynamic behavior for some chosen parameter values.

In general, a linear system $\dot{x} = F(x)$, $x \in \mathbb{R}^n$ can be written in matrix form as $\dot{x} = Ax$, $A \in \mathbb{R}^{n \times n}$. Points x^* where $F(x^*) = 0$, named equilibrium

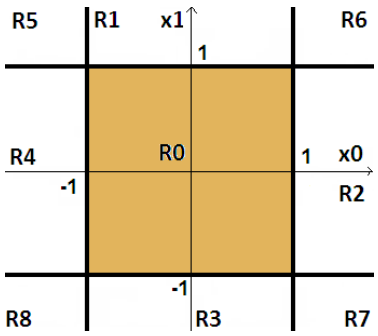


Figure 2.2: Regions on the plane where the CNN system is linear.

points, and the eigenvalues and eigenvectors of matrix A, let us print the system trajectories in the phase plane. This analysis help us to understand the relation between the parameters of the cloning template and the final trajectory solutions.

We divide this geometric study into two different sections. The symmetric one, where we should obtain similar results as those obtained from the Lyapunov study and the general one.

2.4.1 Symmetric parameter range

For a symmetric parameter range $p_+ = p_- = p$, we will study the system equilibrium points in order to see that the final outputs of the CNN system are $+1$ or -1 if parameter s is bigger than 1.

Let R_i, $i \in \{0, \ldots, 8\}$ be the nine regions of the plane limited by lines $x_i = \pm 1$ for $i = 0, 1$ where the CNN system is linear (Figure 2.2). The central region is called R_0, the middle regions are called R_1, R_2, R_3 and R_4 and the out regions R_5, R_6, R_7 and R_8. Local equilibrium points of each region are (m_i, k_i), $i = 0, \ldots, 8$.

Cellular Neural Network general equations can not be solved at once. Due to the piecewise linear function $y_i = pwl(x_i)$, trajectory solutions must be found solving the differential equations system along the process in each region R_i.

$$\begin{cases} \dot{x_0} = -x_0 + sy_0 + py_1 + u'_0 \\ \dot{x_1} = -x_0 + sy_1 + py_0 + u'_1 \end{cases} \tag{2.7}$$

Let A_i be the CNN matrix of the system corresponding to each one of the nine regions R_i for $i = 0, 1, \ldots, 8$ in the symmetric case. The matrix form for the CNN system equations (2.7) can then be written as $\dot{x_k} = A_i x_k + u'_k$,

x_i^*	m_i	k_i
R_0	0	0
R_1	$-\frac{p}{s-1}$	$s-\frac{p^2}{s-1}$
R_2	$s-\frac{p^2}{s-1}$	$-\frac{p}{s-1}$
R_3	$\frac{p}{s-1}$	$-s+\frac{p^2}{s-1}$
R_4	$-s+\frac{p^2}{s-1}$	$\frac{p}{s-1}$
R_5	$p-s$	$-p+s$
R_6	$p+s$	$p+s$
R_7	$-p+s$	$p-s$
R_8	$-p-s$	$-p-s$

Table 2.1: Symmetric parameter range. Equilibrium points (m_i, k_i) in each of the nine regions R_i where the system is linear in the autonomous case where $u'_j = 0$, $j = 0, 1$.

$k = 0, 1$, where $u'_k = Bu_k + I$ corresponds to the B-transformation (2.3). Local equilibrium points in each region can be seen in Table 2.1 for $u'_i = 0$.

In region R_0, the CNN matrix A_0 of the system is:

$$A_0 = \begin{pmatrix} s-1 & p \\ p & s-1 \end{pmatrix} \tag{2.8}$$

In the central and out regions, CNN matrices are:

$$\begin{aligned} &A_1 = A_3 = \begin{pmatrix} s-1 & 0 \\ p & -1 \end{pmatrix}, A_2 = A_4 = \begin{pmatrix} -1 & p \\ 0 & s-1 \end{pmatrix}, \\ &A_5 = A_6 = A_7 = A_8 = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}. \end{aligned} \tag{2.9}$$

Central region R_0

Working inside R_0, where $y_i = x_i$, CNN system equations can be written as:

$$\begin{pmatrix} \dot{x_0} \\ \dot{x_1} \end{pmatrix} = \begin{pmatrix} s-1 & p \\ p & s-1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \end{pmatrix} + \begin{pmatrix} u'_0 \\ u'_1 \end{pmatrix}$$

To linearize the system, we do the variable change,

$$x_0 = X_0 + m_0, \quad x_1 = X_1 + k_0 \tag{2.10}$$

where $m_0 = \frac{(1-s)u_0' - pu_1'}{p^2-(1-s)^2}$ and $k_0 = \frac{(1-s)u_1' - pu_0'}{p^2-(1-s)^2}$. Now, in order to find the diagonal form, we must find the eigenvalues and the eigenvectors.

$$p(\lambda) = \det \begin{pmatrix} s-1-\lambda & p \\ p & s-1-\lambda \end{pmatrix} = 0$$

Eigenvalues are $\lambda_0 = (s-1)+p$, $\lambda_1 = (s-1)-p$ and the associated eigenvectors are $(\frac{1}{\sqrt{2}}, \pm\frac{1}{\sqrt{2}})$. The linear system solutions are then,

$$\begin{pmatrix} X_0 \\ X_1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} C_0 e^{(s-1+p)t} \\ C_1 e^{(s-1-p)t} \end{pmatrix}$$

where C_0 and C_1 are arbitrary constants, and their Cartesian equations are

$$X_0 + X_1 = 2C_0(e^t)^{s-1+p}, X_0 - X_1 = 2C_1(e^t)^{s-1-p} \Rightarrow$$

$$(X_0 + X_1)^{s-1-p} = K(X_0 - X_1)^{s-1+p}$$

At this point, we are able to discuss the equilibrium point kind depending on parameters s and p. Let us suppose $p \geq 0$. The equilibrium point in this region is:

$$(m_0, k_0) = \left(\frac{(s-1)u_0' - pu_1'}{p^2-(s-1)^2}, \frac{(s-1)u_1' - pu_0'}{p^2-(s-1)^2} \right)$$

1. If $\lambda_0 \neq \lambda_1 \in \mathbb{R}$, $\Rightarrow p \neq 0$.
   (a) If $s-1 < -p$, equilibrium point is an attractive point called sink.
   (b) If $-p < s-1 < p$, CNN solutions explode whether when $t \to \pm\infty$ except along the straight line solution associated to each eigenvalue. In this case, equilibrium point is a saddle.
   (c) If $s-1 > p$, CNN solutions explode tangent to the straight line solution associated to the biggest eigenvalue. Equilibrium point must be a source.
   (d) If $s-1 = \pm p$, the system has zero as an eigenvalue and the other eigenvalue is $\lambda = 2(s-1)$. In this case, there exists a line of equilibrium points. If $s < 1$, the straight line solutions tend to the line of equilibrium points (parallel to he eigenvector associated to the eigenvalue λ), and if $s > 1$ the straight line solutions get away from the line of equilibrium points.

2. If $\lambda_0 = \lambda_1 \in \mathbb{R} \Rightarrow p = 0$, the CNN matrix is diagonal. For $s < 1$, solutions (except for the equilibrium point) are straight lines approaching the equilibrium point. For $s > 1$, solutions goes to infinity straight in every direction.

From this study we can conclude that trajectories will leave the unit square if parameter s is bigger than one because in these cases, there is a repulsive local equilibrium point. Now we are going to do a similar study of the dynamic behavior in the other regions named middle and out regions.

Middle regions: R_1, R_2, R_3, R_4

Studying the local equilibrium points in each of these regions, we will find the parametric system solutions, and the different possible equilibrium points kinds depending on the CNN parameters.

- Region R_1. The state equation in region $R_1 = \{(x_0, x_1) \in \mathbb{R} | x_0 \in [-1, 1], x_1 > 1\}$ is:

$$\begin{cases} \dot{x_0} = -x_0 + sx_0 + p + u_0' \\ \dot{x_1} = -x_1 + s + px_0 + u_1' \end{cases}$$

The local equilibrium point is:

$$(m_1, k_1) = \left(-\frac{p + u_0'}{s - 1}, s + u_1' - \frac{p}{s - 1}(p + u_0') \right)$$

Applying the linear change $x_0 = X_0 + m_1, x_1 = X_1 + k_1$ at $\dot{\vec{x}} = A_1\vec{x}$, we obtain a diagonal system with eigenvalues $\lambda_0 = s - 1, \lambda_1 = -1$ and associated eigenvectors $S(\lambda_0) = \ker(A_1 - \lambda_0 Id) =< (s, p) >$, and $S(\lambda_1) = \ker(A_1 - \lambda_1 Id) =< (0, 1) >$.

Parametric solutions are:

$$\begin{cases} x_0(t) = \alpha_0 s e^{(s-1)t} + m_1 \\ x_1(t) = \alpha_0 e^{(s-1)t} + \alpha_1 e^{-t} + k_1 \end{cases} \tag{2.11}$$

Cartesian solutions are:

$$(x_0 - m_1)(s(x_1 - k_1) - p(x_0 - m_1))^{s-1} = C,$$

where $C = s\alpha_0(s\alpha_1)^{s-1}$.

Local equilibrium point (m_1, k_1) can be attractive or repulsive depending on parameter s as can be seen in Table 2.2.

Attractor node	$s < 0$
Improper attractor node	$s = 0$
Attractor node	$0 < s < 1$
Saddle point if $\lambda_1 < 0 < \lambda_0$	$s > 1$

Table 2.2: Equilibrium point kind (m_1, k_1) of region R_1. The special case where $s = 1$ do not have a single equilibrium point but a line of them.

For $s = 1$, the system has zero as an eigenvalue and there exists a line of equilibrium points. The other eigenvalue is -1 and so the straight line solutions tend to the line of equilibrium points parallel to the eigenvector associated to the eigenvalue -1.

- Region R_2. The state equation in $R_2 = \{(x_0, x_1) \in \mathbb{R} | x_0 > 1, x_1 \in [-1, 1]\}$ is:

$$\begin{cases} \dot{x_0} = -x_0 + s + px_1 + u_0' \\ \dot{x_1} = -x_1 + sx_1 + p + u_1' \end{cases}$$

Local equilibrium point is:

$$(m_2, k_2) = \left(s + u_0' - \frac{p}{s-1}(p + u_1'), -\frac{p + u_1'}{s-1} \right)$$

Doing a linear change $x_0 = X_0 + m_2, x_1 = X_1 + k_2$, we obtain a diagonal system with eigenvalues $\lambda_0 = -1, \lambda_1 = s - 1$ just like in Region R_1. The associated eigenvectors are: $S(\lambda_0) = \ker(A_2 - \lambda_0 Id) =< (1, 0) >$ and $S(\lambda_1) = \ker(A_2 - \lambda_1 Id) =< (p, s) >$.

Parametric solution is:

$$\begin{cases} x_0(t) = \alpha_0 e^{-t} + \alpha_1 p e^{(s-1)t} + m_2 \\ x_1(t) = \alpha_1 s e^{(s-1)t} + k_2 \end{cases}$$

Cartesian solution is:

$$(x_1 - k_2)(s(x_0 - m_2) - p(x_1 - k_2))^{s-1} = C,$$

where $C = s\alpha_1(s\alpha_0)^{s-1}$. Local equilibrium point (m_2, k_2) can be an attractor node, a saddle point or an improper attractive point just like the equilibrium point of region R_1 (Table 2.2).

- Region R_3. The state equation in $R_3 = \{(x_0, x_1) \in \mathbb{R} | x_0 \in [-1, 1], x_1 < -1\}$ is:

$$\begin{cases} \dot{x_0} = -x_0 + sx_0 - p + u_0' \\ \dot{x_1} = -x_1 - s + px_0 + u_1' \end{cases}$$

Equilibrium point is:

$$(m_3, k_3) = \left(\frac{p - u_0'}{s-1}, u_1' - s + \frac{p}{s-1}(p - u_0') \right)$$

Doing a linear change $x_0 = X + m_3, x_1 = X_1 + k_3$ we obtain a diagonal system with eigenvalues $\lambda_0 = s - 1, \lambda_1 = -1$ with associated eigenvectors like in in Region R_1: $S(\lambda_0) = (s, p)$ and $S(\lambda_1) = (0, 1)$. Parametric solution is:

$$\begin{cases} x_0(t) = \alpha_0 s e^{(s-1)t} + m_3 \\ x_1(t) = \alpha_0 p e^{(s-1)t} + \alpha_1 e^{-t} + k_3 \end{cases}$$

Cartesian solution is:

$$(x_0 - m_3)(s(x_1 - k_3) - p(x_0 - m_3))^{s-1} = C,$$

where $C = s\alpha_0(s\alpha_1)^{s-1}$. And the local equilibrium point (m_3, k_3), as in Region 1, can be an attractor node, a saddle point or an improper attractor node (Table 2.2).

- Region R_4. The state equation in $R_4 = \{(x_0, x_1) \in \mathbb{R} | x_0 < -1, x_1 \in [-1, 1]\}$ is:

$$\begin{cases} \dot{x_0} = -x_0 - s + px_1 + u_0' \\ \dot{x_1} = -x_1 + sx_1 - p + u_1' \end{cases}$$

Equilibrium point is:

$$(m_4, k_4) = \left(u_0' - s + \frac{p}{s-1}(p - u_1'), \frac{p - u_1'}{s-1} \right)$$

The system eigenvalues and eigenvectors are the same as in Region R_2: $\lambda_0 = -1$, $\lambda_1 = s - 1$, $(1, 0)$ and $= (p, s)$.

Parametric solution is:

$$\begin{cases} x_0(t) = \alpha_0 e^{-t} + \alpha_1 p e^{(s-1)t} + m_4 \\ x_1(t) = \alpha_1 s e^{(s-1)t} + k_4 \end{cases}$$

Equilibrium point (m_4, k_4), as in Region R_2, can be an attractor node, a saddle point or an improper attractor node (Table 2.2) .

Cartesian solution is:

$$(x_1 - k_4)(s(x_0 - m_4) - p(x_1 - k_4))^{s-1} = C,$$

where $C = \alpha_1 s(s\alpha_0)^{s-1}$.

Out regions: R_5, R_6, R_7, R_8

We will study the equilibrium points kind of the CNN system in each one of the out regions. From the eigenvalues and eigenvectors obtained in each case, we will deduce the local system solution.

- Region R_5. The state equation in $R_5 = \{(x_0, x_1) \in \mathbb{R} | x_0 < -1, x_1 > 1\}$ is:

$$\begin{cases} \dot{x_0} = -x_0 - s + p + u_0' \\ \dot{x_1} = -x_1 + s - p + u_1' \end{cases} \tag{2.12}$$

  Local equilibrium point is:

$$(m_5, k_5) = (u_0' + p - s, u_1' - p + s)$$

  Eigenvalues are : $\lambda_0 = \lambda_1 = -1 < 0$ with associated eigenvectors $S(\lambda_0) = (1, 0)$ and $S(\lambda_1) = (0, 1)$. The parametric solution is:

$$\begin{cases} x_0(t) = \alpha_0 e^{-t} + m_5 \\ x_1(t) = \alpha_1 e^{-t} + k_5 \end{cases}$$

  and the Cartesian one is: $(x_0 - m_5) = C(x_1 - k_5)$ where $C = \alpha_0/\alpha_1$. The local equilibrium point (m_5, k_5) is an attractor focus, because A diagonalize and do not depend on parameter s.

- Region R_6. The state equation in $R_6 = \{(x_0, x_1) \in \mathbb{R} | x_0 > 1, x_1 > 1\}$ is:

$$\begin{cases} \dot{x_0} = -x_0 + s + p + u_0' \\ \dot{x_1} = -x_1 + s + p + u_1' \end{cases}$$

  Local equilibrium point is:

$$(m_6, k_6) = (u_0' + p + s, u_1' + p + s)$$

  Eigenvalues and eigenvectors are equal to Region R_5. The parametric solution is:

$$\begin{cases} x_0(t) = \alpha_0 e^{-t} + m_6 \\ x_1(t) = \alpha_1 e^{-t} + k_6 \end{cases}$$

  And Cartesian solution is: $(x_0 - m_6) = C(x_1 - k_6)$. Equilibrium point (m_6, k_6) is an attractor focus.

- Region R_7. The state equation in $R_7 = \{(x_0, x_1) \in \mathbb{R} | x_0 > 1, x_1 < -1\}$ is:

$$\begin{cases} \dot{x_0} = -x_0 + s - p + u_0' \\ \dot{x_1} = -x_1 - s + p + u_1' \end{cases}$$

  Local equilibrium point is:

$$(m_7, k_7) = (u_0' - p + s, u_1' + p - s)$$

  Eigenvalues and eigenvectors are equal to Region R_5. The system solution is:

$$\begin{cases} x_0(t) = \alpha_0 e^{-t} + m_7 \\ x_1(t) = \alpha_1 e^{-t} + k_7 \end{cases}$$

  And Cartesian solution is: $(x_0 - m_7) = C(x_1 - k_7)$. Equilibrium point (m_7, k_7) is again an attractor focus.

- Region R_8. The state equation in $R_8 = \{(x_0, x_1) \in \mathbb{R} | x_0 < -1, x_1 < -1\}$ is:

$$\begin{cases} \dot{x_0} = -x_0 - s - p + u_0' \\ \dot{x_1} = -x_1 - s - p + u_1' \end{cases}$$

  Local equilibrium point is:

$$(m_8, k_8) = (u_0' - p - s, u_1' - p - s)$$

  Eigenvalues and eigenvectors are equal to Region R_5. The parametric solution is:

$$\begin{cases} x_0(t) = \alpha_0 e^{-t} + m_8 \\ x_1(t) = \alpha_1 e^{-t} + k_8 \end{cases}$$

  Cartesian solution is: $(x_0 - m_8) = C(x_1 - k_8)$. Local equilibrium point (m_8, k_8) is an attractor focus.

From this study we can conclude that for parameter $s > 1$, every local equilibrium point located in central and middle regions, is not an attractive point (Figure 2.3). Only in the out regions there exist attractive equilibrium points so, if they are located inside their corresponding regions, the system will converge to one of them. The final output will then be $+1$ or -1 because of the local equilibrium points position.

On the other hand, for a symmetric template $p_+ = p_- = p$, almost one of the equilibrium points of the out regions must be placed inside its

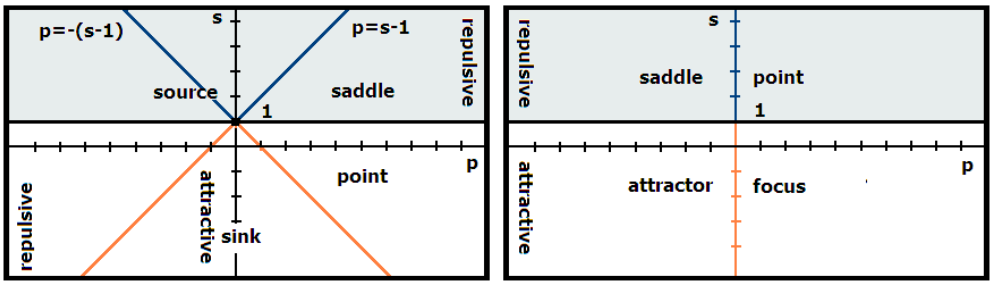


Figure 2.3: Stability results for equilibrium points of central and middle regions.

corresponding region becoming a fixed-point for the general CNN system. On the contrary, if all attractive points are out from their regions,

$$m_5 = u'_0 + p - s > -1, \quad m_6 = u'_0 + p + s < 1,$$

$$m_8 = u'_0 - p - s > -1, \quad m_7 = u'_0 - p + s < 1.$$

we obtain for a positive parameter p,

$$u'_0 > \max\{s - 1 - p, s - 1 + p\} > 0$$

and

$$u'_0 < \min\{-(s-1) - p, -(s-1) + p\} < 0$$

which is a contradiction.

From these results, we can conclude that for a symmetric A_0 template, the system converges to a fixed-point. Furthermore, if parameter $s > 1$, such fixed-point is located inside an out region and so, the final state takes a value $|x_i| > 1$. Final outputs of the CNN system belongs then to $\mathcal{S} = \{(+1,+1),(+1,-1),(-1,-1),(-1,+1)\}$.

2.4.2 Non symmetric parameter range

Although parameters in this section are not symmetric, the procedure to study the system stability will be the same as used before. As we are interested on the CNN dynamics we will study the autonomous case where $u'_k = 0$ without loss of generality. Using the same plane division (Figure 2.2), general equilibrium points in each region are shown in Table 2.3.

- Central region: R_0. Following the same procedure as in the symmetric case, the CNN matrix in this region is A_0 (2.13), with eigenvalues

x_i^*	m_i	k_i
R_0	0	0
R_1	$-\frac{p_+}{s-1}$	$s-\frac{p_+p_-}{s-1}$
R_2	$s-\frac{p_+p_-}{s-1}$	$-\frac{p_-}{s-1}$
R_3	$\frac{p_+}{s-1}$	$-s+\frac{p_+p_-}{s-1}$
R_4	$-s+\frac{p_+p_-}{s-1}$	$\frac{p_-}{s-1}$
R_5	$+p_+-s$	$-p_-+s$
R_6	p_++s	p_-+s
R_7	$-p_++s$	p_--s
R_8	$-p_+-s$	$-p_--s$

Table 2.3: Non symmetric parameter range. Equilibrium points in each of the nine regions where the system is linear for the autonomous case $u_i'=0$.

$\lambda_0=(s-1)+\sqrt{p_+p_-}$, $\lambda_1=(s-1)-\sqrt{p_+p_-}$ and eigenvectors $\vec{v_0}=\left(p_+,\sqrt{p_+p_-}\right)$ and $\vec{v_1}=\left(-p_+,\sqrt{p_+p_-}\right)$.

$$A_0=\begin{pmatrix} s-1 & p_+ \\ p_- & s-1 \end{pmatrix} \tag{2.13}$$

Parametric solutions for the local CNN system are shown in Table 2.4, where (X_0,Y_0) is the result of the variable change (2.10), and ρ,β are described as $C_0=\rho\sin\beta$ and $C_1=\rho\cos\beta$.

$p_+p_->0$	$p_+p_-<0$
$\begin{cases} X_0=C_0e^{(s-1+\sqrt{p_+p_-})t} \\ X_1=C_1e^{(s-1-\sqrt{p_+p_-})t} \end{cases}$	$\begin{cases} X_0=\rho e^{(s-1)t}\cos(\beta-\sqrt{\lvert p_+p_-\rvert}t) \\ X_1=\rho e^{(s-1)t}\sqrt{\lvert\frac{p_-}{p_+}\rvert}\sin(\beta-\sqrt{\lvert p_+p_-\rvert}t) \end{cases}$

Table 2.4: Parametric solution in the central region for the non symmetric case.

Again, CNN parameters will determine the local equilibrium point (m_0,k_0) type.

1. If $\lambda_0\neq\lambda_1\in\mathbb{R}\Rightarrow p_+p_->0$.
   (a) If $s-1<-\sqrt{p_+p_-}$, equilibrium point P_0 is a sink.
   (b) If $-\sqrt{p_+p_-}<s-1<\sqrt{p_+p_-}$, equilibrium point P_0 is a saddle.
   (c) If $s-1>\sqrt{p_+p_-}$, equilibrium point is a source.

2. If $\lambda_0 = \lambda_1 \in \mathbb{R} \Rightarrow p_+ p_- = 0$. So $\lambda_0 = \lambda_1 = s - 1$.
   (a) If $p_+ = p_- = 0$ and $s < 1$, solutions are straight lines approaching the equilibrium point. For $s > 1$, solutions go to infinity straight in every direction.
   (b) If p_+ or p_- are different form zero. For $s < 1$, solutions tend to the equilibrium point (m_0, k_0) tangent to the only straight line solution. For $s > 1$, solutions go to infinity tangent to the straight line solution
   (c) If $s = 1$, both eigenvalues are 0.
3. If $\lambda_i \in \mathbb{C}$, this is $p_+ p_- < 0$ then $\lambda_i = (s-1) \pm j\sqrt{|p_+ p_-|}$ and depending on the real part we will have different solutions.
   (a) If $s < 1$, equilibrium point is a spiral sink.
   (b) If $s > 1$, equilibrium point is a spiral source.
   (c) If $s = 1$, solutions are periodic. Equilibrium point is called center and the trajectory directions are determined by the complex part sign.

- Middle regions: R_1, R_2, R_3, R_4. Like in the symmetric case, CNN matrices A_i, $i = 1, 2, 3, 4$ (2.14) allow us to find the eigenvalues determining the local equilibrium points type. The main difference in this general case is that we work with parameters p_+ and p_- instead of p. Nevertheless, the discussion and the results obtained are the same because the equilibrium point kind only depends on parameter s (Table 2.2). In the middle regions, equilibrium points can be attractor nodes for $s < 1$, improper attractor nodes for $s = 0$ and saddle points for $s > 1$.

$$A_1 = A_3 = \begin{pmatrix} s-1 & 0 \\ p_- & -1 \end{pmatrix}; A_2 = A_4 = \begin{pmatrix} -1 & p_+ \\ 0 & s-1 \end{pmatrix} \tag{2.14}$$

- Out regions: R_5, R_6, R_7, R_8. Exactly the same discussion as in the symmetric case. CNN matrices are equal (2.15), and each local equilibrium point x_i*, $i = 5, 6, 7, 8$ is an attractor focus.

$$A_5 = A_6 = A_7 = A_8 = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \tag{2.15}$$

The dynamic behavior for a non symmetric parameter range is exactly the same as in the symmetric case for real valued eigenvalues ($p_+ p_- \geq 0$). In the case where eigenvalues are complex numbers, the feedback parameters p_+ and p_- have different sign and there can be closed trajectories. Conclusions

when p_+ and p_- have the same sign are then similar as those obtained for a symmetric parameter range. For $s > 1$, the local equilibrium points in the middle and central regions are repulsive while in the out ones are attractive. If one of the local attractive points is located inside its region, it becomes a fixed-point where the system converges.

Summarizing the results obtained, a two neuron CNN in the non-symmetric case can have basically two different dynamic behavior: converge to a fixed-point if one attractive equilibrium point lays inside its region or converge to a closed curve (closed trajectory or limit cycle).

On the other hand, solve the system equations in a parametric way allows to design a program to print the system solutions on the plane. Fixing the parameters relating both neurons $A = (p_+, s, p_-)$, and fixing the external influence (u'_0, u'_1), we start the network evolution at a point $(x_0(0), x_1(0))$, usually located into the unit square (region R_0). We compute the parametric equations $x_0(t), x_1(t)$ in each region R_i, $i = 0, \ldots, 8$, moving parameter t at a fixed ratio δ. To print the CNN solutions, basically one single problem must be solved: how to connect solutions belonging to neighboring regions. An easy way to solve it is by doing a linear interpolation between the last point of one region and the first one of the next region. From this result we will be able to print the CNN system solutions in each case. Along the paper, figures showing the different convergence results like for example those where the system converges to a limit cycle are printed using this program.

2.5 Conclusions

These results are significant for Cellular Neural Networks because they implies that working in certain parameter range, the circuit converges to a binary valued output. For a symmetric template $p_+ = p_- = p$, the system converges to a fixed-point as final output. Furthermore, for parameter $s > 1$, each CNN cell settles down at a stable equilibrium point with magnitude greater than 1 and so, because of the piecewise linear function, the system has binary-value outputs $y_i = \pm 1$, $i = 0, 1$. For $s < 1$ the system is completely stable, it can converge to a fixed-point but it can be out from the saturation regions. So the output can not be $+1$ or -1. It will depend on the local attractive equilibrium points position. From the results obtained, it's reasonable to use the parameter range where the final output is already known in order to find some possible CNN applications. Let us note that the use of the piecewise linear function defining the final outputs has been crucial in order to determine their concrete values.

Due to the geometry of the problem, for the rest of the parameter range

we can have basically two different dynamic behavior: converge to a fixed-point or converge to a closed curve. These two different convergence options, characterize the two neuron CNN dynamics. Next chapters are then classified into the possible problems and uses when the system converges to a fixed-point and those when the system converges to a closed curve.

Part II

Convergence to a fixed-point

Chapter 3

Dependence on initial conditions

Once the system stability has been studied, another problem related to CNN convergence arise: the dependence on initial conditions of the final CNN output. In this chapter we analyze this dependence using the Lyapunov function $L(t)$ (2.6) defined previously to study the stability of the CNN system. We will use it as a quadratic form of the output values y_i, $i = 0, 1$. From this point of view, $L(t) = L(y_0(t), y_1(t))$ can be an elliptic paraboloid, a hyperbolic paraboloid or a parabolic cylinder depending on the CNN weights s and p.

Its existence, and the restriction to work in certain parameter range, assures that the network evolves to a fixed-point. It is logical then to think on use it, in order to find a parameter range where the final output does not depend on initial conditions $(x_0(0), x_1(0))$.

Usually these initial conditions are taken inside the unit square. Hence, $(y_0(0), y_1(0))$ are also inside it. At this point, Max-min theorem can be used to conclude that $L(y_0, y_1)$ has a local minimum in the square to which the system will converge.

Working inside the compact domain defined by the unit square, we study in each of the three cases mentioned before, where does the center and the principal axes of the Lyapunov quadratic form remain. These geometrical elements determine different regions in the unit square, with different final outputs where the system can converge. If the network evolution starts at a point out from the significant lines of the quadratic form, and inside the unit square, the system will converge to the point with lowest Lyapunov function value without any dependence. Moreover, as we have seen in the previous section, for parameter $s > 1$, this point will be one of the four corner points

$\mathcal{S}$ of the unit square.

$$\mathcal{S} = \{(+1,+1)(+1,-1),(-1,+1),(-1,-1)\}.$$

Yet, to completely avoid any dependence on initial conditions, the principal directions and the center of the quadratic form must be out from the unit square, if we consider that the network evolution starts inside it. We are going to find some parameter conditions to assure that these elements do not pass across the unit square. Cutting the Lyapunov function with the four border lines of the unit square, different local curves are obtained. The location of the maximum of these local curves is the key point to find the desired parameter conditions. If this point is out from $[-1,1]$ for any local curve, the system converges to the point with lower Lyapunov function value with independence of initial conditions.

Since, the position and kind of Lyapunov function will let us design templates where the system converge to a point without any dependence on initial conditions. Of course, another way to solve this problem is starting always the network evolution at the same initial point. For instance, we can take the origin $(0,0)$ as this point.

3.1 Lyapunov function

In this section we will see that the Lyapunov function (2.6) is a quadratic form as a function of the output values $y_0(t)$ and $y_1(t)$. On this ground, it can be classified depending on the CNN weights s and p. Theorem 7 shows that in fact it can only be an elliptic paraboloid, a parabolic cylinder or a hyperbolic paraboloid. Its geometry will give us the key points to manage the dependence on initial conditions problem. In order to prove it, we will use Max-Min theorem.

Theorem 6 (Max-Min theorem)**.** If f is a continuous function defined on a compact set $K \subset \mathbb{R}^2$, then f has an absolute maximum and an absolute minimum on K. In particular, f must be bounded on the compact set K.

Proof. Since K is a compact(bounded and closed) set in the plane and f is a continuous function, $f(K)$ is compact. The compact set f(K) is bounded so that f is bounded on K. The compact set f(K) also contains its infimum and supremum, so f has an absolute minimum and maximum in K. □

Theorem 7. The Lyapunov function $L(t)$ (2.6) is a quadratic form as a function of the output values $L(y_0(t), y_1(t))$. It can be classified as a parabolic cylinder for $(s-1)^2 = p^2$, an elliptic paraboloid for $(s-1)^2 > p^2$, and as a hyperbolic paraboloid for $(s-1)^2 < p^2$.

Proof. We will find the diagonal form of the Lyapunov function using the Lagrange method in order to classify it. Let us remember the use of the B-transformation (2.3) to simplify our notation $u_0' = b_0u_0 + b_+u_1 + I$, $u_1' = b_-u_0 + b_0u_1 + I$, and let us define $S = s - 1$.

$$L(t) = -py_0y_1 - \frac{S}{2}(y_0^2 + y_1^2) - u_0'y_0 - u_1'y_1 =$$

$$= -\left(\sqrt{\frac{S}{2}}y_0 + \frac{u_0'}{\sqrt{2S}}\right)^2 - \left(\sqrt{\frac{S}{2}}y_1 + \frac{u_1'}{\sqrt{2S}}\right)^2 - \left(\sqrt{\frac{p}{2}}y_0 + \sqrt{\frac{p}{2}}y_1\right)^2 +$$

$$+\frac{p}{2}y_0^2 + \frac{p}{2}y_1^2 + \frac{1}{2S}(u_0')^2 + \frac{1}{2S}(u_1')^2$$

At this point, we are going to apply different linear changes in order to simplify the quadratic form expression. First linear change is

$$\begin{cases} Y_0 = \sqrt{\frac{S}{2}}y_0, \\ Y_1 = \sqrt{\frac{S}{2}}y_1. \end{cases} \qquad \begin{cases} U_0 = \frac{u_0'}{\sqrt{2S}}, \\ U_1 = \frac{u_1'}{\sqrt{2S}}. \end{cases}$$

$$L(t) = -(Y_0 + U_0)^2 - (Y_1 + U_1)^2 - p\frac{2}{S}Y_0Y_1 + U_0^2 + U_1^2.$$

Second linear change,

$$\begin{cases} Y_0' = Y_1 + U_0, \\ Y_1' = Y_1 + U_1. \end{cases}$$

$$L(t) = -(Y_0')^2 - (Y_1')^2 - \frac{2p}{S}(Y_0' - U_0)(Y_1' - U_1) + U_0^2 + U_1^2.$$

Third linear change,

$$\begin{cases} Y_0' = Y_0'' - \frac{p}{S}Y_1', \\ Y_1' = Y_1''. \end{cases}$$

$$L(t) = -(Y_0'' - \frac{p}{S}Y_1'')^2 - (Y_1'')^2 - \frac{2p}{S}(Y_0'' - \frac{p}{S}Y_1'')Y1'' + \frac{2p}{S}U_0Y_1'' +$$

$$+\frac{2p}{S}U_1(Y_0'' - \frac{p}{S}Y_1'') \underbrace{-\frac{2p}{S}U_0U_1 + U_0^2 + U_1^2}_{K} = -(Y_0'')^2 + 2\frac{p}{S}Y_0''Y_1'' -$$

$$-(\frac{p}{S})^2(Y_1'')^2 - (Y_1'')^2 - \frac{2p}{S}Y_0''Y_1'' + (\frac{2p}{S})^2Y_1''^2 + \frac{2p}{S}U_0Y_1'' + \frac{2p}{S}U_1Y_0'' - 2(\frac{p}{S})^2Y_1''U_1 + K.$$

Fourth linear change,

$$\begin{cases} Y_0'' = Y_0''' + \frac{p}{S}U_1, \\ Y_1'' = Y_1''' \end{cases}$$

$$L(t) = -\left(Y_0''' + \frac{p}{S}U_1\right)^2 + \frac{2p}{S}\left(Y_0''' + \frac{p}{S}U_1\right)U_1 - Y_1'''^2(1 - \left(\frac{p}{S}\right)^2)$$

$$+Y_1'''(-2\left(\frac{p}{S}\right)^2 U_1 + \frac{2p}{S}U_0) + K$$

Finally, let us rewrite $A = (1 - \left(\frac{p}{S}\right)^2)$, and $B = (-2\left(\frac{p}{S}\right)^2 U_1 + \frac{2p}{S}U_0)$

$$L(t) = -(Y_0''')^2 - \frac{2p}{S}Y_0'''U_1 + \frac{2p}{S}Y_0'''U_1 - A(Y_1''')^2 + By \underbrace{-\left(\frac{p}{S}\right)^2 U_1^2 + 2\left(\frac{p}{S}\right)^2 U_1^2 + K}_{K'} =$$

$$= -(Y_0''')^2 - A(Y_1''')^2 + BY_1''' + K' = L(t)$$

Now, depending on parameters A and B we can have different quadratic forms (Figure 3.1).

1. If A=0 $\Rightarrow S^2 = p^2$, $L(t)$ is a parabolic cylinder.
2. If $A \neq 0$ and $A > 0 \Leftrightarrow S^2 > p^2$, $L(t)$ is an elliptic paraboloid.
3. If $A \neq 0$ and $A < 0 \Leftrightarrow S^2 < p^2$, $L(t)$ is a hyperbolic paraboloid.

□

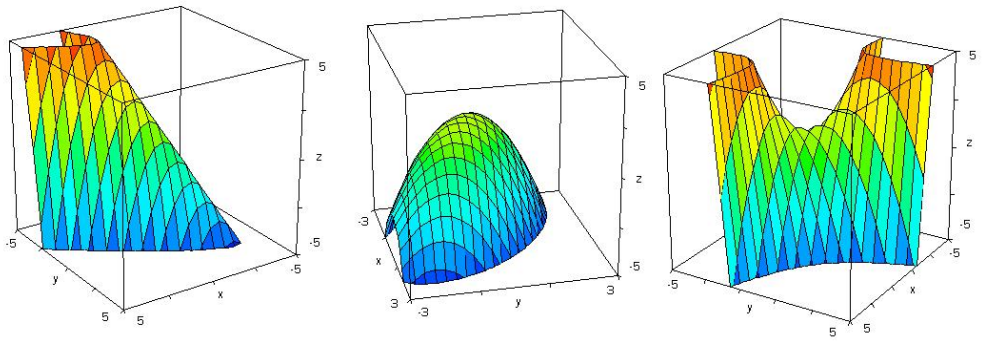


Figure 3.1: Lyapunov functions in each of the three different cases. A parabolic cylinder is obtained for parameters $p = s - 1 = 2$, $u_0' = 1, u_1' = 3$, an elliptic paraboloid for parameters $p = 1 < s - 1 = 2$, $u_0' = u_1' = 2$, and a hyperbolic paraboloid for $p = 2 > s - 1 = 1$, $u_0' = u_1' = 2$.

At this point, and using the Max-Min theorem, we are able to see that the Lyapunov function has a minimum in the unit square where $y_i(t)$ belongs thanks to the piecewise linear function.

Using the Lyapunov function (Figure 3.1) as a quadratic form of y_0, y_1, and working inside the unit square domain D=$[-1,1]\times[-1,1]$, L(y_0,y_1) is a $\mathcal{C}^\infty$-function of two variables. It is a two variables polynomial defined in a closed and bounded domain D$\subset\mathbb{R}^2$. Therefore, from the Max-Min Theorem L(y_0,y_1) has a minimum in D.

3.2 Dependence problem

To study the dependence on initial conditions, we will use the Lyapunov function geometry in each of the three cases studied before. We are going to find their principal axes and the center of the original Lyapunov function. To do it, we can not use the Lagrange study used before, but we can do an orthogonal diagonalisation to find the center and the axes.

Let us rewrite the Lyapunov function (2.6) in a matrix way.

$$L(t)=\begin{pmatrix}y_0 & y_1\end{pmatrix}^T\underbrace{\begin{pmatrix}-(s-1)/2 & -p/2\\ -p/2 & -(s-1)/2\end{pmatrix}}_{L}\begin{pmatrix}y_0\\ y_1\end{pmatrix}-\begin{pmatrix}u_0' & u_1'\end{pmatrix}^T\begin{pmatrix}y_0\\ y_1\end{pmatrix}$$

Let us consider a rotation of angle θ centered at the origin.

$$R_\theta=\begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}$$

Applying the rotation to the Lyapunov function, we make a base change in order to diagonalize the matrix of the quadratic form.

$$D=R_\theta^T\begin{pmatrix}-(s-1)/2 & -p/2\\ -p/2 & -(s-1)/2\end{pmatrix}R_\theta=\begin{pmatrix}\lambda_0 & 0\\ 0 & \lambda_1\end{pmatrix}$$

This implies that $p=0$ or $\sin^2\theta=\cos^2\theta$. If $p=0$, we already have a diagonal form and so it's not necessary to diagonalize. The diagonal matrix is then:

$$L=D=\begin{pmatrix}-(s-1)/2 & 0\\ 0 & -(s-1)/2\end{pmatrix}$$

If $\sin^2\theta=\cos^2\theta$, then $\tan\theta=\pm1\Leftrightarrow\theta=\frac{\pi}{4}+k\frac{\pi}{2},k\in\mathbb{Z}$. For instance, for $\theta=\frac{\pi}{4}$, and $k=0$, the diagonal form is:

$$D=\begin{pmatrix}-(s-1+p)/2 & 0\\ 0 & -(s-1-p)/2\end{pmatrix}.$$

Now applying the rotation to $L(t)$ in this case, we obtain:

$$L = R_\theta D R_\theta^T = \begin{pmatrix} y_0 & y_1 \end{pmatrix}^T \underbrace{R_\theta D R_\theta^T}_{L} \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} - \begin{pmatrix} u_0' & u_1' \end{pmatrix}^T R_\theta R_\theta^T \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} =$$

$$= \left(R_\theta^T \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \right)^T D \left(R_\theta^T \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \right) - \left(R_\theta^T \begin{pmatrix} u_0' \\ u_1' \end{pmatrix} \right)^T \left(R_\theta^T \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} \right)$$

Let us rename

$$R_\theta^T \begin{pmatrix} y_0 \\ y_1 \end{pmatrix} = \begin{pmatrix} Y_0 \\ Y_1 \end{pmatrix} \quad R_\theta^T \begin{pmatrix} u_0' \\ u_1' \end{pmatrix} = \begin{pmatrix} U_0 \\ U_1 \end{pmatrix}$$

The diagonal quadratic form is then:

$$L(t) = \lambda_0 Y_0^2 + \lambda_1 Y_1^2 - U_0 Y_0 - U_1 Y_1,$$

where $\lambda_0 = -\frac{s-1+p}{2}$ and $\lambda_1 = -\frac{s-1-p}{2}$.

To find the center and the principal axes, we must now distinguish the different Lyapunov function cases. If $L(t)$ is an elliptic paraboloid then $S^2 > p^2$, if it is a hyperbolic paraboloid then $S^2 < p^2$ and at last , if it is a parabolic cylinder, parameters fulfill $S^2 = p^2$, where $S = s - 1$.

In the elliptic and hyperbolic cases, principal axes of the Lyapunov function $L(t)$ are described by lines $Y_0 - \frac{1}{2\lambda_0} U_0 = 0$, and $Y_1 - \frac{1}{2\lambda_1} U_1 = 0$.

$$L(t) = \lambda_0 1 \left(Y_0 - \frac{1}{2\lambda_0} U_0 \right)^2 + \lambda_1 \left(Y_1 - \frac{1}{2\lambda_1} U_1 \right)^2 - \left(\frac{U_0}{2\lambda_0} \right)^2 - \left(\frac{U_1}{2\lambda_1} \right)^2$$

Undoing the changes made to diagonalize, and choosing the angle $\theta = \frac{\pi}{4}$, we obtain $L(t)$-principal directions.

$$\begin{aligned} r_0: &\quad y_0 + y_1 = \tfrac{-1}{p+S}(u_0' + u_1') \\ r_1: &\quad -y_0 + y_1 = \tfrac{1}{p-S}(-u_0' + u_1') \end{aligned}$$

The Lyapunov function center can be found via the axes intersection.

$$(C_0, C_1) = \frac{1}{S^2 - p^2} \left(-S u_0' + p u_1', p u_0' - S u_1' \right)$$

At this point, we study the center, and the principal directions possible positions in the plane. To avoid the dependence on initial conditions, in the elliptic case, center (C_0, C_1) must be out from the unit square, and the axes must not pass across it. In the hyperbolic case, center (C_0, C_1) must be again out from the square, but only one of the axes, determined by the direction of

the eigenvector associated to the biggest eigenvalue, must not pass across the unit square. At last, in the parabolic cylinder case where parameters fulfill $S^2 = p^2$, one of the eigenvalues λ_i is 0.

If $\lambda_0 \neq 0$, and $\lambda_1 = 0$, the Lyapunov function $L(t)$ is

$$L(t) = \lambda_0 Y_0^2 + U_0 Y_0 + U_1 Y_1 = \lambda_1 \left(Y_0 + \frac{1}{2\lambda_0} U_0 \right)^2 - \frac{U_0}{2\lambda_0}^2 + U_1 Y_1$$

Principal direction is

$$r_0 : y_0 + y_1 = \frac{1}{2\lambda_0}(u_0' + u_1').$$

This line divides the square into two different convergence regions. Therefore, if it does not pass across the unit square, we avoid the dependence on the initial conditions problem. On the other hand, if $\lambda_0 = 0$, and $\lambda_1 \neq 0$, the Lyapunov function is

$$L(t) = \lambda_1 Y_1^2 + U_0 Y_0 + U_1 Y_1 = \lambda_1 \left(Y_1 + \frac{1}{2\lambda_1} U_1 \right)^2 - \frac{U_1}{2\lambda_1}^2 + U_0 Y_0$$

Principal direction in this case is

$$r_1 : -y_0 + y_1 = \frac{1}{2\lambda_1}(-u_0' + u_1')$$

In order to see and understand how to use a Lyapunov function in the dependence on initial conditions problems, let us study it in a couple of examples.

3.3 Example

Using the hyperbolic paraboloid case for parameters $p = 2$, $S = s - 1 = 1$ and $u_0' = u_1' = 2$, the Lyapunov function is:

$$L(t) = -2y_0 y_1 - \frac{1}{2}(y_0^2 + y_1^2) - 2y_0 - 2y_1$$

The associated eigenvalues and eigenvectors are $\lambda_0 = -3/2$, $\lambda_1 = 1/2$, $\vec{v_0} = (\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$, and $\vec{v_1} = (-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$. The hyperbolic paraboloid center is

$$(C_0, C_1) = (-\frac{2}{3}, -\frac{2}{3})$$

Principal direction associated to the first eigenvalue is $y_1 = y_0$. The second one, which is the projection of the concave parabola separating the square in two different convergence regions, is $y_1 = -y_0 - \frac{4}{3}$.

In Figure 3.2, we can see two different views of the Lyapunov function in the hyperbolic case. Principal direction, which divides the unit square into two different convergence regions, can be seen as a plane cutting the quadratic form. In this case, the system will converge to $(+1, +1)$ or $(-1, -1)$ depending on where the network evolution starts.

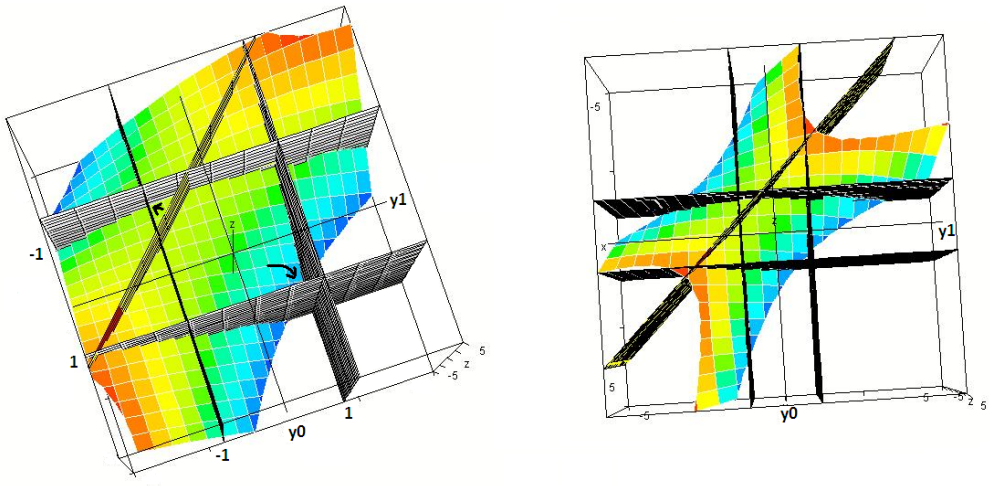


Figure 3.2: Hyperbolic paraboloid for $s - 1 = 1$ and $p = 2$ with principal directions across the unit square.

Let us note that if principal direction do not cross the unit square, then starting the network evolution at any point inside it, the system converges to the same final output. This output will depend on the CNN parameters, and is the point where the Lyapunov function takes its minimum. To see this dependence we print on the plane the level curves in each case as can be seen in Figure 3.3. Clearly, we can see that the network evolution will depend on the initial conditions position because principal directions pass across the unit square.

From this study we are able now to find sufficient conditions for the CNN parameters in order to avoid the dependence on initial conditions. Our main objective is to find a way to assure that the Lyapunov function principal directions do not cross the unit square. We will cut the quadratic form with the unit square borders $y_i = \pm 1$. From this intersection, four different curves are obtained. If the maximum of these curves lies out from $[-1, 1]$, principal

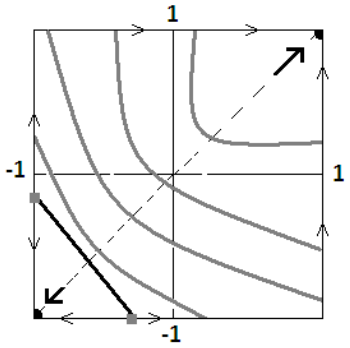


Figure 3.3: Level curves in the hyperbolic case for parameters $S = 1$ and $p = 2$.

axes will not pass across the unit square. These axes divide the square into different convergence regions. Therefore, if they do not pass across it, we will not have such a dependence.

In order to find it, let us cut the Lyapunov function (2.6) with planes $y_0 = \pm 1$ and $y_1 = \pm 1$, in order to study where does the maximum of the intersection curve remains. If the maximum is out from $[-1, 1]$, there is no dependence on initial conditions.

1. $L(t) \bigcap \{y_0 = 1\}$. Intersection curve is

$$L(1, y_1(t)) = -py_1 - \frac{S}{2}(1 + y_1^2) - u_0' - u_1' y_1.$$

To find the maximum of this curve, we look for its derivative to be zero.

$$\frac{d}{dt} L(1, y_1(t)) = -\frac{dy_1}{dt}(p + Sy_1 + u_1') = 0$$

Solving this equation, the extreme candidate is $y_1 = -\frac{p+u_1'}{S}$ for $S \neq 0$. Our purpose is to find the necessary conditions to make this maximum to be out form $[-1, 1]$. For this, parameters must fulfill $|p + u_1'| > |S|$.

2. $L(t) \bigcap \{y_0 = -1\}$. Intersection curve is

$$L(-1, y_1(t)) = +py_1 - \frac{S}{2}(1 + y_1^2) + u_0' - u_1' y_1$$

$$\frac{d}{dt} L(-1, y_1(t)) = \frac{dy_1}{dt}(p - Sy_1 - u_1') = 0$$

The extreme candidate is $y_1 = -\frac{p-u_1'}{S}$ for $S \neq 0$. Again, if parameters fulfill $|p - u_1'| > |S|$, the maximum is out from the unit square.

3. $L(t) \bigcap \{y_1 = 1\}$. Intersection curve is

$$L(y_0(t), 1) = -py_0 - \frac{S}{2}(1 + y_0^2) - u_0' y_0 - u_1'$$

$$\frac{d}{dt} L(y_0(t), 1) = -y_0'(p + Sy_0 + u_0') = 0$$

The extreme candidate is $y_0 = -\frac{p+u_0'}{S}$ for $S \neq 0$. If parameters fulfill $|p + u_0'| > |S|$, the maximum is out from the unit square.

4. $L(t) \bigcap \{y_1 = -1\}$. Intersection curve is

$$L(y_0(t), -1) = +py_0 - \frac{S}{2}(1 + y_0^2) + u_0' y_0 - u_1'$$

$$\frac{d}{t} L(y_0(t), 1) = y_0'(p - Sy_0 - u_0') = 0$$

The extreme candidate is $y_0 = \frac{p-u_0'}{S}$ for $S \neq 0$. If parameters fulfill $|p - u_0'| > |S|$, the maximum is out from the unit square.

From this study, we can conclude that working with parameters p and S fulfilling equation (3.1), the maximum in all the possible cases, is out from $[-1, 1]$. Principal direction dividing the square into different converge regions do not cross the unit square. Hence, there is no dependence on initial conditions.

$$|p \pm u_i'| > |S|, \quad i = 0, 1. \tag{3.1}$$

3.4 Conclusions

From the existence of the Lyapunov function, we have seen that it can be used to avoid the problem of the dependence on initial conditions. Using its geometry as a quadratic form of the output values $y_0(t)$ and $y_1(t)$, which depends on the CNN parameters, we can work in a parameter range where its center is out from the unit square. Principal direction dividing it into different convergence regions do not cross the unit square. Starting the network evolution at any point $(x_0(0), x_1(0))$ inside the unit square, the CNN system will converge to the point with lowest Lyapunov function value.

To find this parameter range, we have cut the quadratic form with planes delimiting the unit square obtaining four different curves. This let us study where does the maximum of these curves remains. Imposing this maximum to be out from $[-1, 1]$, we have obtained the parameter conditions were is needed to work in order to solve this problem.

Chapter 4

On CNN applications

Although the autonomous two neuron CNN is determined by simple general equations defining a non linear system, its dynamic behavior allows the system to converge either to a fixed-point or to closed curve. The way each cell is related with its neighbors and the use of the piecewise linear function to define the final outputs, characterize its dynamics and limits the different problems that can be solved.

Many of the usual tasks that should be dealt with a CNN, requires a relation between an input and an output. For this, we will focus our study in the case where the system converges to a fixed-point as final output. Usually, in the literature, the output choice is the external state at the end of the process y_i, and the internal states $x_i(0)$ or the external ones $u_i, i = 0, 1$, are chosen as input. The consequences of this choice are the different problems that can be solved.

Choosing initial conditions $x_i(0)$, $i = 0, 1$ as inputs, we can use the CNN system to reproduce some probability distributions. For certain parameter range, the different final outputs obtained from the stability analysis are $\mathcal{S} = \{(\pm 1, \pm 1)\}$. They can be used to define a probability space $(\mathcal{S}, \mathcal{A}, p)$ where $\mathcal{A}$ is a σ-algebra, and p is a probability function. On this ground, we will discuss about different ways to assign probabilities to each point in $\mathcal{S}$ using statistical results, and the Lyapunov function geometry.

On the other hand, choosing the external inputs u_i, $i = 0, 1$ as inputs, we can reproduce linear functions, find a way to design templates performing some input-output relations, and solve classification problems. Each one of these applications need the system to converge to a fixed-point. Again, parameters must fulfill some restrictions. To reproduce linear functions, only one restriction is needed: work inside the unit square $[-1, 1] \times [-1, 1]$. Template design and classification problems requires the use of the Lyapunov function so, symmetric parameter range, and a self feedback parameter larger

than one are needed in order to assure the final outputs existence in $\mathcal{S}$.

To design templates, we will construct a map in the plane relating the CNN parameters, the inputs, and the outputs. This map will be crucial not only to design but also to address another important problem, the composition of different templates.

The results obtained from the classification problems let us use the CNN system to reproduce different Boolean functions. Finally, we will try to reproduce the header action of a universal Turing machine.

4.1 Probability distributions

Choosing the internal inputs $x_i(0)$, $i = 0, 1$ in the unit square as inputs for the CNN system, we can reproduce some probability distributions. To do it, let us consider the experiment of running the CNN system with these initial conditions in order to obtain the external inputs y_0, y_1. Outputs y_i can be considered as discrete random variables assuming values $+1$ and -1. For this random event we need the correspondence input-output to be unique. Therefore, we are going to use the Lyapunov function (2.6) inside the unit square for parameter $s > 1$ where it's known that the system converge to one of the points in $\mathcal{S}$.

The probability space where we are going to work is $(\mathcal{S}, \mathcal{A}, p)$ where $\mathcal{S}$ is the sample space, $\mathcal{A}$ is the sigma algebra, and p the probability function. There are different ways to define the probability function p. One way is using statistic results of convergence. Taking the initial conditions $x_i(0)$, $i = 0, 1$ uniformly distributed inside $[-1, 1]$:

$$x_0 = -1 + \frac{k_0}{n}, n \in \{0, \ldots, 2n\}, \quad x_1 = -1 + \frac{k_1}{m}, m \in \{0, \ldots, 2m\},$$

and running the CNN with these initial conditions, we count how many times the network has converged to one of the four possible output points. These results let us define a $(y_0, y_1)-$joint probability function $p : \mathcal{S} \to [-1, 1]$ as

$$p(i, j) = P(y_0 = i, y_1 = j) = \frac{\sharp\text{times converging to } (i, j)}{(2n + 1)(2m + 1)} \quad i, j \in \{-1, 1\}$$

The correspondent marginal density functions of y_0 and y_1 are then:

$$p_{y_0}(k) = P(y_0 = k) = \sum_{l \in \{-1,1\}} P(y_0 = k, y_1 = l) \quad k \in \{-1, 1\}$$

$$p_{y_1}(l) = P(y_1 = l) = \sum_{k\in\{-1,1\}} P(y_0 = k, y_1 = l) \quad l \in \{-1, 1\}$$

Outputs y_0 and y_1 are then Bernoulli random variables with probability $p_i = P(y_i = 1)$, and $q_i = 1 - p_i = P(y_i = -1)$ for $i = 0, 1$.

Another way to define a probability function is the geometric one. Using the Lyapunov function $L(y_0, y_1)$ as a quadratic form of (y_0, y_1), we study the regions of the unit square where its minimum remains. The area of this region will be defined as the probability function.

To begin, let us remember that depending on parameters s and p, $L(y_0, y_1)$ can be an hyperbolic paraboloid if $|s - 1| < |p|$, a parabolic cylinder if $|s - 1| = |p|$ or an elliptic paraboloid if $|s - 1| < |p|$ (Figure 3.1) as we have seen previously.

Like in the dependence on initial conditions study, this quadratic form becomes diagonal doing a rotation of certain angle centered at the origin. $L(y_0, y_1)$ axes are then parallel to oy_0 and oy_1. Applying a translation, the Lyapunov function center moves to the origin. This result let us recognize L, find its principal elements, its principal directions, and its center.

$$\begin{cases} y_0 - C_0 = -(y_1 - C_1) \\ y_0 - C_0 = +(y_1 - C_1) \end{cases} \qquad (C_0, C_1) = \left(\frac{-(s-1)u_0' + pu_1'}{(s-1)^2 - p^2}, \frac{pu_0' - (s-1)u_1'}{(s-1)^2 - p^2} \right)$$

Theses axes divide the unit square into different convergence regions depending on the center's position on the plane: inside or outside the unit square. Printing the level curves of the Lyapunov function (Figure 4.1), we find in all the cases that one of the principal axes, which correspond to the eigenvector of maximum eigenvalue, defines two different convergence regions. This study let us define a $(y_0, y_1)-$joint probability distribution as:

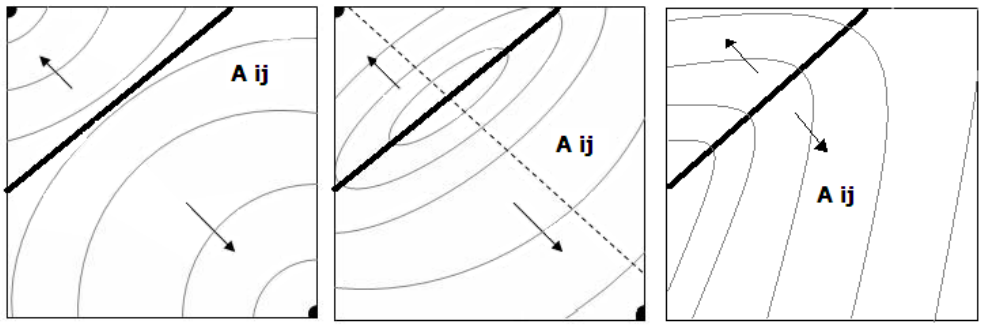


Figure 4.1: Level curves and principal axis in the paraboloid, hyperbolic and cylinder cases.

$$p(i,j) = P(y_0 = i, y_1 = j) = \frac{A_{ij}}{4} \quad i,j \in \{-1,1\},$$

where $A_{ij} \geq 0$ is the area of the region containing the local minimum point (i,j) of the Lyapunov function for $i,j = \pm 1$.

The correspondent marginal density functions of $y_i \sim$Bernoulli(p_i) are

$$p_{y_i}(k) = P(y_i = k) = \sum P(y_i = k, y_j = l) \quad i,j,k,l \in \{-1,1\}$$

At last, let us note that Bernoulli variables y_0 and y_1 are not independent. For certain parameters s and p, like for example those of Figure 4.1, y_0 and y_1 are not independent because $P(y_0 = 1, y_1 = 1) = 0 \neq P(y_0 = 1)P(y_1 = 1)$.

$$P(y_0 = 1) = P(y_0 = 1, y_1 = 1) + P(y_0 = 1, y_1 = -1) = 0 + \tfrac{A_{1-1}}{4}$$
$$P(y_1 = 1) = P(y_0 = 1, y_1 = 1) + P(y_0 = -1, y_1 = 1) = 0 + \tfrac{A_{-11}}{4}$$

4.2 Reproducing linear functions

The CNN dynamical system converges to a fixed-point for certain parameter range, as we have seen in the stability analysis. Output values, defined by the piecewise linear function (1.2), are restricted to be either ± 1 or $x_i \in (-1,1)$ depending on where does the system converges. Commonly, CNN applications are restricted to the case where it has binary valued outputs. Nevertheless, possible CNN applications for the system converging to a fixed-point located inside the unit square, can be an interesting subject of study. We will see that, in this case, two neuron CNNs can be used to reproduce linear functions $y = F(u)$. Actually, input-output relation inside region R_0 is linear because the system dynamic behavior in this case is purely linear.

In order to reproduce a linear functional relation (4.1) between an input and an output, let us take u_i, $i = 0,1$ as input variables, and y_i, $i = 0,1$ at the end of the process, as the output ones. We will work inside the unit square where piecewise linear function defines $y_i = x_i$, $i = 0,1$. From the geometrical stability analysis explained in Chapter 2, we know that CNN systems can converge to a fixed-point located in the unit square, for certain parameter conditions.

$$\begin{aligned} y_0 &= F_0(u_0, u_1) \\ y_1 &= F_1(u_0, u_1) \end{aligned} \tag{4.1}$$

The CNN system settle down to the steady state if $\dot{x}_i = 0$. From this, we obtain the system equations

$$\begin{cases} 0 = -x_0 + sy_0 + p_+ y_1 + u_0' \\ 0 = -x_1 + sy_1 + p_- y_0 + u_1' \end{cases}$$

Working inside the unit square where $x_i = y_i$, we solve this system finding y_i as a function of $u_i, i = 0, 1$.

$$\begin{cases} y_0^* = \frac{1}{(s-1)^2 - p_+p_-}\left(p_+u_1' - (s-1)u_0'\right) \\ y_1^* = \frac{1}{(s-1)^2 - p_+p_-}\left(p_-u_0' - (s-1)u_1'\right) \end{cases}$$

Now, applying the B-transformation (2.3) to work in u_i-plane we obtain:

$$\begin{cases} y_0^* = \left(\frac{b_0(1-s)+p_+b_-}{(s-1)^2-p_+p_-}\right)u_0 + \left(\frac{b_+(1-s)+p_+b_0}{(s-1)^2-p_+p_-}\right)u_1 + I\frac{1-s+p_+}{(s-1)^2-p_+p_-} \\ y_1^* = \left(\frac{b_0p_-+(1-s)b_-}{(s-1)^2-p_+p_-}\right)u_0 + \left(\frac{p_-b_++(1-s)b_0}{(s-1)^2-p_+p_-}\right)u_1 + I\frac{1-s+p_-}{(s-1)^2-p_+p_-} \end{cases}$$

This result gives us a linear function $(y_0, y_1) = (F_0(u_0, u_1), F_1(u_0, u_1)$ relating the external inputs $u_i, i = 0, 1$ and the final outputs.

$$y_0 = F_0(u_0, u_1) = A_{00}u_0 + A_{01}u_1 + C_0$$

$$y_1 = F_1(u_0, u_1) = A_{10}u_0 + A_{11}u_1 + C_1$$

Parameters A_{00}, A_{01}, A_{10} and A_{11} are:

$$A_{00} = \frac{b_0(1-s) + p_+b_-}{(s-1)^2 - p_+p_-} \quad A_{01} = \frac{b_+(1-s) + p_+b_0}{(s-1)^2 - p_+p_-} \quad C_0 = I\frac{1-s+p_+}{(s-1)^2 - p_+p_-}$$

$$A_{10} = \frac{b_0p_- + (1-s)b_-}{(s-1)^2 - p_+p_-} \quad A_{11} = \frac{p_-b_+ + (1-s)b_0}{(s-1)^2 - p_+p_-} \quad C_1 = I\frac{1-s+p_-}{(s-1)^2 - p_+p_-}$$

Let us note that this functional relation is consistent if and only if the external variables y_i are located inside the unit square. Therefore, functions must be defined as $F_i : [-1, 1] \to [-1, 1]$. In addition, linear parameters must fulfill conditions $|A_{00}u_0 + A_{01}u_1 + C_0| \leq 1$ and $|A_{10}u_0 + A_{11}u_1 + C_1| \leq 1$.

4.3 Template design

In this section we will study the template design problem in order to relate an input to a desired output. The geometry and position on the plane of the Lyapunov function will give us the key point. Choosing the external inputs u_i as inputs, and using the Lyapunov function (2.6), we will construct a map. This convergence map relates u_i, the parameters of the cloning template, and the final outputs y_i. From this map, we will find a recipe to find the template parameters $\tau = (I, b_0, b_+, b_-, p, s)$.

Of course, to make this happens, we need some restrictions on the CNN parameters. Those relating both neurons must be symmetric $p_+ = p_- =$

p and the center element of the cloning template $s > 1$. From this, the magnitude of the CNN system fixed-points is bigger than 1, and so the output function $y_i = f(x_i)$ takes values $\{+1, -1\}$.

To study where does the system converges, let us now compare the Lyapunov function (2.6) at this four possible final outputs

$$\mathcal{S} = \{(+1,+1),(-1,+1),(+1,-1),(-1,-1)\},$$

in order to find where does its minimum remains. This comparison will let us relate the CNN external inputs (u_0, u_1) with the final outputs (y_0, y_1) obtaining the convergence map.

Firstly, we will work with u'_i, $i = 0, 1$ (2.3). Then, we will apply the B-transformation to relate the CNN parameters with the external inputs $u_i, i = 0, 1$.

$$L(+1,+1) = -p - \frac{S}{2}(1+1) - u'_0 - u'_1 = -p - S - u'_0 - u'_1$$

$$L(+1,-1) = +p - \frac{S}{2}(1+1) - u'_0 + u'_1 = p - S - u'_0 + u'_1$$

$$L(-1,+1) = +p - \frac{S}{2}(1+1) + u'_0 - u'_1 = p - S + u'_0 - u'_1$$

$$L(-1,-1) = -p - \frac{S}{2}(1+1) + u'_0 + u'_1 = -p - S + u'_0 + u'_1$$

Comparing these four different Lyapunov function values, we find its minimum depending on the CNN parameters, and $u'_i, i = 0, 1$ as can be seen in Table 4.1.

$L(+1,+1) \leq L(-1,-1)$	$-p - S - u'_0 - u'_1 \leq -p - S + u'_0 + u'_1$	$u'_0 + u'_1 \geq 0$
$L(+1,+1) \leq L(-1,+1)$	$-p - S - u'_0 - u'_1 \leq p - S + u'_0 - u'_1$	$p + u'_0 \geq 0$
$L(+1,+1) \leq L(+1,-1)$	$-p - S - u'_0 - u'_1 \leq p - S - u'_0 + u'_1$	$p + u'_1 \geq 0$
$L(-1,+1) \leq L(+1,-1)$	$p - S + u'_0 - u'_1 \leq p - S - u'_0 + u'_1$	$u'_0 \leq u'_1$
$L(-1,-1) \leq L(-1,+1)$	$-p - S + u'_0 + u'_1 \leq p - S + u'_0 - u'_1$	$u'_1 \geq p$
$L(-1,-1) \leq L(+1,-1)$	$-p - S + u'_0 + u'_1 \leq p - S - u'_0 + u'_1$	$u'_0 \geq p$

Table 4.1: Lyapunov comparison table for symmetric parameters $p_+ = p_-$ and $s > 1$.

At this point, we are able to relate the CNN parameters with the final outputs looking for the minimum value of the Lyapunov function. Figure 4.2 shows the convergence map obtained from the comparison study.

1. $L(+1,+1)$ is the minimum if $\{u'_0 + u'_1 \geq 0; u'_0 \geq -p; u'_1 \geq -p\}$

2. $L(-1,-1)$ is the minimum if $\{u'_0 + u'_1 \leq 0; u'_0 \leq p; u'_1 \leq p\}$

3. $L(-1,+1)$ is the minimum if $\{u'_0 \leq -p; u'_0 - u'_1 \leq 0; u'_1 \geq p\}$

4. $L(+1,-1)$ is the minimum if $\{u'_0 \geq p; u'_0 - u'_1 \geq 0; u'_1 \leq -p\}$

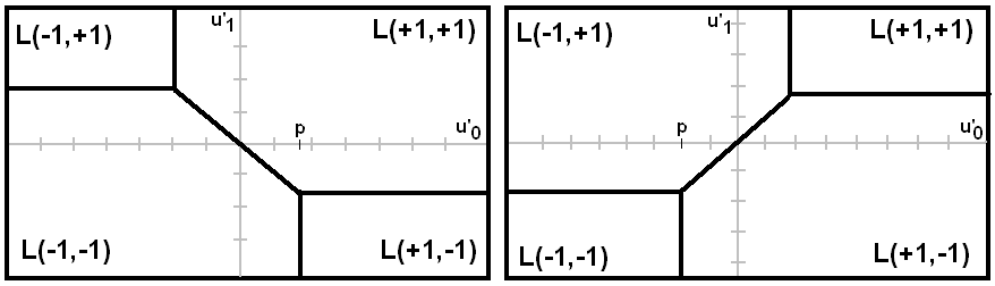


Figure 4.2: Convergence map showing the different convergence regions for parameters $p > 0$ and $p < 0$ in the u'-plane.

Applying the B-transformation defined on (2.3), we relate the external inputs u_i with their correspondent $u'_i, i = 0, 1$. For sake of clarity, it can be written as $T : \mathbb{R}^2 \to \mathbb{R}^2$,

$$T(u_0, u_1) = B(u_0, u_1) + (I, I) = (t \circ B)(u_1, u_1) \tag{4.2}$$

where t is a translation of vector (I, I) and B is a linear application defined by the B-matrix (2.3).

Using this transformation (4.2), we map the director vectors of the boundary lines of the convergence regions, and the corresponding intersection points in order to find the convergence map in the $u_i, i = 0, 1$ plane. To transform the director vectors defining the different lines, we apply only the linear transformation B because translation does not modify lines direction. Let us take a modulus 1 vector $\vec{v} = (\cos\theta, \sin\theta)$ in $\{u_0, u_1\}$-plane, in order to relate it with the director vector $\vec{w} \in \mathbb{R}^2$ of the image lines in the $\{u'_0, u'_1\}$-plane.

$$B(\cos\theta, \sin\theta) = \vec{w}$$

We will first fix parameter $p < 0$ and then we will repeat the study for $p > 0$.

1. Director vectors mapped to lines $u'_0 = \pm p$, $p < 0$.

$$\begin{array}{rccc} B: & \mathbb{R}^2_{\{u_0,u_1\}} & \longrightarrow & \mathbb{R}^2_{\{u'_0,u'_1\}} \\ & r_p & \longrightarrow & u'_0 = p \\ & r_{-p} & \longrightarrow & u'_0 = -p \end{array}$$

$$B\vec{v} = \begin{pmatrix} b_0 & b_+ \\ b_- & b_0 \end{pmatrix} \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} = \begin{pmatrix} b_0\cos\theta + b_+\sin\theta \\ b_-\cos\theta + b_0\sin\theta \end{pmatrix} = \lambda \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow$$

$$\tan\theta = -\frac{b_-}{b_0} \quad \text{and} \quad \lambda = \frac{\mp \det B}{\sqrt{b_0^2 + b_-^2}}$$

The lines incline corresponding to $u_0' = \pm p$ is $m_0 = \frac{-b_-}{b_0} = \tan\theta_0$.

2. Director vectors mapped to lines $u_1' = \pm p$, $p < 0$.

$$\begin{array}{rccc} B: & \mathbb{R}^2_{\{u_0,u_1\}} & \longrightarrow & \mathbb{R}^2_{\{u_0',u_1'\}} \\ & s_p & \longrightarrow & u_1' = p \\ & s_{-p} & \longrightarrow & u_1' = -p \end{array}$$

$$B\vec{v} = \begin{pmatrix} b_0 & b_+ \\ b_- & b_0 \end{pmatrix} \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} = \begin{pmatrix} b_0\cos\theta + b_+\sin\theta \\ b_-\cos\theta + b_0\sin\theta \end{pmatrix} = \mu \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Rightarrow$$

$$\tan\theta = -\frac{b_0}{b_+} \quad \text{and} \quad \mu = \frac{\pm \det B}{\sqrt{b_0^2 + b_+^2}}$$

The lines incline corresponding to $u_1' = \pm p$ is $m_1 = \frac{-b_0}{b_+} = \tan\theta_1$.

3. Intersection point mapped to (p, p), $p < 0$.

$$\begin{array}{rccc} T: & \mathbb{R}^2_{\{u_0,u_1\}} & \longrightarrow & \mathbb{R}^2_{\{u_0',u_1'\}} \\ & (x_p, y_p) & \longrightarrow & (p, p) \end{array}$$

$$T(u_0, u_1) = \begin{pmatrix} b_0 & b_+ \\ b_- & b_0 \end{pmatrix} \begin{pmatrix} x_p \\ y_p \end{pmatrix} + \begin{pmatrix} I \\ I \end{pmatrix} = \begin{pmatrix} b_0 x_p + b_+ y_p + I \\ b_- x_p + b_0 y_p + I \end{pmatrix} = \begin{pmatrix} p \\ p \end{pmatrix} \Rightarrow$$

$$(x_p, y_p) = \frac{p - I}{b_+ b_- - b_0^2}(b_+ - b_0, b_- - b_0)) = \frac{I - p}{b_0}\left(\frac{m_1 + 1}{m_0 - m_1}, \frac{m_0 + 1)}{m_0 - m_1}\right)$$

Let us note that if $b_0^2 - b_- b_+ = detB \neq 0$, then $m_0 \neq m_1$ because $m_0 = -\frac{b_-}{b_0} \neq -\frac{b_0}{b_+}$.

4. Intersection point mapped to $(-p, -p)$, $p < 0$.

$$\begin{array}{rccc} T: & \mathbb{R}^2_{\{u_0,u_1\}} & \longrightarrow & \mathbb{R}^2_{\{u_0',u_1'\}} \\ & (x_{-p}, y_{-p}) & \longrightarrow & (-p, -p) \end{array}$$

$$\Rightarrow \begin{pmatrix} x_{-p} \\ y_{-p} \end{pmatrix} = \frac{1}{b_0^2 - b_+ b_-} \begin{pmatrix} b_0 & -b_+ \\ -b_- & b_0 \end{pmatrix} \begin{pmatrix} -I - p \\ -I - p \end{pmatrix}$$

$$(x_{-p}, y_{-p}) = \frac{I+p}{b_+ b_- - b_0^2}(b_0 - b_+, b_0 - b_-) = \frac{I+p}{b_0}\left(\frac{1+m_1}{m_0 - m_1}, \frac{1+m_0}{m_0 - m_1}\right)$$

Line $u'_0 = u'_1$ in plane $\{u'_0, u'_1\}$ is the lineal transformation of the line $u_1 - y_p = \left(\frac{b_0 - b_-}{b_0 - b_+}\right)(u_0 - x_p)$ connecting points (x_p, y_p) and (x_{-p}, y_{-p}) in the plane $\{u_0, u_1\}$. Simplifying the line equation, we obtain $u_1 = \left(m_1 \frac{1+m_0}{1+m_1}\right) u_0$.

Let us study now where do intersection points go under the linear transformation for a positive parameter p.

1. Intersection point mapped to $(-p, p)$, $p > 0$.

$$\begin{array}{rccc} T: & \mathbb{R}^2_{\{u_0,u_1\}} & \longrightarrow & \mathbb{R}^2_{\{u'_0,u'_1\}} \\ & (x_{-p}, y_p) & \longrightarrow & (-p, p) \end{array}$$

$$T(x_{-p}, y_p) = \begin{pmatrix} b_0 & b_+ \\ b_- & b_0 \end{pmatrix} \begin{pmatrix} x_{-p} \\ y_p \end{pmatrix} + \begin{pmatrix} I \\ I \end{pmatrix} = \begin{pmatrix} -p \\ p \end{pmatrix} \Rightarrow$$

$$\Rightarrow \begin{pmatrix} x_{-p} \\ y_p \end{pmatrix} = \frac{1}{b_o^2 - b_+ b_-} \begin{pmatrix} b_0 & -b_+ \\ -b_- & b_0 \end{pmatrix} \begin{pmatrix} -p - I \\ p - I \end{pmatrix}$$

$$(x_{-p}, y_p) = \left(\frac{I(b_+ - b_0) - p(b_+ + b_0)}{detB}, \frac{p(b_0 + b_-) + I(b_- - b_0)}{detB}\right) =$$

$$= \frac{1}{b_0}\left(\frac{(p-I) - m_1(I+p)}{m_1 - m_0}, m_1 \frac{(p-I) - m_0(p+I)}{m_1 - m_0}\right)$$

2. Intersection point mapped to $(p, -p)$, $p > 0$.

$$\begin{array}{rccc} T: & \mathbb{R}^2_{\{u_0,u_1\}} & \longrightarrow & \mathbb{R}^2_{\{u'_0,u'_1\}} \\ & (x_p, y_{-p}) & \longrightarrow & (p, -p) \end{array}$$

$$T(x_p, y_{-p}) = \begin{pmatrix} b_0 & b_+ \\ b_- & b_0 \end{pmatrix} \begin{pmatrix} x_p \\ y_{-p} \end{pmatrix} + \begin{pmatrix} I \\ I \end{pmatrix} = \begin{pmatrix} b_0 x + b_+ y + I \\ b_- x + b_0 y + I \end{pmatrix} = \begin{pmatrix} p \\ -p \end{pmatrix} \Rightarrow$$

$$\Rightarrow \begin{pmatrix} x_p \\ y_{-p} \end{pmatrix} = \frac{1}{b_0^2 - b_+ b_-} \begin{pmatrix} b_0 & -b_+ \\ -b_- & b_0 \end{pmatrix} \begin{pmatrix} p - I \\ -p - I \end{pmatrix}$$

$$(x_p, y_{-p}) = \left(\frac{I(b_+ - b_0) + p(b_+ + b_0)}{detB}, \frac{I(b_- - b_0) - p(b_- + b_0)}{detB}\right) =$$

$$= \frac{1}{b_0}\left(\frac{m_1(p-I) - (I+p)}{m_1 - m_0}, m_1 \frac{m_0(p-I) - (I+p)}{m_1 - m_0}\right)$$

Line $u'_0 + u'_1 = 0$ in plane $\{u'_0, u'_1\}$ is the lineal transformation of line $(u_1 - y_p) = -\left(\frac{b_0+b_-}{b_0+b_+}\right)(u_0 - x_{-p}) = \left(m_1\frac{m_0-1}{m_1-1}\right)(u_0 - x_{-p})$ connecting points (x_{-p}, y_p), and (x_p, y_{-p}) in $\{u_0, u_1\}$-plane .

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

Table 4.2: Intersection points and boundary lines of the convergence regions for $p < 0$.

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

Table 4.3: Intersection points and boundary lines of the convergence regions for $p > 0$.

From this study, we can now print the boundary lines of the convergence regions in $\{u_0, u_1\}$ plane. Depending on p-sign, we have different regions as can be seen in Tables 4.2 and 4.3.

The convergence map obtained establishes the main points in order to design templates $\tau = (I, b_0, b_+, b_-, p, s)$ making a set of inputs converge to some desired outputs.

Let us suppose to have an amount of points which we want to classify. We have seen that the convergence regions are determined basically on four parameters: two slopes, and two intersection points (m_0, m_1, P_0, P_1). We

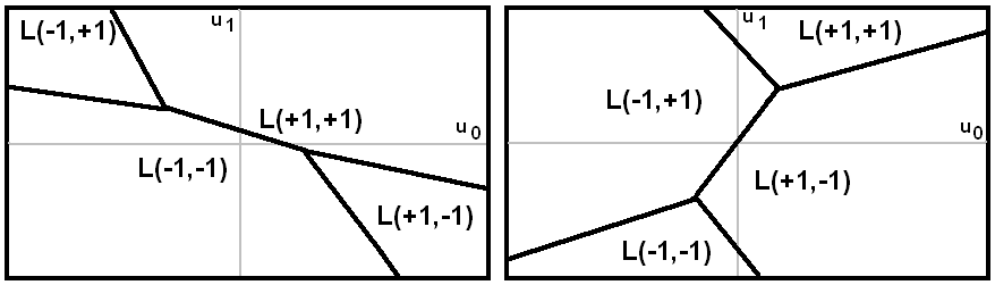


Figure 4.3: Regions of convergence for $p > 0$ and $p < 0$ in the u-plane.

can choose them adequately in order to make the system converge where we want. Choosing two slopes of the four parallel lines,

$$m_0 = -\frac{b_-}{b_0}, \quad m_1 = -\frac{b_0}{b_+},$$

we can determine parameters b_- and b_+ depending on b_0. Choosing two intersection points, we complete the convergence map design. In each particular example, we will decide the parameter p sign depending on the input sets geometry.

For instance, let us suppose a set of points located inside four different circles C_i in the plane $\mathbb{R}^2$ with centers P_i an radius r_i. Each of these sets contains elements which has to be classified into four different elements. These different elements can be mapped to the four final possible outputs $\mathcal{S}$. The convergence map lines should be designed in order to separate each different circle.

$$\begin{aligned}
C_1(P_1, r_1) &= \{(u_0, u_1) \in \mathbb{R}^2 | (u_0 - 6)^2 + (u_1 - 4)^2 \leq 1\} \\
C_2(P_2, r_2) &= \{(u_0, u_1) \in \mathbb{R}^2 | (u_0 - 8)^2 + (u_1 + 2)^2 \leq 1\} \\
C_3(P_3, r_3) &= \{(u_0, u_1) \in \mathbb{R}^2 | (u_0 + 7)^2 + (u_1 - 3)^2 \leq 1\} \\
C_4(P_4, r_4) &= \{(u_0, u_1) \in \mathbb{R}^2 | (u_0 + 5)^2 + (u_1 + 5)^2 \leq 1\}
\end{aligned}$$

Let us note that there are different solutions to solve this problem, and they strongly depend on the position of circles C_i.

To begin, let us choose parameter p sign as negative (choice 0). This choice will determine the geometry of the convergence map, and the line equations. Next, an incline of the line between circles C_1 and C_2 (choice 1).

$$m_0 = -\frac{1}{4} \tag{4.3}$$

Now, we take the incline of the line between circles C_1 and C_3 (choice 2).

$$m_1 = \frac{4}{3}. \tag{4.4}$$

At this point, we can find the line connecting intersection points.

$$u_1 = m_1 \left(\frac{m_0 + 1}{m_1 + 1} \right) u_0 = \frac{3}{7} u_0.$$

Last choices are now the two intersection points $P_0 = (x_p, y_p)$ and $P_1 = (x_{-p}, y_{-p})$ connecting the five different convergence map lines.

$$P_0 = (x_p, y_p) = \frac{I - p}{b_0} \left(\frac{28}{19}, \frac{12}{19} \right) \quad P_1 = (x_{-p}, y_{-p}) = \frac{I + p}{b_0} \left(\frac{28}{19}, \frac{12}{19} \right)$$

which clearly depend on parameters I, p and b_0.

For instance, taking the scalar factor (choice 3),

$$\frac{I + p}{b_0} = -\frac{19}{12} \tag{4.5}$$

and the other one (choice 4),

$$\frac{I - p}{b_0} = \frac{19}{12} \tag{4.6}$$

intersection points are determined, and so the line equations defining a convergence map for this particular example (Table 4.4).

intersection points	
$P_1 = (x_{-p}, y_{-p}) = (7/3, 1)$	$P_0 = (x_p, y_p) = (-7/3, -1)$
line equations	
$3(u_1 - 1) = 4(u_0 - 7/3)$	$4(u_1 - 1) = -(u_0 - 7/3)$
$3(u_1 + 1) = 4(u_0 + 7/3)$	$4(u_1 + 1) = -(u_0 + 7/3)$
$u_1 = 3/7 u_0$	

Table 4.4: Intersection points and lines defining a convergence map.

Once the convergence map is found, we are able to find the six template parameters named $\tau = (I, b_0, b_+, b_-, p, s)$. As $m_0 = -\frac{b_-}{b_0}$ (4.3) and $m_1 = -\frac{b_0}{b_+}$ (4.4), parameters $b_- = \frac{1}{4} b_0$ and $b_+ = -\frac{3}{4} b_0$ are determined by the scalar b_0. Solving the system equations (4.5) and (4.6), parameter $I = 0$ and $p = -19/12 b_0$. A general template solving this problem will then be:

$$\tau = (0, b_0, -\frac{3}{4} b_0, \frac{1}{4} b_0, -\frac{19}{12} b_0, s)$$

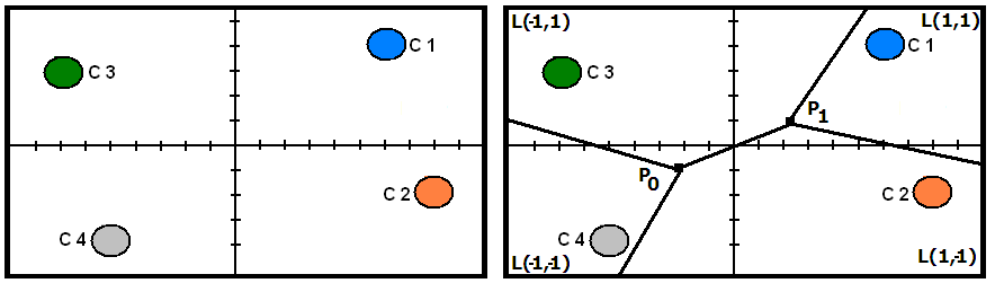


Figure 4.4: In the first image we can see the four different circles. In the second one, we have printed the convergence map classifying them into four different final outputs.

For parameter $p < 0$, we need b_0 to be positive. For example: $b_0 = 4$. This choice gives us τ-template,

$$\tau = (0, 4, -3, 1, -19/3, s)$$

Let us note that parameter s plays no role in the template design problem. It can be fixed to any value bigger than one in order to assure the system convergence to the final outputs $\mathcal{S}$. We can take for instance $s = 2$ and so

$$\tau = (0, 4, -3, 1, -19/3, 2) \tag{4.7}$$

This example is a linearly separable one. Nevertheless, a two neuron CNN can solve problems slightly more difficult than theses ones, like those shown in Figure 4.5.

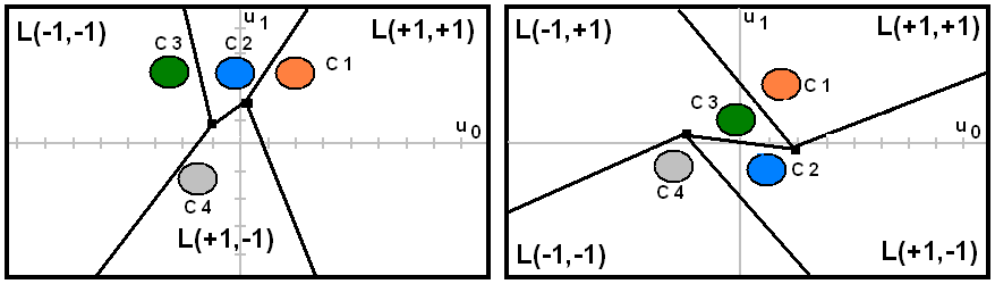


Figure 4.5: Convergence map examples for non linearly separable problems.

From the results obtained, we may think on a new problem: the composition of different templates. If one template is not enough to solve a concrete

problem, we can sequentially apply different templates. Using the convergence map to design a new template ,outputs of the first one are inputs for the second one.

When we compose two templates, the first one shall drive the system to one of the four points in $\mathcal{S}$. Studying the position of these points in the new convergence map obtained from a second template, we can make the system converge to a desired final output. Moreover, this effect can be used to design a template that changes the output assignment of the first one.

Let us suppose that for certain template $\tau_0 = (I, b_0, b_+, b_-, p, s)$, we have obtained as output values $(y_0, y_1) \in \mathcal{S}$. We are going to use these values as new external inputs $(u_0, u_1) \in \mathcal{S}$ of a second template τ_1. To design this second template, we apply the B-transformation to the external inputs in order to obtain the correspondent u'-value. This let us use the convergence map in the u'-plane, and decide which are the new parameters to make possible some desired input-output relation. For instance, to map points of circles C_1 and C_3 to $(-1, 1)$, and points of circles C_2 and C_4 to the same outputs as the first template, we can compose $\tau_0 = \tau$ (4.7) with another one τ_1, to map point $(1, 1)$ to $(-1, 1)$ as can be seen in Figure 4.6.

$$
\begin{array}{cccccc}
 & \tau_0 & & & \tau_1 & \\
\mathbb{R}^2 & \rightarrow & \mathcal{S} & \rightarrow & \mathcal{S} & \\
\vec{u}_{\tau_0} & \rightarrow & \vec{y}_{\tau_0} = \vec{u}_{\tau_1} & \rightarrow & \vec{y}_{\tau_1} & \\
C_1 & \rightarrow & (+1,+1) & \rightarrow & (-1,+1) & \\
C_2 & \rightarrow & (+1,-1) & \rightarrow & (+1,-1) & \\
C_3 & \rightarrow & (-1,+1) & \rightarrow & (-1,+1) & \\
C_1 & \rightarrow & (-1,-1) & \rightarrow & (-1,-1) &
\end{array}
$$

In general, to find a new template changing the initial input-output association, we study where does outputs $(y_0, y_1) \in \mathcal{S}$ of the first one goes under the B-transformation. Let us take the external inputs (u_0, u_1) of a second template inside the output subset $\mathcal{S}$, and let us apply the B-transformation composed with a translation (4.2), to each one of these four points.

- If $(u_0, u_1) = (+1, +1)$, $T(+1, +1) = (b_0 + b_+ + I, b_0 + b_- + I)$
- If $(u_0, u_1) = (+1, -1)$, $T(+1, -1) = (b_0 - b_+ + I, -b_0 + b_- + I)$
- If $(u_0, u_1) = (-1, +1)$, $T(-1, +1) = (-b_0 + b_+ + I, b_0 - b_- + I)$
- If $(u_0, u_1) = (-1, -1)$, $T(-1, -1) = (-b_0 - b_+ + I, -b_0 - b_- + I)$

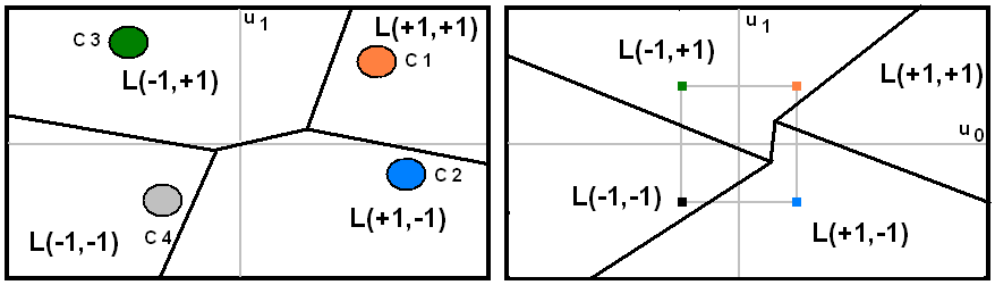


Figure 4.6: Template composition using a convergence map. First figure shows the map obtained from the first template τ_0, and the second one shows a new map obtained from a second template τ_1. Outputs of the first one are printed as inputs in the second convergence map.

Now we are able to define a new convergence map for these four points in order to make the system converge where we want using this second template. For example, to make $T(+1,+1) = (u'_0, u'_1)$ converge to one of the convergence regions, we impose the correspondent parameters to fulfill the necessary conditions to place it inside a convergence region.

To make $T(+1,+1)$ be inside convergence region $L(+1,+1)$ in the $\{u'_0, u'_1\}$-plane,

$$\{b_0 + b_+ + I \geq -p, \quad b_0 + b_- + I \geq -p, \quad 2b_0 + 2I + b_- + b_+ \geq 0\}$$

To be inside convergence region $L(-1,-1)$,

$$\{b_0 + b_+ + I \leq p, \quad b_0 + b_- + I \leq p, \quad 2b_0 + 2I + b_- + b_+ \leq 0\}$$

To be inside convergence region $L(-1,+1)$,

$$\{b_0 + b_+ + I \leq -p, \quad b_0 + b_- + I \geq p, \quad b_+ - b_- \leq 0\}$$

At last, to be inside convergence region $L(+1,-1)$,

$$\{b_0 + b_+ + I \geq p, \quad b_0 + b_- + I \leq -p, \quad b_+ - b_- \geq 0\}$$

Table 4.5 summarize the different conditions for $T(-1,-1)$, $T(1,-1)$ and $T(-1,1)$. Finally from all this study, we have obtained the necessary information to find a recipe to design CNN templates. From the geometry of the convergence map, we have seen that only four parameters are necessary: b_+, b_-, p and , I. This significant result, clearly simplifies the learning problem, and let us design different templates performing some desired input-output

	$T(-1,-1)$	$T(1,-1)$	$T(-1,1$
$L(+1,+1)$	$I - b_0 - b_+ \geq -p$	$b_0 - b_+ + I \geq -p$	$-b_0 + b_+ + I \geq -p$
convergence	$I - b_0 - b_- \geq -p$	$-b_0 + b_- + I \geq -p$	$b_0 - b_- + I \geq -p$
region	$2I - 2b_0 \geq b_- + b_+$	$2I + b_- - b_+ \geq 0$	$2I - b_- + b_+ \geq 0$
$L(-1,-1)$	$I - b_0 - b_+ \leq p$	$b_0 - b_+ + I \leq p$	$-b_0 + b_+ + I \leq p$
convergence	$I - b_0 - b_- \leq p$	$-b_0 + b_- + I \leq p$	$b_0 - b_- + I \leq p$
region	$2I - 2b_0 \leq b_+ - b_- \leq 0$	$2I + b_- - b_+ \leq 0$	$2I + b_- - b_+ \leq 0$
$L(-1,1)$	$I - b_0 - b_+ \leq -p$	$b_0 - b_+ + I \leq -p$	$-b_0 + b_+ + I \leq -p$
convergence	$I - b_0 - b_- \geq p$	$-b_0 + b_- + I \geq p$	$b_0 - b_- + I \geq p$
region	$-b_+ + b_- \leq 0$	$2b_0 - b_+ - b_- \leq 0$	$-2b_0 + b_+ + -b_- \leq 0$
$L(1,-1)$	$I - b_0 - b_+ \geq p$	$b_0 - b_+ + I \geq p$	$-b_0 + b_+ + I \geq p$
convergence	$I - b_0 - b_- \leq -p$	$-b_0 + b_- + I \leq -p$	$b_0 - b_- + I \leq -p$
region	$-b_+ + b_- \geq 0$	$2b_0 - b_+ - b_- \geq 0$	$-2b_0 + b_+ + b_- \geq 0$

Table 4.5: Parameter conditions for inputs $T(-1,-1)$, $T(1,-1)$ and $T(-1,1)$ to be inside each convergence region in order to construct a second template in the template composition problem.

relation depending on the scalar factor b_0. Choosing two slopes m_0 (choice 1) and m_1 (choice 2), and two intersection points $P_0 = (x_p, y_p)$ (choice 3) and $P_1 = (x_{-p}, y_{-p})$ (choice 4) in the p negative case, there is enough information to construct a converge map.

Nevertheless, to design a template, parameters s and b_0 are needed. We have seen that these parameters are not relevant, b_0 is only a scalar factor, and s bigger than one plays an indirect role on the network output.

A general recipe in order to find the CNN parameters requires first to print in the plane the convergence problem in order to visualize the possible solutions. Once p sign is fixed, we find the CNN parameters $\tau = (I, b_0, b_+, b_-, p, s)$ following instructions:

1. choose incline m_0,
2. choose incline m_1,
3. choose intersection point P_0,
4. choose intersection point P_1.

   From these choices, we obtain parameters I, b_+, b_- and p depending on the scalar factor b_0.

5. At last, we choose b_0 and s.

Let us note that using the convergence map (Figure 4.3) we can solve more than linearly separable problems. We may speak of piecewise linear separable problems. On the other hand, 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, one can use the convergence map and the template composition to find which input-output relations can be achieved.

4.4 Classification problems

Using one more time the results obtained from the Cellular Neural Networks stability study, we have seen that working in a symmetric parameter range, CNN system is completely stable, in the sense that it converges to a fixed-point. Moreover, if the central element $s > 1$, fixed-point lays in a region where output values at the end of the process $y_i(\infty)$ are $\{+1, -1\}$. Hence, one may think on using the external inputs u_i to work in this parameter range in order to study the existence of a functional relation between the external inputs u_i's and the final outputs y_i's.

$$y_i = F(u_i)$$

For every input u_i, we can run the CNN system in order to find the correspondent output y_i. Because the set of final states is discrete, this correspondence can be thought as a classification problem where each one of the different classes is defined by the different final states which, will depend on the parameters of the CNN system. This will be the objective of this section: study which input-output functional relations can be performed using a CNN, and scrutinize the way to find the parameters solving such a problems.

Given a set of input points $\{(u_0, u_1)_i, i = 1, \ldots n\}$, we run the CNN system using the adequate CNN parameters. Depending on the position of the fixed-points where the system converges, we obtain the output values in $\mathcal{S} = \{(+1, +1), (-1, +1), (+1, -1), (-1, -1)\}$. Therefore, inputs can be classified into different subsets which are defined by four different final outputs.

In order to tackle the classification problems that can be solved using a CNN, we use the Lyapunov function (2.6). $L(t)$ let us find convergence regions, and learn the adequate CNN parameters in order to make some desired input-output functional relation.

To begin, we restrict the input choice (u_0, u_1) to values ± 1, without loss of generality. Using the B-transformation (2.3), 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 4.2). This may be 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 image points $(i,j) + (\varepsilon, \varepsilon), i, j = -1, 1$, $\varepsilon \neq 0$, and proceed as we have explained before.

To simplify the notation, let us rename $\mathcal{S}$-points with the correspondence shown in Table 4.6. Convergence regions $L(i,j)$ will then be $R(i)$, for $i = 1, 2, 3, 4$.

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

Table 4.6: $\mathcal{S}$-points correspondence.

Nevertheless, both notations will be used along the paper in order to clarify some explanations. Taking the different input points $u_i \in \mathcal{S} = \{1, 2, 3, 4\}$ and using the B-transformation, we impose u_i to converge to an output point $y_i \in \mathcal{S} = \{\pm 1, \pm 1\}$ located in each one of the convergence regions $R(i)$, $B(u_i) = y_i$. From the system equations obtained, we find the CNN parameters defining the cloning template T_i which perform the desired input-output association.

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

$$\begin{cases} b_0 + b_+ + I = 1, \\ b_- + b_0 + I = 1. \end{cases} \tag{4.8}$$

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

$$\begin{cases} -b_0 - b_+ + I = 1, \\ -b_- - b_0 + I = 1. \end{cases} \tag{4.9}$$

Solving the system equations (4.8) and (4.9), we find parameters

$$I = 1, \quad b_+ = b_- = -b_0.$$

This relation is compatible with a 2-neuron CNN. Now we study the four possible outputs for inputs $2 \equiv (1,-1)$ and $4 \equiv (-1,1)$ where $B(2) = (1 - 2b_0, 1 + 2b_0)$ and $B(4) = (1 + 2b_0, 1 - 2b_0)$, summarized in Table 4.7.

Let us note that parameter conditions are incompatible for certain values of b_0. Using those shown in Table 4.7, we obtain that input point 2 can

2	4	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 4.7: Convergence study for points $(-1,1)$ and $(1,-1)$.

converge to outputs $(1,1),(1,-1)$ and $(-1,1)$ depending on parameters p, b_0.

If $-1-2b_0 < -1+2b_0$, then $\max\{-1-2b_0, -1+2b_0\} = -1+2b_0$. This implies that $b_0 > 0$. If $b_0 > 1/2$, then $-1+2b_0 > 0$ and so, for $p > -1+2b_0 > 0$ input 2 converges to $(1,1)$,

$$B(2) = (1,1) \in R(1) \Leftrightarrow p > -1+2b_0, b_0 > 1/2,$$

and for $0 < p < -1+2b_0$, input 2 converges to $(1,-1)$,

$$B(2) = (1,-1) \in R(2) \Leftrightarrow p < -1+2b_0, b_0 > 1/2.$$

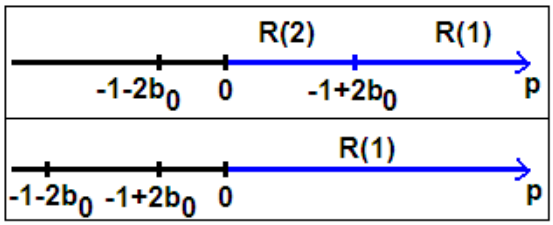


Figure 4.7: Convergence study for input 2 depending on parameter $p > 0$.

Otherwise if $0 < b_0 < 1/2$ and $p > 0$, input point 2 can only converge to $(1,1)$ because $-1+2b_0 < 0$ (Figure 4.7). In a similar way but for a negative parameter b_0, input 2 can converge either to $(1,1)$ or to $(-1,1)$.

However, some input-output associations can not be performed. For instance if $B(3) = (-1,1)$, we have

$$\begin{cases} -b_0 - b_+ + I = -1, \\ -b_- - b_0 + I = 1. \end{cases} \tag{4.10}$$

Solving the system equations (4.8) and (4.10), parameter I must be equal to 0 and 1. Therefore, such an association can not be achieved by one

single 2-neuron CNN. In a similar way, input 3 can not converge to $(1,-1)$ because there is no solution for the system equations obtained from (4.8) and $B(3)=(1,-1)$.

Finally, input 3 converges to itself for parameters, $I=0$ and $b_+=b_-=1-b_0$. The other 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 $(\pm1,\pm1)\rightarrow(\pm1,\pm1)+(\varepsilon,\varepsilon)$ in order to solve $B(i)=(\pm1+\varepsilon,\pm1+\varepsilon)$, $\varepsilon\in\mathbb{R}-\{0\}$, $i=1,2,3,4$.

From equations $B(1)=(1,1)+(\varepsilon,\varepsilon)$ and $B(3)=(-1,-1)+(\varepsilon,\varepsilon)$, we find parameters

$$I=\varepsilon,\quad b_+=b_-=1-b_0.$$

The rest of the 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, input 2 converges to $(1,1)$ if and only if conditions defining this convergence region (4.11) are fulfilled.

$$\begin{cases} -1+2b_0+\varepsilon>-p \\ 1-2b_0+\varepsilon>-p \\ 1-2b_0+\varepsilon\geq 1-2b_0-\varepsilon\Rightarrow\varepsilon\geq 0 \end{cases} \tag{4.11}$$

Parameter conditions are then $p>\max\{-1+2b_0-\varepsilon,1-2b_0-\varepsilon\}$, $\varepsilon\geq 0$. Doing a similar study for the other convergence regions we obtain the rest.

Table 4.8 summarize all possible input-output relations obtained for the case where the first input point 1 converges to itself and $p>0$. We use the two row notation in order to describe it. For example,

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

represents the convergence of each input point $\{1,2,3,4\}$ to output $1\equiv(1,1)$. To design a template τ_1 performing this input-output relation, parameters must fulfill $I=1$, $b_+=b_-=-b_0$, $p>\max\{-1\pm 2b_0\}$ and $s>1$. Choosing $b_0=2$, $p=4>3$ and $s=3$ we find $\tau_1=(I,b_0,b_+,b_-,p,s)=(1,2,-2,-2,4,3)$.

In Appendix B we describe the rest of the input-output relations for $p>0$ and $p<0$ with their correspondent parameter conditions.

Let us note that 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. Therefore, only 3 parameters (b_0,p,s) are relevant to design a CNN template.

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 4.8: Possible outputs for the case where $B(1) = (1, 1)$.

We have found 25 possible convergence options with their correspondent template parameter conditions. Rewriting $T_i = (abcd) = \begin{pmatrix} 1 & 2 & 3 & 4 \\ a & b & c & d \end{pmatrix}$ we summarize the different input-output relations obtained in Table 4.9.

T_1	(1111)	T_2	(2222)	T_3	(1131)	T_4	(3313)	T_5	(2242)
T_6	(1333)	T_7	(3111)	T_8	(2244)	T_9	(4422)	T_{10}	(2442)
T_{11}	(4224)	T_{12}	(3232)	T_{13}	(3434)	T_{14}	(1214)	T_{15}	(1412)
T_{16}	(3432)	T_{17}	(3234)	T_{18}	(4121)	T_{19}	(1232)	T_{20}	(1432)
T_{21}	(3412)	T_{22}	(3214)	T_{23}	(4123)	T_{24}	(2341)	T_{25}	(1234)

Table 4.9: 25 elements converging to one, two, three and four output values.

They can be classified into four sets, depending on the convergence to one , two, three or four different outputs. Let us name

$$S_1 = \{(aaaa), a = 1, \ldots, 4\},$$

the set where any input converges to one single output,

$$S_2 = \{(aaba), (baaa), (aabb), (abba), (abab), a, b = 1, \ldots, 4\},$$

the set where inputs converge to two different outputs,

$$S_3 = \{(abac), (abcb), a, b, c = 1, \ldots 4\},$$

the set where inputs converge to three different outputs and S_4 a set belonging to the permutation group P_4 of four elements.

Observing Table 4.9 we may remark that there are missing relations such as (2222) or (1121), so we will compose the 25 elements found above in order to obtain all the possible input-output relations between points in $\mathcal{S}$ using a two neuron CNN.

The elements composition will be written as a product using the two row notation. For example, taking an arbitrary element T_i and T_7,

$$T_i \circ T_7 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ a & b & c & d \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ c & a & a & a \end{pmatrix}.$$

Beginning with input 1 of T_j, 1 moves to 3, then 3 moves to c in T_i. So the final element moves 1 to c.

Composing for instance $T_1 = (1111)$ with any other input-output relation where input 1 converges to 2, we obtain (2222). For example, composing T_1 and T_5,

$$T_5 \circ T_1 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 2 & 4 & 2 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 2 & 2 & 2 \end{pmatrix}.$$

Using T_1 every input choice converges to output 1. Taking output 1 as a new input we apply T_5 where 1 converges to output 2. In the same way, $T_9 \circ T_1 = (4444)$. So we found all possible elements of S_1-set.

$$S_1 = \{(1111), (2222), (3333), (4444)\}$$

Now we are going to find all the possible input-output relations belonging to S_2. To do it, we start with subset $S_2^{(1)} = \{(aaba), a, b = 1, \ldots 4\} \subset S_2$. Only three elements $T_3 = (1131)$, $T_4 = (3313)$ and $T_5 = (2242)$ has been found in the B-transformation study. Composing them with the rest of S_2-elements, we obtain for example $T_{11} \circ T_3 = (4424)$.

$$T_{11} \circ T_3 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 2 & 4 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 3 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 4 & 2 & 4 \end{pmatrix}.$$

Yet, elements like (1121) can not be found composing T_3 with the rest of the S_i-elements because for every T_i, when input $1 \to 1$, then input 3 converges either to 1 or 3, and when $3 \to 2$, input 1 converges to 2 or 4. Using T_4 and T_5 we can neither find different elements of this kind. There are only four elements in this case.

$$S_2^{(1)} = \{(1131), (3313), (2242), (4424)\}.$$

Next, subset $S_2^{(2)} = \{(baaa), a, b = 1, \dots 4\} \subset S_2$ has elements $T_6 = (1333)$ and $T_7 = (3111)$. Composing T_{10} and T_{11} with T_3 we obtain the rest of $S_2^{(2)}$ elements.

$$T_{10} \circ T_6 = (2442)(1333) = (2444)$$

$$T_{11} \circ T_3 = (4224)(1333) = (4222)$$

Again, no other combinations of this kind can be found. For example (3222) do not belong to $S_2^{(2)}$ because we would need an element in which 1 moves to 3 and 2 moves to 2 and we have not found any convergence element fulfilling this property.

$$S_2^{(2)} = \{(1333), (3111), (2444), (4222)\}.$$

Subset $S_2^{(3)} = \{(aabb), a, b = 1 \dots 4\}$ has elements $T_8 = (2244)$ and $T_9 = (4422)$. Composing T_{23} and T_{24} with T_8 we obtain the rest of $S_2^{(3)}$ elements.

$$T_{23} \circ T_8 = (4123)(2244) = (1133)$$

$$T_{24} \circ T_8 = (2341)(2244) = (3311)$$

Since there is no element where 2 converges to 3 and 4 converges to 2, combinations like (3322) can not be found. From this, S_2^3-set is $S_2^{(3)} = \{(2244), (4422), (1133), (3311)\}$.

Subset $S_2^{(4)} = \{(abba), a, b = 1, \dots 4\}$ has elements $T_{10} = (2442)$ and $T_{11} = (4224)$. Composing 25 and 26 with T_{10} we obtain the rest of $S_2^{(4)}$.

$$T_{23} \circ T_{10} = (4123)(2442) = (1331)$$

$$T_{24} \circ T_{10} = (2341)(2442) = (3113)$$

This subset is then, $S_2^{(4)} = \{(2442), (4224), (1331), (3113)\}$.

At last but not least, subset $S_2^{(5)} = \{(abab), a, b = 1 \dots 4\}$ has elements $T_{12} = (3232)$ and $T_{13} = (3434)$. From the composition study we have found ten more elements of this kind.

$$\begin{array}{ll} T_3 \circ T_{12} = (1131)(3232) = (3131) & T_4 \circ T_{12} = (3313)(3232) = (1313) \\ T_5 \circ T_{12} = (2242)(3232) = (4242) & (4424) \circ T_{12} = (4424)(3232) = (2424) \\ T_{14} \circ T_{12} = (1214)(3232) = (1212) & (4121) \circ T_{12} = (4121)(3232) = (2121) \\ T_{15} \circ T_{12} = (1412)(3232) = (1414) & (2143) \circ T_{12} = (2143)(3232) = (4141) \\ T_{24} \circ T_{12} = (2341)(3232) = (4343) & (4321) \circ T_{12} = (4321)(3232) = (2323) \end{array}$$

So,

$$S_2^{(5)} = \{(3232), (2323), (3131), (1313), (4242), (2424), (1212), (2121), (1414), (4141), (3434), (4343)\}.$$

Let us note that in this case we have used some new elements found before and some new elements which will be found in the S_4 and S_3 sets.

Moreover, from the composition study we have found another two subsets of S_2, $S_2^{(6)} = \{(abaa), a, b = 1 \dots 4\} = \{(1311), (3133), (2422), (4244)\}$ and $S_2^{(7)} = \{(aaab), a, b = 1 \dots 4\} = \{(1113), (3331), (2224), (4442)\}$.

$$\begin{array}{ll} (1131)(2341) = (1311) & (1131)(4123) = (1113) \\ (3313)(1311) = (3133) & (3313)(4123) = (3331) \\ (2242)(3133) = (4244) & (2242)(3331) = (4442) \\ (2242)(2341) = (2422) & (2242)(4123) = (2224) \end{array}$$

Let us study now the case where elements convergence to three different outputs. In S_3 there are two subsets of the form $S_3^{(1)} = \{(abac), a, b, c = 1, \dots 4\} = \{(1214), (1412), (3234), (3432)\}$ and $S_3^{(1)} = \{(abcb), a, b, c = 1, \dots 4\} = \{(1232), (4121)\}$. Again the composition study let us find every convergence element belonging to $S_3^{(1)}$ and $S_3^{(2)}$.

$$\begin{array}{ll} (4123)(1214) = (4143) & (4321)(1214) = (4341) \\ (2143)(1214) = (2123) & (2341)(1214) = (2321) \\ (1214)(2341) = (2141) & (3412)(4121) = (2343) \\ (4321)(4121) = (1434) & (3214)(4121) = (4323) \\ (4123)(4121) = (3414) & (2143)(4121) = (3212) \\ (4321)(4121) = (1434) & (3214)(4121) = (4323) \end{array}$$

In the remaining of the section we shall focus on the only but bijective input-output relations summarized in Table 4.10. Let us rename them 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 $p_3 = p_7 \circ p_2$, $p_8 = p_7 \circ p_5$.

p_1	(1234)	p_2	(1432)	p_3	(2143)	p_4	(2341)
p_5	(3214)	p_6	(3412)	p_7	(4123)	p_8	(4321)

Table 4.10: 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 might expect 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 4.10.

For example, the composition of p_2 and p_3 is:

$$p_3 \circ p_2 = (2143)(1432) = (2341) = p_4$$

$$p_2 \circ p_3 = (1432)(2143) = (4123) = p_7$$

The result of all the composition templates represented by product permutations is shown in Table 4.11.

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

We have found a special subset of group S_4 that fulfill the group properties, this is a subgroup. With these results we set a *Convergence Lemma*.

Lemma 1. Let us consider a two neuron CNN defined by equations (2.1) where parameters fulfill $s > 1$ and $p_+ = p_- = p$. Let us name $\mathcal{S} = \{(\pm 1, \pm 1)\}$ 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 4.10.

At this point, all possible input-output relations which can be performed using a two neuron CNN has been found. The results obtained let us scrutinize every two neuron CNN design problem. In Appendix B, we summarize all different elements obtained with their correspondent parameter range.

The particular geometry of the convergence map, limits the different convergence sets yet also gives us the key points in order to design templates, and drive us into the template composition problem. The B-transformation relating the external inputs u_i and u_i''s has been crucial in this study.

Also we have reduced the number of parameters needed to perform an input-output relation. Initially, the two neuron CNN system depends on 6 parameters defining a cloning template (I, b_0, b_+, b_-, p, s). One of the consequences of our study is that choosing only 3 of them (b_0, p, s) we are able to

design a template performing a specific input-output association. This result allows us to simplify the template design problem giving an easy way to find a template. From this analysis, we have established which combinations are possible and which are not.

To design a template, we propose the following procedure. Given a particular classification problem between four elements T, it can be written as $T = \{(abcd), a, b, c, d \in \mathbb{Z}/(4)\}$. Using this notation, it's easy to know if a concrete problem can or can not be solved looking for ϕ in Table B.11 where all elements are described. If it belongs to those which can be solved, we find the CNN parameters $\tau = (I, b_0, b_+, b_-, p, s)$ following the recipe:

1. If T belongs to the elements obtained by single templates,

   - choose p sign,
   - parameter I is already determined,
   - choose parameter b_0 (parameters b_+ and b_- are then determined),
   - choose parameter p,
   - choose parameter $s > 1$.

2. If T belongs to those obtained composing two single templates,

   - choose an element composition $T_i \circ T_j$(Table B.13),
   - find template parameters describing the single templates τ_i and τ_j, as we have done before.

Moreover, these results answer the question about the number of necessary templates needed to solve a desired problem T. From the composition study made between $T_i \circ T_j$, $i, j = 1, \ldots 25$ (Table B.13), we have seen that one or two templates are enough in order to solve T. It is due to the fact that S_1, S_2, S_3 or S_4 have a self-contained structure, and so any combination between S_i elements belongs to $< S_1, S_2, S_3, S_4 >$.

The convergence sets obtained has been classified into those converging to one, two, three or four different outputs. Not every possible combination between four elements can be achieved. Only those compatible with the convergence map. Let us note that any input-output relation of S_i, $i = 1, 2, 3, 4$, can be obtained from the composition between different elements. Nevertheless, a priori any combination in order to obtain such an element is valid.

4.5 Realizing Boolean functions

Working with values ± 1, one may think on using the results obtained in the previous section to implement Boolean functions. Taking as function domain a special subset of the external inputs space $\mathcal{S} = \{(u_0, u_1) \in \mathbb{R}^2 | u_i = \pm 1\}$, we can define Boolean functions,

$$F_i : \mathcal{S} \subset \mathbb{R}^2 \to \mathbb{R}$$

which assign to each input (u_0, u_1) a particular component of the final output like for example $F_0(u_0, u_1) = y_0(\infty)$ or $F_1(u_0, u_1) = y_1(\infty)$.

At this point, one may ask if a two neuron CNN can realize any of the 2^4 Boolean functions of this kind or only a limited number of them. Moreover, we study if any Boolean function is realized using a single template or composing different templates. For example, using the Boolean function $F_0(u_0, u_1) = y_0(\infty)$ and the correspondence $(-1, 1)$ to $(0, 1)$, element $T_1 = (1111)$ reproduce the Boolean function 1111, and element $T_{16} = (3432)$ reproduce 0001 as can be seen in Table 4.12.

(u_0, u_1)	T_{16}	(y_0, y_1)	$y_0(\infty)$	Boolean function
$(+1, +1)$	3	$(-1, -1)$	-1	0
$(+1, -1)$	4	$(-1, +1)$	-1	0
$(-1, -1)$	3	$(-1, -1)$	-1	0
$(-1, +1)$	2	$(+1, -1)$	$+1$	1

Table 4.12: Truth table corresponding to element $T_{16} = (3432)$ for Boolean function F_0.

In a similar way we find which element allows us to reproduce each of the 16 Boolean functions as can be seen in Table 4.13.

From the results obtained, any Boolean function can be realized using one single template T_i but the XOR one, where element composition is needed. We may think this result as a universal property of the two neuron CNN nevertheless, defining the vectorial Boolean function

$$F : D \subset \mathbb{R}^2 \to \mathcal{S} \subset \mathbb{R}^2$$

The 64 elements obtained in the input-output relation study, limit the different Boolean functions which can be performed by a two neuron CNN. Hence, the universal property is missing in this particular case.

element T_i	Boolean function	element T_i	Boolean function
$T_1 = (1111)$	1111	$T_2 = (3333)$	0000
$T_{14} = (1214)$	1110	$T_{16} = (3432)$	0001
$T_3 = (1131)$	1101	$T_4 = (3313)$	0010
$T_8 = (2244)$	1100	$T_9 = (4422)$	0011
$T_{15} = (1412)$	1011	$T_{17} = (3234)$	0100
$T_{44} = (1313)$	1010	$T_{12} = (3232)$	0101
$T_{20} = (1432)$	1001	$T_{11} = (4224)$	0110
$T_6 = (1333)$	1000	$T_7 = (3111)$	0111

Table 4.13: Boolean function F_0 reproduced by T_i elements using a single template except one of them, the XOR using a template composition, $T_{44} = T_4 \circ T_{12} = (1313)$.

4.6 The header of a universal Turing machine

Another example where we can see the results obtained in the classification section, is trying to implement the header action of the Minsky's 7-state 4-color universal Turing machine illustrated in Figure 4.8. One may think that it can be possible using a two neuron CNN because as shown in [8], a programmable 3×3 CNN is universal in the Turing sense because the Game of Life algorithm can be implemented on it.

A Turing machine [64] consists of a tape that can be moved back and forth, a head that possesses a state and can change the property known as color of the active cell below it, and a set of instructions which let the head modify the active cell and move the tape. A Turing machine is universal in the sense that, by appropriate programming using a finite length of input tape, it can act as any Turing machine whatsoever. In 1962 Marvin Minsky discovered a 7-state 4-color universal Turing machine illustrated in Figure 4.8.

To represent the header action on the tape, we use a symmetric two neuron CNN for $s > 1$, with their 4-possible output values coding the four colors. The input color will be coded on u_i's while the output color will be obtained from the final state y_i's of the neurons (Figure 4.9).

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 symbols. To design these templates, we shall use the convergence map relating the CNN parameters with the final outputs, the template composition study, and the results summarized in Tables 4.2 and 4.3.

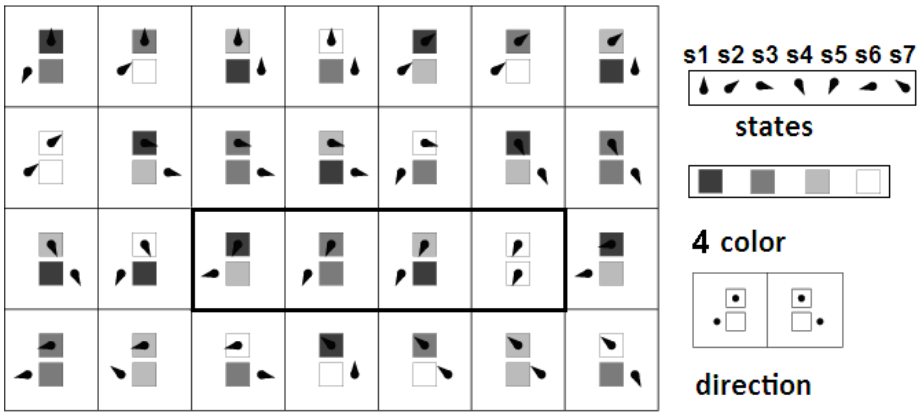


Figure 4.8: Generalization of Minsky's 7-state 4-color universal Turing machine made by Macura [64].

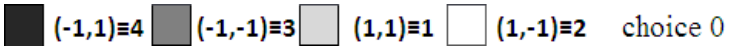


Figure 4.9: One possible correspondence between Turing machine colors and 2-neuron CNN four states.

Let us note that there are 4! different options for this choice. The consequence of this selection could be the number of templates needed in order to achieve some desired input-output association, or the difficulties on the template design. A priori any selection can be done, so we choose for example the association between the colors and the four points shown in Figure 4.9.

From this, the seven states defining the universal Turing machine can be written as elements like $\{(abcd), a, b, c, d = 1 \dots 4\}$ (Table 4.14).

state s_1	(2141)
state s_2	(2243)
state s_3	(1143)
state s_4	(1443)
state s_5	(1243)
state s_6	(1133)
state s_7	(2132)

Table 4.14: Universal Turing machine written as elements $(abcd)$.

Moreover, state $s_6 = (1133)$ belongs to S_2 and has been obtained from the composition of $T_8 = (2244)$ and $T_{23} = (4123)$. So for example, if we want to design a template reproducing s_6, we look for parameter conditions

corresponding to T_8 and T_{23} (Table B.12). Element T_8 has been found in Case 6 of Appendix B, for parameters fulfilling $p < 0$, $I = 0$, $b_+ = 1 - b_0$, $b_- = -1 - b_0$, $b_0 > 2$ and $p < \min\{\pm(1-2b_0), 1+2b_0\}$. Choosing for example $b_0 = 3$, $p = -6$ and $s = 2$ we find τ_8 template

$$\tau_8 = (I, b_0, b_+, b_-, p, s) = (0, 3, -2, -4, -6, 2)$$

Element T_{23} has been found in Case 4 of Appendix B, for parameters $p > 0$, $I = 0$, $b_+ = -1 - b_0$, $b_- = 1 - b_0$, and $p > \max\{-1 - 2b_0, -1 + 2b_0\}$. So for example we can take $b_0 = 1$, $p = 2$ and $s = 2$ obtaining τ_{23} template.

$$\tau_{23} = (I, b_0, b_+, b_-, p, s) = (0, 1, -2, 0, 2, 2)$$

From these parameter values, we can show now an example showing how to define the convergence map. For template τ_8, parameter p is negative. Using Table 4.2, we find the lines equations and the intersection points which let us print the correspondent convergence map. Inclines of the boundary lines are $m_0 = -b_-/b_0 = 4/3$, $m_1 = -b_0/b_+ = 3/2$ and $m_1\frac{1+m_0}{1+m_1} = 7/5$. Intersection points are $(x_p, y_p) = (-30, -42)$, $(x_{-p}, y_{-p}) = (30, 42)$. Boundary lines of convergence regions are then,

$$\begin{array}{ll} u_1 - 42 = 4/3(u_0 - 30), & u_1 - 42 = 3/2(u_0 - 30), \\ u_1 + 42 = 4/3(u_0 + 30), & u_1 + 42 = 4/3(u_0 + 30). \end{array}$$

For template τ_{23}, parameter p is positive. From Table 4.3 we find the lines equations and the intersection points. Inclines of the boundary lines are $m_0 = -b_-/b_0 = 0$, $m_1 = -b_0/b_+ = 1/2$ and $m_1\frac{1-m_0}{1-m_1} = 1$. Intersection points are $(x_{-p}, y_p) = (2, 2)$, $(x_p, y_{-p}) = (-2, -2)$. Boundary lines of convergence regions are then,

$$\begin{array}{cc} u_1 + 2 = 0, & u_1 - 2 = 0, \\ u_1 + 2 = 1/2(u_0 + 2), & u_1 - 2 = 1/2(u_0 - 2). \end{array}$$

Printing the correspondent convergence maps, we see that using template τ_8, which can be written as $T_8 = (2244)$, input points 1 and 2 converge to $(1, -1)$ and the rest converge to $(-1, 1)$. Using these output values as new inputs, we apply a second template τ_{23}. From Figure 4.10 it is clear that $(1, -1) = 2$ converges to $(1, 1)$, and $(-1, 1) = 4$ converges to $(-1, -1)$. We have found then a template composition performing $s_6 = (1133)$.

Nevertheless, it is clear that only state $s_1 \in S_3$ and state $s_6 \in S_2$ can be reproduced by a two neuron CNN. The rest do not belong to any set S_i described in the previous section. Of course, these results seems to depend on

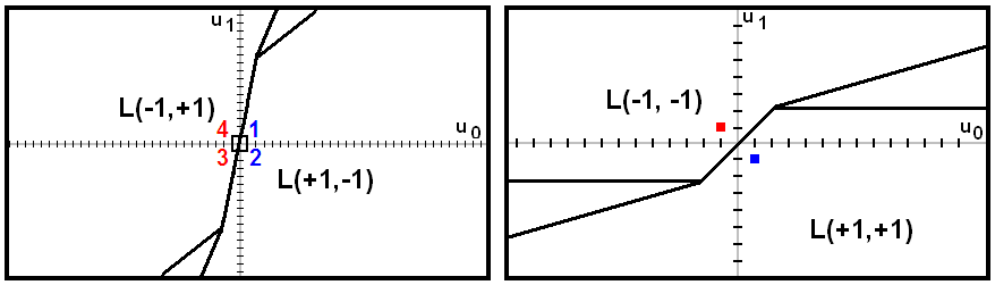


Figure 4.10: Convergence maps geometry for elements T_8 and T_{23} respectively.

the choice between colors and $\mathcal{S}$-points. So one may think on choosing a different relation between $\mathcal{S}$-points and the Turing machine colors to implement the states.

To see it, the key point is state s_5 because it is the unique state belonging to permutations subgroup. We may try then different associations fulfilling s_5. We shall see in all the cases that they do not fulfill some other state s_i, $i \neq 5$, $i = 1, \ldots, 7$, where the system converges to three different outputs. We can conclude that a 2-neuron CNN can not be used to reproduce this header action of the universal Turing machine.

For instance, another possible choice between colors and input points can be for example those shown in Figure 4.11.

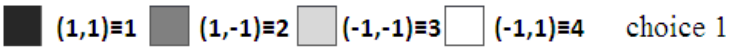


Figure 4.11: Another correspondence between the Turing machine colors and $\mathcal{S}$-points.

From this selection, seven states of the universal Turing machine are shown in Table 4.15. In this case, permutation state s_5 is fulfilled by a two neuron CNN because it can be performed by p_5 while for example other states like s_4 or s_7 are not included in the elements set S_3. Other possible cases are summarized in Appendix B.

4.7 Conclusions

In this chapter we have done a deep analysis about the different problems that can be solved using a two neuron CNN when it converges to a fixed-point. From the stability results, we have basically used the existence of the

state s_1	(2412)
state s_2	(3414)
state s_3	(3212)
state s_4	(3211)
state s_5	(3214)
state s_6	(3232)
state s_7	(4432)

Table 4.15: Universal Turing machine written as elements ($abcd$) using a new association between colors and $\mathcal{S}$-points.

Lyapunov function limiting the template parameter range for p_+, p_- and s. Basic probability distributions like the Bernoulli ones, can be reproduced by a two neuron CNN based on the results obtained from the stability study. The linear equations of the CNN system while working inside the unit square allows us to reproduce linear functions relating the external inputs with the final outputs.

On the other hand, the Lyapunov function has been the key point to construct a convergence map. The importance of this map can be clearly seen in the template design problem. In this particular two neuron CNN system, the common algorithms to learn their parameters like for example gradient descend methods exhibit little success. Nevertheless, the convergence map has given us a way to find a simple recipe to design templates. It also allows us to create template libraries solving those problems which perform different input-output associations compatible with the CNN system. Moreover, it shed light on the template composition problem. The composition of two templates $T_i \circ T_j$ has been done running the CNN system using the first template T_j. This drive the system to one of the four points $\vec{y} \in \mathcal{S}$. Using these points as external inputs $\vec{u} = \vec{y}$ and finding the correspondent convergence map for the second template T_j, we can design a new template mapping each one of these new inputs to some desired output.

One of the main uses of very many models of Artificial Neural Networks is to solve classification problems, which immediately rises the discussion about which kind of problems can be solved with this particular type. Using the different final outputs where the CNN system converges, we are able to classify different inputs (u_0, u_1) when they belong to $\mathcal{S}$. We have seen that a two neuron CNN can solve more than linearly separable problems but of course there are some limits. In order to see and understand which are these possible CNN limits, we have done a complete study of the different classification problems that can be solved using a two neuron CNN. From

this study, we have obtained a relation between the CNN parameters and the output values which is more than a learning process since we may design the system parameters by fixing the problem specifications. Of course, we do not learn exactly the parameter values but we obtain enough information to place them into a region of the plane. Hence, we are able to design a CNN performing an input-output relation. Also, we have found that not all combinations are possible in a classification problem. This fact answer the question if the CNN system can converge where we want composing different templates formulated in the convergence map section. Clearly the answer is no. It can only solve those problems compatible with the convergence map.

One important consequence of this study is that a 2-neuron CNN can not perform any logical bijective function but only a subset of them. We have seen as an example, the problem to reproduce the header action of the universal Turing machine. Only some of the input-output functional relations between the four colors of the Turing machine can be performed using a two neuron CNN. We have seen that there is no dependence on the different possible choices between the colors and the $\mathcal{S}$ elements. In all the cases, we obtain similar results.

The two neuron CNN is also capable to realize any Boolean function defined by $F(u_0, u_1) = y_i(\infty)$, where $y_i(\infty)$ is one of the two final output values where the system converges at the end of the process. The results obtained in the classification problems are compatible with this application, and gives us a novel approach for designing, in an easy way, CNN templates. Like in [20], the two neuron CNN has the universal property in the sense that every 2^{2^2} local Boolean functions of two input variables can be realized.

Nevertheless, thinking each classification problem performing an input-output relation as a vectorial Boolean function $F(u_0, u_1) = (y_0(\infty), y_1(\infty))$ we have found that not all of them can be realized by a two neuron CNN.

This leads to the discussion about the universality of Cellular Neural Networks. From [8] we know that a CNN is a universal Turing machine in higher dimensions. We may then ask which should be the minimal number of neurons needed in a CNN to be a universal Turing machine.

Part III

Convergence to a limit cycle

Chapter 5

Limit Cycles: Antisymmetric Case

Cellular neural networks are a complex dynamical system. As such they exhibit all forms of stable dynamical behavior: they may converge to a fixed-point or a limit cycle or evolve along a chaotic trajectory. In the autonomous two neuron CNN case, there are no chaotic trajectories yet the possibility of having limit cycles still remains. However, an exhaustive study relating the CNN parameter values to the dynamical behavior does not exist, and no sufficient conditions for a general case are defined nor a systematic classification of the dynamical behavior is performed.

In this chapter we focus on the regions in parameter space related to limit cycles for the two neuron autonomous continuous time CNN. As we have seen in the stability analysis, there can be such a curves in a non-symmetric parameter range. We start from a particular example with an anti-symmetric template, where a limit cycle exists, in order to develop the main concepts for a systematic method to determine sufficient existence conditions. Then we generalize the results obtained to the antisymmetric case. The key elements are a combination of the background ideas of the Poincaré-Bendixon theorem, which cannot be applied as is, with some properties of the index theory, and the geometry of the CNN equilibrium points positions.

We explain the main tools to find sufficient conditions for the existence of limit cycles and apply them on a classical example from [12] before tackle the anti-symmetric case. There are many cases to be treated in the antisymmetric CNN so we focus our explanation only in some significant ones, and explain the rest on Appendix D.

5.1 Geometry on the CNN.

The geometric representation of the trajectories in the phase plane, is an invaluable tool in studying dynamical systems. This reveals information such as whether a stable point, a repulsive point, or a limit cycle is present in the dynamic behavior for a chosen parameter values.

In general, a linear system $\dot{x} = F(x), x \in \mathbb{R}^n$ can be written in matrix form as $\dot{x} = Ax$, $A \in \mathbb{R}^{n\times n}$. In dimension two, points x^* where $F(x^*) = 0$, named equilibrium points, and the eigenvalues and eigenvectors of matrix A, let us print the system trajectories moving in the phase plane. For the CNN case, it can help us to understand the relation between the parameters of the cloning template and the final trajectory solutions. Hence, it let us find limit cycles.

Two neuron network general equations studied on (1.1) is a piecewise linear system which can be solved as a linear system in nine different regions as we have seen in Chapter 2. In this chapter, we fix $u'_0 = u'_1 = 0$ in order to simplify the study. Let us remember R_i, $i \in \{0, \ldots, 8\}$ the nine regions of the plane limited by lines $x_i = \pm 1$ for $i = 0, 1$ where the CNN system is linear (Figure 2.2). We use A_i as the notation for the system CNN matrices corresponding to each one of the nine regions R_i for $i = 0, 1, \ldots, 8$.

Table 5.1: Equilibrium points in each of the nine regions where the system is linear in the general case.

x_i^*	m_i	k_i
R_0	0	0
R_1	$-\frac{p_+}{s-1}$	$s - \frac{p_+p_-}{s-1}$
R_2	$s - \frac{p_+p_-}{s-1}$	$-\frac{p_-}{s-1}$
R_3	$\frac{p_+}{s-1}$	$-s + \frac{p_+p_-}{s-1}$
R_4	$-s + \frac{p_+p_-}{s-1}$	$\frac{p_-}{s-1}$
R_5	$+p_+ - s$	$-p_- + s$
R_6	$p_+ + s$	$p_- + s$
R_7	$-p_+ + s$	$p_- - s$
R_8	$-p_+ - s$	$-p_- - s$

Parameter range we are going to work is defined by $s > 1$ and $p_+p_- < 0$, where it's known that the system can converge to a limit cycle. For this parameter range, equilibrium point x_0^* of the linear system in region R_0, is a spiral source because the CNN matrix A_0 (2.13) has a complex pair of

eigenvalues $\lambda_i = s - 1 \pm \sqrt{p_+ p_-}$ with positive real part. Equilibrium points $x_i^*, i = 1, \ldots 4$ are saddle points because eigenvalues $\lambda_0 = s - 1$ and $\lambda_1 = -1$ have different sign. The rest, are stable nodes with the special case of having equal eigenvalues $\lambda_0 = \lambda_1 = -1$ (Figure 5.1).

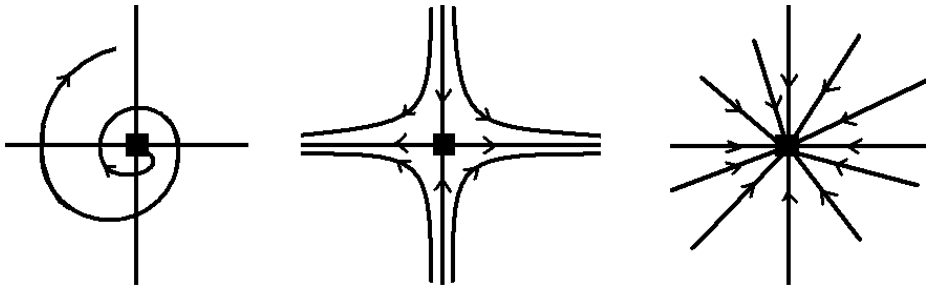


Figure 5.1: Equilibrium types: spiral source, saddle point and stable node.

The actual combination of equilibrium points $x_i^* = (m_i, k_i), i = 0, \ldots 8$ (Table 5.1) position and principal directions, defined by the eigenvectors of the CNN matrix, determines the overall system dynamical behavior. In particular, the existence of limit cycles. Since these positions depend on the cloning template, we find here the way to set the relation between the system parameters and the existence of limit cycles. Let us note that we can have a large number of combinations of equilibrium point positions.

5.2 Limit cycles

Limit cycles are inherently nonlinear phenomena: they cannot occur in linear systems. Therefore they cannot be found in a single region R_i. Of course a linear system can have closed orbits, but they won't be isolated. A limit cycle is an isolated closed trajectory of the system. This means that neighboring trajectories are not closed, they spiral either towards or away from the limit cycle.

5.2.1 Searching limit cycles

Poincaré-Bendixon theorem establishes conditions for the existence of closed orbits and it is a common tool to prove limit cycles existence in system dynamics.

Theorem 8 (Poincaré-Bendixon theorem)**.** Suppose that:

(a) D is a closed, bounded subset of the plane;

(b) $\dot{x} = F(x)$ is a continuously differentiable vector field on an open set containing D;

(c) D does not contain any equilibrium points; and

(d) There exists a trajectory C that is confined in D, in the sense that it starts in D and stays in D for all future time.

Then either C is a closed orbit, or it spirals towards a closed orbit as $t \to \infty$. In either case, D contains a closed orbit.

Obviously (b)-condition is not satisfied by our system because of the piecewise linear function. Therefore, Poincaré-Bendixon theorem can not be applied, but it shall be of help to find limit cycles: we shall keep the idea of finding a connect, closed set in the plane with no equilibrium points nor fixed-points inside, where trajectories $(x_0(t), x_1(t))$ remain for t$\to \infty$. If so, in that region we shall not have simple closed orbits nor chaotic ones, for there is no chaos in two dimensions, and this such trajectories may be limit cycles.

Equilibrium points of the nine linear systems can be also equilibrium points for the general CNN system (2.1) if they are located inside their corresponding region. Stable nodes are attractive points so, they can become fixed-points for the general system.

To guarantee no equilibrium points are present inside region D, we study the equilibrium points position on the plane. We must also take care of the boundaries of regions $R_i, i = 0, \ldots 8$. In effect, although in each of the nine regions system trajectories are consistent, on the boundary lines we can have fixed-points. We just need trajectories from neighboring regions having opposite directions which converge from each side onto the border line, and the system will have a fixed-point.

Therefore, we aim at finding a closed, connected and bounded region D, containing no fixed-points nor equilibrium points. It will be limited by two closed, non-overlapping boundary curves C_1 and C_2, as can be seen in Figure 5.2. These curves will fulfill the following conditions:

- Inside C_1, there must exist an unstable equilibrium point so that any trajectory crossing C_1 is driven into region D.
- Any trajectory crossing C_2 is driven into region D or, at least, is tangent to C_2, but never leaves region D.

If these conditions are fulfilled, it becomes clear that any trajectory entering D shall stay bounded in D. Since there are no fixed-point by construction, we should find a closed orbit in D.

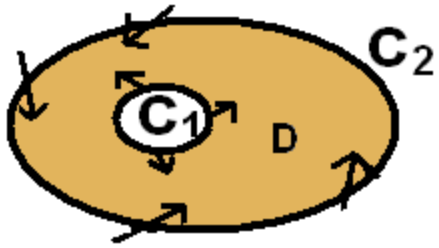


Figure 5.2: Region D where a limit cycle can exist.

To assure that there are no fixed-points in the boundaries of the different regions $R_i, i = 0, 1, \ldots 8$ and inside region D, we need to define first the index of a curve.

5.2.2 Boundary problem

The index of a closed curve C is an integer that measures the winding of the vector field defining the differential equation on C [57]. If we represent at each point on C, the value of the vector field as we move counterclockwise around C, vector rotates an angle $2I_C\pi$. We call I_C the index of C.

Therefore, the index of a closed curve containing no fixed-points can be calculated by integrating the change in the angle of the vectors at each point in C around C,

$$I_C = \frac{1}{2\pi}\oint_C d\phi = \frac{1}{2\pi}\oint_C d\left(\tan^{-1}\frac{\dot{x_1}}{\dot{x_0}}\right)$$

where ϕ is the angle made by the vector field with the horizontal axis. I_C is the net number of counterclockwise revolutions made by the vector field as x moves one around C. To compute the index we only need to know the vector field along C.

Using index properties of a curve we can deduce that fixed-points must not be on the boundaries of the different regions $R_i, i = 0, \ldots, 8$.

First, the index of a closed trajectory of the system C *i.e.* a closed orbit, is $I_C = +1$ simply because vector field is tangent to the trajectory and so it can only make one full turn. Now, the index of a closed curve C' that can be continuously deformed into C without passing through a fixed-point must have the same index as C. Since the index of a limit cycle must be $+1$ and C_1 and C_2 must be closed curves around C that can be continuously deformed into C they must have index $+1$,

$$I_{C_1} = I_{C_2} = I_C = +1. \tag{5.1}$$

The index of an isolated equilibrium point x^* is defined as I_C where C (not necessarily a system solution) is any closed curve that encloses x^* and no other equilibrium points. From this, we see that the index of a saddle point is -1 while for any other type of equilibrium point or for a periodic orbit is $+1$.

If C is a closed trajectory for the system, then it must enclose equilibrium points whose indices sum to $+1$ and $I_C = +1$. Furthermore, if a closed curve C surrounds n isolated equilibrium points x_i^* for $i = 1, \ldots, n$, then $I_C = \sum_{i=1}^{n} I_{x_i^*}$.

Finally, let us note that if we reverse the sense of the vector field , the index is unchanged. This last property will be useful if we need to reverse all trajectory senses. These index properties set the conditions to define C_1 and C_2 correctly with respect to the system equilibrium points.

5.2.3 Towards limit cycles

We shall now tackle the existence of limit cycles using the properties mentioned above. The index of a limit cycle C, if it exists, must be $I_C = 1$ just as the indices of C_1 and C_2, $I_{C_1} = I_{C_2} = +1$ because they are closed curves surrounding C which can be continuously deformed into C without passing through a fixed-point (5.1). Therefore, there cannot be equilibrium points in the region D limited by C_1 and C_2, as required.

Taking C_1 around $x_0^* = (0, 0)$ (Table 5.1), we obtain that x_0^* must be a repulsive point with index $+1$. For parameters $s > 1$ and $p_+ p_- < 0$ this equilibrium point is a spiral source.[1]

On one hand, this choice ensures that trajectories crossing C_1 go into region D and $I_{C_1} = +1$, which are conditions required to C_1. On the other hand, trajectories crossing C_2 must either go into D or remain tangent to the boundary of region D, but never leave it. So C_2 cannot be inside region R_0, (condition needed on C_2). In short, C_2 has to be a curve surrounding C_1 and x_0^*, and no other equilibrium points, so its index must be $+1$.

The other equilibrium points are already determined. In regions R_i for $i = 5, 6, 7, 8$, x_i^* are stable nodes. Therefore, they may not be inside region D limited by C_1 and C_2: they must be outside. In regions R_i for $i = 1, 2, 3, 4$, x_i^* are saddle points and they cannot be in D either. This region will be bounded by x_0^*, and the remaining equilibrium points.

To define the geometry of curve C_2 we use the saddle point positions and their principal directions defined by

[1] We are not considering here degenerate cases where $p_+ = 0$ or $p_- = 0$.

$$r_i^* : x_i^* + \lambda \vec{v_{ij}}; i = 1, \ldots, 4; j = 0, 1, \lambda \in \mathbb{R}$$

where $\vec{v_{ij}}$ are the eigenvectors of the corresponding CNN-matrix. To meet the requirements, curve C_2 must be such that trajectories cannot cross the principal lines. In this way, all the system flow remains in region D. Remark that principal directions in regions R_i for $i = 1, 2, 3, 4$, have certain symmetry because their director vectors are the same, $\{\vec{v_{i0}} = (s, p_-), \vec{v_{i1}} = (0, 1)\}$ in R_1, R_3 because the CNN matrices A_1 and A_3 are equal (2.14). Director vectors are also equal in R_2, R_4 where $\{\vec{v_{i0}} = (1, 0), \vec{v_{i1}} = (p_+, s)\}$.

From these results we will construct C_2 using the repulsive principal directions of the saddle points in regions R_1, R_2, R_3, R_4 composed with curves in regions R_5, R_6, R_7 and R_8. Such curves do not include, by construction, any equilibrium point of the above mentioned regions in D, and will be defined using the geometry of the problem in each case.

5.2.4 Example

In order to illustrate these results, let us study a particular example [12] where parameter values are $2 = s$, $p_+ = -p_- = 2$ and $u'_0 = u'_1 = 0$. Let us note that in this case, parameters p_+ and p_- are totally antisymmetric. Calling $p_s = \frac{p_+ + p_-}{2}$ and $p_a = \frac{p_+ - p_-}{2}$, we have that $p_s = 0$ and $p_a = 2 \neq 0$.

x_i^*	R_0	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8
m_i	0	-2	6	2	-6	0	4	0	-4
k_i	0	6	2	-6	-2	4	0	-4	0

Table 5.2: Equilibrium points for $2 = s$, $p_+ = -p_- = 2$ and $u'_0 = u'_1 = 0$

Equilibrium points positions (Table 5.2) are antisymmetric with respect x_0^* and there exist $\pi/2$ rotation centered on x_0^* called $G_{\pi/2}$ between $x_i^*, i = 1, 2, 3, 4$ and $x_j^*, j = 5, 6, 7, 8$ as can be seen in Figure 5.3. Studying x_1^* we can deduce $x_i^*, i = 2, 3, 4$ positions because $x_i^* = G_{-\pi/2}^{i-1}(x_1^*)$ and studying x_5^*, we can deduce $x_j^*, j = 6, 7, 8$ because $x_j^* = G_{-\pi/2}^{j-5}(x_5^*)$.

Hence, there is a central symmetry between R_1-trajectories and R_3-trajectories with respect the origin x_0^*. We shall use this symmetry in the limit cycle search. Using the geometry of the equilibrium points positions, we can find C_1 and C_2 continuously closed curves limiting the desired closed region D.

In R_0, we take C_1 as a circle surrounding the spiral source $x_0^* = (0, 0)$, so C_1 have index 1 and all trajectories starting inside C_1 pass across leaving

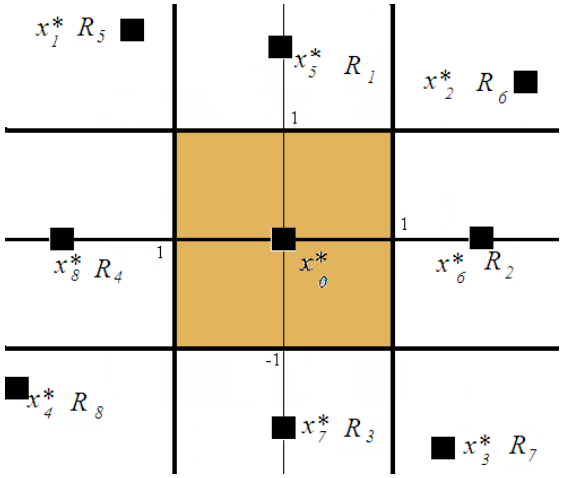


Figure 5.3: Equilibrium points positions for $s = p_+ = -p_- = 2$.

the inner region.

$$C_1 = \{(x_0, x_1) \in \mathbb{R}^2 | x_0^2 + x_1^2 = \rho^2, \rho \in (0, 1)\} \tag{5.2}$$

In effect, using polar coordinates $x_0 = \rho\cos\theta, x_1 = \rho\sin\theta$ we obtain that

$$\dot{x_0} = \dot{\rho}\cos\theta - \rho\sin\theta\dot{\theta} = (s-1)\rho\cos\theta + p_a\rho\sin\theta. \tag{5.3}$$

$$\dot{x_1} = \dot{\rho}\sin\theta + \rho\cos\theta\dot{\theta} = (s-1)\rho\sin\theta - p_a\rho\cos\theta. \tag{5.4}$$

Doing $(5.3)\cos\theta + (5.4)\sin\theta$,we obtain $\dot{\rho} = (s-1)\rho > 0$ for $s > 1$. Hence, trajectories leave C_1-inner region.

Now, to find C_2 we will use $x_1^*, \ldots, x_8^*$ and their trajectory directions. Studying in more detail each case and using the geometry of the problem, we focus our study on x_5^* and x_1^* and extend by symmetry the results. As x_5^* is an stable node out from its region, trajectories on R_5 must connect R_4-trajectories with R_1-trajectories.

C_2 can not be inside R_0, and must be a continuous closed curve with index $+1$ so, it has to be defined in regions $R_1, \ldots, R_8$ connecting curves C_{2j} from adjacent regions.

$$C_2 = \{C_{2j};\ \ if\ \ (x_0, x_1) \in R_j, j = 1, \ldots 8\}.$$

These curves should not let system solutions cross outside D. Now, in the saddle points regions, we are going to use their repulsive principal directions to define C_{2i} , $i = 1, 2, 3, 4$. For instance, in region R_1 we define

$$C_{21} = \{(x_0, x_1) = x_1^* + \lambda(2, -2), \lambda \in \mathbb{R}\}$$

where $(2, -2) = (s, p_-) = \vec{v_0}$ is an eigenvector of region R_1. In this particular case, $C_{21} = \{(x_0, x_1) \in \mathbb{R}^2 | x_1 = -x_0 + 4\}$.

Doing the phase portrait in this case, we see that trajectories leaving region R_5 reach R_1 at a point under C_{21} connecting trajectories from regions R_4 and R_1 (Figures 5.4 and 5.5).

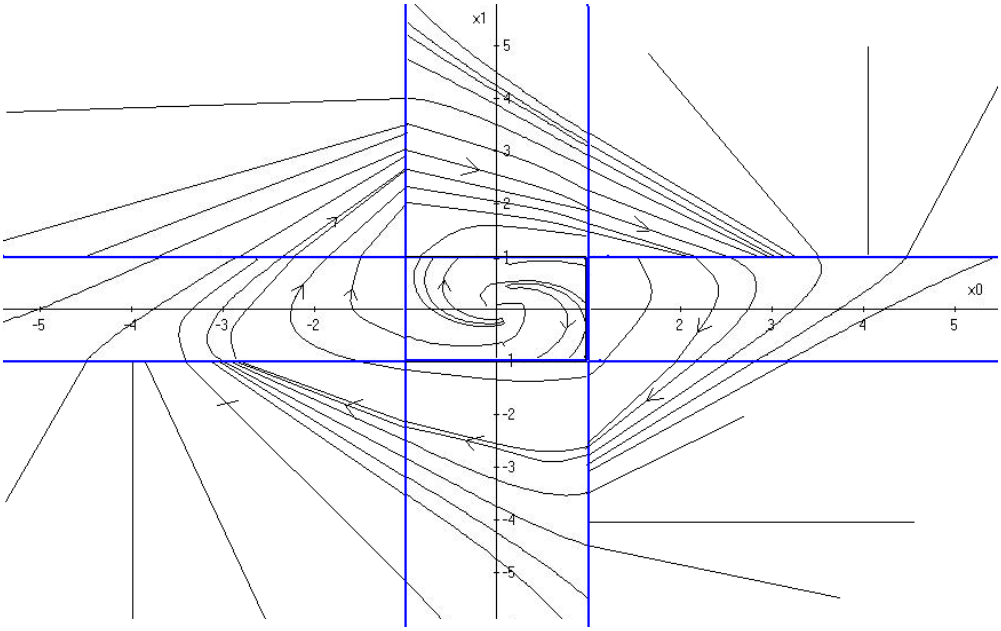


Figure 5.4: Phase portrait for $s = p_+ = -p_- = 2$ and $u_0' = u_1' = 0$.

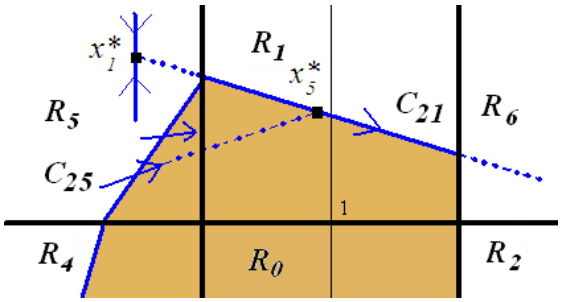


Figure 5.5: Construction of C_{21} line connecting regions R_5 and R_6.

Now in region R_5 we define C_{25} as a line connecting C_{21} with C_{24}. It is a line including the points in which C_{21} and C_{24} reach their corresponding

boundary lines. In this case, $C_{25} = \{(x_0, x_1) \in \mathbb{R}^2 | x_1 = 2x_0 + 7\}$. Trajectories do not cross this line, and they leave the inner region D because of the equilibrium point position $x_5^* \in R_1$.

From the central symmetry of this problem, we define the remaining curves C_{2i} in regions $R_2, R_3, R_4, R_6, R_7, R_8$. Let C_2 be a closed curve which trajectories cannot cross leaving D. A trajectory starting in R_0 cross C_1 leaving R_0 and will remain in region D for $t \to \infty$ (See Figure 5.7). Let us note that, in this example, equilibrium points lay all out of their corresponding region and so they are not fixed-points for the system.

These conditions are sufficient to demonstrate the existence of a limit cycle. There can not be another possible output for a trajectory remaining inside D. In effect, if there was a fixed-point $O^* \in D$ (See Figure 5.6), it would be on a border of a region $R_i, i = 1, \ldots, 8$ where two trajectories with opposite sense would meet. Its index would then be $I_{O^*} \neq 0$ and so from the index properties $I_{C_2} = I_{x_0^*} + I_{O^*} \neq 1$. This can not be possible because C_2-index sums to $+1$ (5.1). This is a general property. In a concrete example, we can also find trajectory directions on the border regions lines in order to study the existence of such a point O^* .

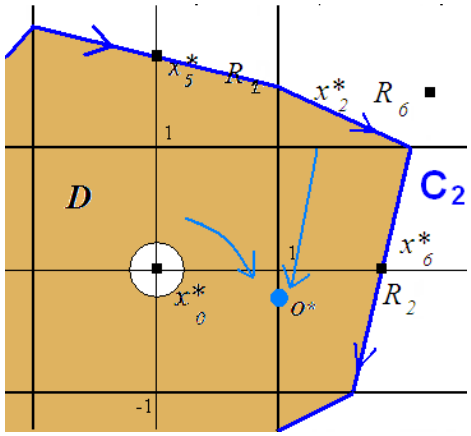


Figure 5.6: Fixed-point O^* on a boundary line of the unit square.

A limit cycle must then exists because there is no chaotic behavior in autonomous two dimensional systems. Remark that from this example we see that the fact that parameters of the cloning template are totally antisymmetric is not enough to ensure the existence of a limit cycle. Equilibrium points positions and its principal directions play also an important role on its existence. Finally, let us note that system dynamics does not depend on initial conditions. Starting inside the unit square, the system will always converge to a limit cycle around R_0.

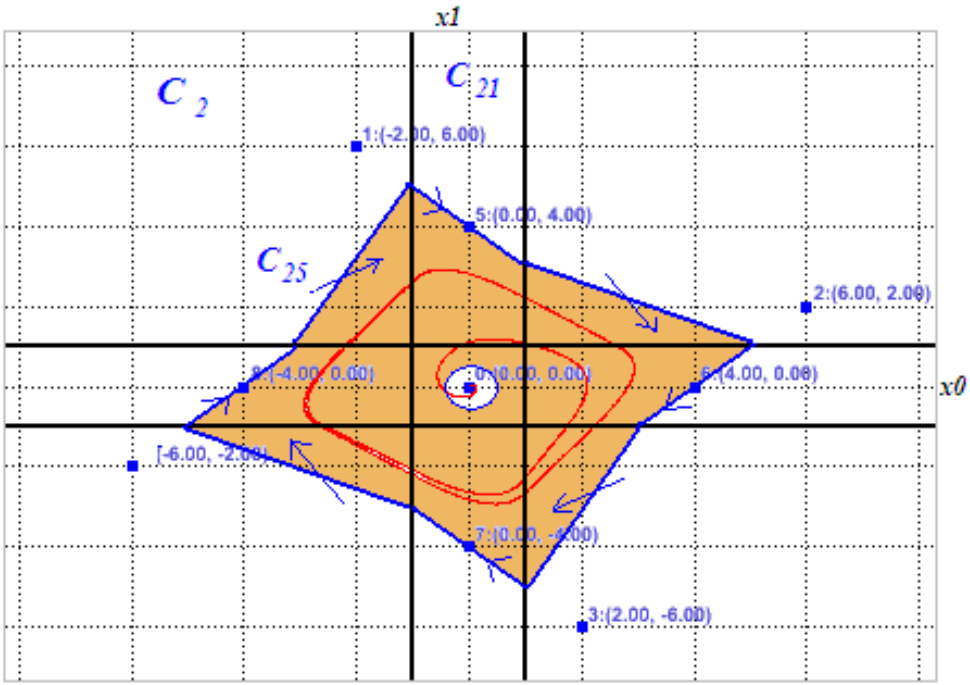


Figure 5.7: Limit cycle for $s = p_+ = -p_- = 2$ in region D limited by curves C_2 and C_1.

5.3 Antisymmetric case.

We shall now proceed to generalize the particular result to the autonomous antisymmetric case $p_s = 0, p_a = p_+ = -p_-$ and $u'_0 = u'_1 = 0$. The argument of the previous section relies only on the nature and position of the equilibrium points so we need to obtain similar geometric conditions. Fixed-point of R_0 is a spiral source and so $s > 1$ and $p_+p_- = p_s^2 - p_a^2 = -p_a^2 < 0$. From this s value, equilibrium points in R_1, R_2, R_3, R_4 are saddle points and in regions R_5, R_6, R_7, R_8 are stable nodes (Table 5.3). Let us note that equilibrium points positions are antisymmetric with respect x_0^* just like in the example so we only need to study two of them like for example x_1^* and x_5^*.

x_i^*	m_i	k_i
R_0	0	0
R_1	$-\frac{p_a}{s-1}$	$s + \frac{p_a^2}{s-1}$
R_2	$s + \frac{p_a^2}{s-1}$	$\frac{p_a}{s-1}$
R_3	$\frac{p_a}{s-1}$	$-s - \frac{p_a^2}{s-1}$
R_4	$-s - \frac{p_a^2}{s-1}$	$-\frac{p_a}{s-1}$
R_5	$p_a - s$	$p_a + s$
R_6	$p_a + s$	$s - p_a$
R_7	$s - p_a$	$-s - p_a$
R_8	$-s - p_a$	$p_a - s$

Table 5.3: Equilibrium points for $s > 1, p_s = 0$

To reproduce a similar argument in order to demonstrate the existence of a limit cycle, curve C_1 can be taken as in the example, namely, a circle centered on x_0^* with radius $\rho \in (0, 1)$ (5.2). Trajectories cross C_1 towards region D. To get the equilibrium points in the same regions as in the example studied before we need $x_1^* \in R_5$ and $x_5^* \in R_1$. For this, parameters fulfill $p_a > s - 1; s + \frac{p_a^2}{s-1} > 1; |p_a - s| < 1$ and $p_a + s > 1$.

	x_1^*	x_2^*	x_3^*	x_4^*	x_5^*	x_6^*	x_7^*	x_8^*
R_i	R_5	R_6	R_7	R_8	R_1	R_2	R_3	R_4

Studying the rest of the regions we obtain:

$$s > 1 \quad |p_a - s| < 1$$

Focusing our study on saddle points and their principal directions, we take for example the principal direction in R_1 with director vector $(s, -p_a)$. To

define C_{21} as a line passing across x_1^* with incline $\tan\beta = -\frac{p_a}{s}$ as in the example, we study necessary conditions to make the trajectories connect its neighboring regions R_5 and R_6 (Figure 5.8). Inclines of these lines $\tan\beta$ and

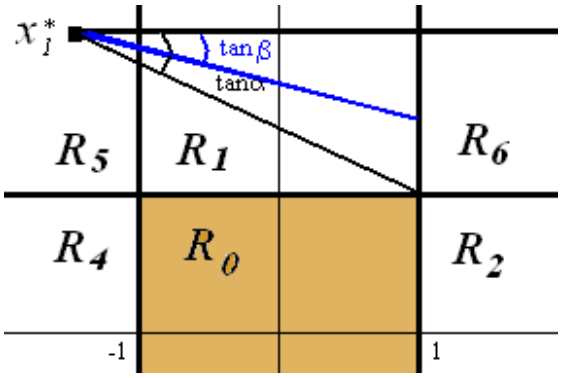


Figure 5.8: Inclines of the lines C_{21} and C_{22} in regions R_1, R_2.

$\tan\alpha$ must fulfill $\tan\alpha < \tan\beta$ and so

$$\frac{s + p_a^2/(s-1) - 1}{-p_a/(s-1)} < -\frac{p_a}{s} \Leftrightarrow \frac{(s-1)[-s(s-1) - p_a(p_a-1)]}{s[p_a + (s-1)]} < 0$$

$$\Leftrightarrow s(s-1) + p_a(p_a-1) > 0 \Leftrightarrow$$

$$(s - 1/2)^2 + (p_a - 1/2)^2 > \frac{1}{2} \tag{5.5}$$

because $s > 1$ and $p_a > 0$

In region R_2, we do a similar study where principal direction has director vector (p_a, s) and so its incline is $\tan\beta = \frac{s}{p_a}$. The incline of the line passing across x_2^* and $(-1,-1)$ is $\tan\alpha$. In this region $\alpha < \beta$ and so

$$\tan\alpha < \tan\beta \Leftrightarrow \frac{p_a/(s-1) + 1}{s + p_a^2/(s-1) - 1} < \frac{s}{p_a}$$

$$\Leftrightarrow \frac{(s-1)[-s(s-1) - p_a(p_a-1)]}{p_a[p_a^2 + (s-1)^2]} < 0 \Leftrightarrow s(s-1) + p_a(p_a-1) > 0$$

we obtain the same condition as we have found before. This condition is also valid for regions R_3 and R_4. Hence, curve C_{21} is defined as

$$C_{21} = \{(x_0, x_1) = x_1^* + \lambda(s, -p_a), \lambda \in \mathbb{R}\} = \{sx_1 + p_a x_0 = p_a^2 + s^2\}$$

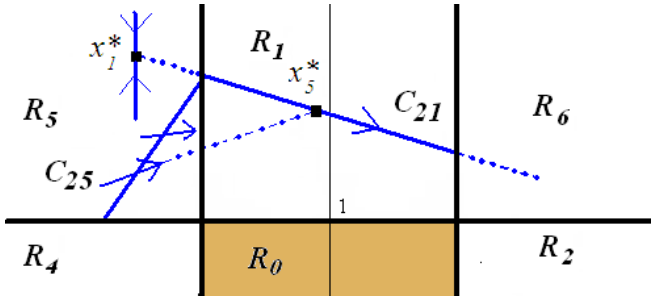


Figure 5.9: C_2 construction in regions R_1 and R_5.

Last but not least, to define C_2 curve in R_5, we study the relative position of x_5^* in region R_1 and we find that $x_5^* \in C_{21}$ because

$$sk_5 + p_a m_5 = s(p_a + s) + p_a(p_a - s) = sp_a + s^2 + p_a^2 - sp_a = p_a^2 + s^2$$

Therefore, trajectories leaving R_5 reach R_1 in points under C_{21} (Figure 5.3). Now we can define C_{25} as the line connecting C_{21} and C_{24}. Trajectories on this line will not leave the inner region D. In a same way we can define C_{26}, C_{27} and C_{28}. From this study, we have obtained sufficient conditions to ensure a limit cycle will exist in D.

Finally, by symmetry, if $p_a < -(s-1)$ a similar study can be done so that the sufficient conditions to get a limit cycle are:

$$\begin{gathered}(s-1/2)^2 + (p_a + 1/2)^2 > \tfrac{1}{2} \\ s > 1 \quad |p_a + s| < 1\end{gathered}$$

From these results we may now discuss the existence of limit cycles around the unit square depending on the equilibrium points position, and so depending on s and p_a. We have seen that for $s > 1$ and $p_s = 0$, equilibrium points in the nine regions are either spiral sources, saddle points or stable nodes. Studying their positions on the plane (Table 5.5), and doing the phase portrait in each case, we find in which cases a limit cycle exists. We have studied case named $O1$ where

$$s > 1; \;\; |p_a - s| < 1; \;\; s(s-1) + p_a(p_a - 1) > 0$$

Plotting on (p_a, s)-plane (Figure 5.10) the boundary lines determined by the different conditions obtained in each case O_i, $i = 1, \ldots 6$ and I_1, we find the CNN parameter regions in the antisymmetric case, and also we find a functional relation between CNN parameters and CNN outputs.

In Appendix D, we study the rest of these different parameter regions classified into different cases. Out cases Oi for $i = 2, 3, 4, 5, 6$, where all equilibrium points are out from their corresponding region, and In-cases $I1$ where all of them are inside their region (Tables 5.4 and 5.5).

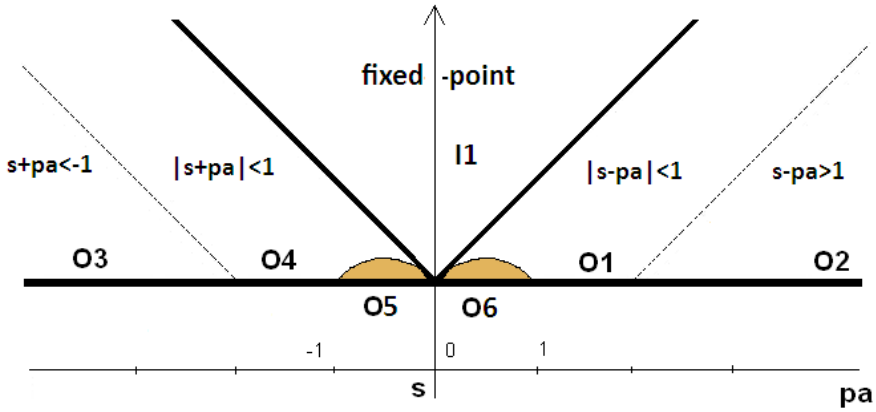


Figure 5.10: Parameter regions where a limit cycle exist. Case Oi correspond to those parameter regions where the system converge to a limit cycle. Inside regions OIi, system converges to a limit cycle across the unit square. Taking parameters inside $I1$, it converges to a fixed-point.

$O1$	$s > 1$	$\|p_a - s\| < 1; (s-1/2)^2 + (p_a - 1/2)^2 > 1/2$
$O2$	$s > 1$	$p_a > s + 1$
$O3$	$s > 1$	$p_a < -1 - s$
$O4$	$s > 1$	$\|p_a + s\| < 1; (s-1/2)^2 + (p_a - 1/2)^2 > 1/2$
$O5$	$s > 1$	$\|p_a - s\| < 1; (s-1/2)^2 + (p_a - 1/2)^2 < 1/2$
$O6$	$s > 1$	$\|p_a + s\| < 1; (s-1/2)^2 + (p_a - 1/2)^2 < 1/2$
$I1$	$s > 1$	$\|p_a\| < s - 1$

Table 5.4: Different parameter regions in the antisymmetric case.

	x_1^*	x_2^*	x_3^*	x_4^*	x_5^*	x_6^*	x_7^*	x_8^*
$O1$	R_5	R_6	R_7	R_8	R_1	R_2	R_3	R_4
$O2$	R_5	R_6	R_7	R_8	R_6	R_7	R_8	R_5
$O3$	R_6	R_7	R_8	R_5	R_4	R_1	R_2	R_3
$O4$	R_6	R_7	R_8	R_5	R_8	R_5	R_6	R_7
$O5$	R_5	R_6	R_7	R_8	R_1	R_2	R_3	R_4
$O6$	R_6	R_7	R_8	R_5	R_8	R_5	R_6	R_7
$I1$	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8

Table 5.5: Equilibrium points positions in each case.

5.4 Conclusions

In this section, we have found sufficient conditions for the existence of limit cycles in the antisymmetric case. For this, we must work in a parameter range where limit cycles has a chance. If we are able to work with local repulsive equilibrium points or with attractive local equilibrium points but placed out from their correspondent region, then the system can not converge to a fixed-point. As we work with the autonomous two neuron CNN, there is no chaotic behavior and so, the system must converge to a closed orbit.

In order to find it, we have used the idea of Poincaré-Bendixson theorem to construct two different curves C_1 and C_2 which act as boundary lines of a closed region surrounding a cyclic dynamic behavior. Firstly, we have studied an example where it's known that a limit cycle exist [12]. Then, we have extended the results to the autonomous antisymmetric case. Using the geometry of the dynamic behavior, we have used the local equilibrium points kind to construct curves limiting the possible limit cycle. Depending on the position on the plane of the different equilibrium points, we have obtained different cases named Out or In cases.

From this study, we can conclude that in the Out cases, if condition (5.5) is fulfilled there exists a limit cycle around R_0 because curves C_1 and C_2 can be found as described above. If condition (5.5) is not fulfilled, parameters are in region O5. In this case, there exists also a limit circle but it goes then through regions R_0, R_1, R_2, R_3 and R_4, because curve C_{21} (and also $C_{2j}, j = 2, 3, 4$) reaches the unit square.

However if one equilibrium point is inside its own region, then all of them are inside their corresponding regions (case $I1$) because of the symmetry between them. In this case, a limit cycle around R_0 can not exist. The system will converge to a stable node, and so the final output will be in $\mathcal{S} = (\pm 1, \pm 1)$. These different configurations depend then on parameters s and p_a for $s > 1$.

Actually, we can establish two classes of dynamic behavior for the CNN, depending on the equilibrium points positions. Figure 5.10 shows parameter range limiting this behavior. Taking parameters inside O_i zones, limit cycles exist around R_0 or crossing R_0. Inside $I1$, limit cycles do not exist, the system is completely stable and will converge to one of the equilibrium points $x_i^*, i = 5, 6, 7, 8$.

Chapter 6

Limit Cycles: General Case

In this chapter we will extend the results in the previous section to the non symmetric template case, using the same techniques. Again we will define two curves: C_1 and C_2 where on C_1, trajectories cross leaving the inner region and on C_2 trajectories never go through but, they go tangent or stay inside region D. To define these curves we will use again the system geometry depending on equilibrium points type and position on the plane.

6.1 General case

As we have seen in the stability analysis, working in the parameter range $s > 1$, $p_+p_- < 0$ the local equilibrium points (m_i, k_i) are repulsive except those of the Out regions. Inside the unit square, the origin behaves as a spiral source while in the middle regions we have saddle points. In the Out regions the equilibrium points are stable nodes.

In this general case, where parameters p_+ and p_- only has the restriction to have different sign, local equilibrium points are those shown in Table 6.1.

Using the repulsive behavior of the equilibrium point located on the origin as we have done in the antisymmetric case, curve C_1 will be defined as a circle around the origin with radius lower than 1 (5.2).

To define C_2 we use again the rest of the equilibrium points. Let us note first that there is a central symmetry between some of the equilibrium points as can be seen in Table 6.1. Hence, we only need to study four of them, x_1^*, x_2^*, x_5^* and x_6^*.

$$x_3^* = G_\pi(x_1^*); \quad x_4^* = G_\pi(x_2^*); \quad x_7^* = G_\pi(x_5^*); \quad x_8^* = G_\pi(x_6^*)$$

Next, all stable equilibrium points lay on the repulsive principal directions of the saddle points. For example, $x_5^* = (p_- - s, s - p_-)$ lay on repulsive

x_i^*	m_i	k_i
R_0	0	0
R_1	$m_1 = -\frac{p_+}{s-1}$	$k_1 = s + -\frac{p_-p_+}{s-1}$
R_2	$m_2 = s - \frac{p_-p_+}{s-1}$	$k_2 = -\frac{p_-}{s-1}$
R_3	$-m_1 = \frac{p_+}{s-1}$	$-k_1 = -s + \frac{p_-p_+}{s-1}$
R_4	$-m_2 = -s + \frac{p_-p_+}{s-1}$	$-k_2 = \frac{p_-}{s-1}$
R_5	$m_5 = p_+ - s$	$k_5 = -p_- + s$
R_6	$m_6 = p_+ + s$	$k_6 = p_- + s$
R_7	$-m_5 = -p_+ + s$	$-k_5 = p_- - s$
R_8	$-m_6 = -p_+ - s$	$-k_6 = -p_- - s$

Table 6.1: Local equilibrium points in each region R_i where parameters p_+ and p_- only has the restriction to have different sign.

principal direction r_1 of the saddle point x_1^* because it fulfill line equation $p_-x_0 = sx_1 - s^2 + p_+p_-$. In a same way, equilibrium point of region R_6, $x_6^* = (p_+ + s, p_- + s) \in r_2$ where r_2 is the repulsive principal direction on region R_2 with equation $sx_0 = p_+x_1 + s^2 - p_+p_-$. Using the symmetry of the problem, equilibrium points x_7^* and x_8^* are on its corresponding repulsive principal directions r_3 and r_4.

Finally, equilibrium points are restricted to be in some regions on the plane. For example, $x_1^* = \left(-\frac{p_s+p_a}{s-1}, s - \frac{p_s^2-p_a^2}{s-1}\right)$ can only be in regions R_5, R_1 and R_6 because parameters $s > 1$ and $p_+p_- < 0$ and so $s - \frac{p_s^2-p_a^2}{s-1} > 1$.

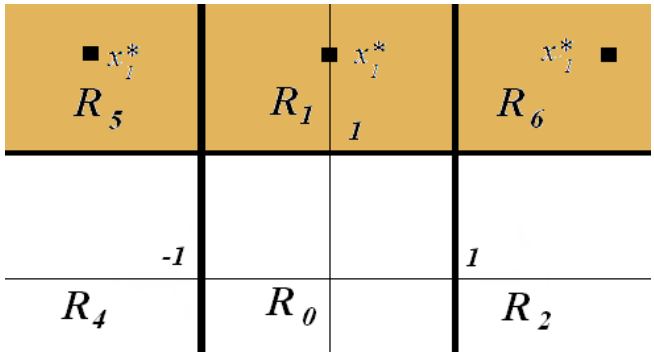


Figure 6.1: Possible positions for the equilibrium point x_1^*.

Studying all equilibrium points positions, we find all the cases summarized in Table 6.2.

For example case O1:

	x_1^*	x_2^*	x_5^*	x_6^*		x_1^*	x_2^*	x_5^*	x_6^*
O1	R_5	R_6	R_1	R_7	OI1	R_5	R_2	R_1	R_6
O2	R_5	R_6	R_1	R_2	OI2	R_5	R_2	R_6	R_6
O3	R_5	R_6	R_6	R_7	OI3	R_1	R_6	R_5	R_7
O4	R_5	R_6	R_6	R_2	OI4	R_1	R_6	R_5	R_2
O5	R_6	R_7	R_4	R_1	OI5	R_1	R_7	R_4	R_6
O6	R_6	R_7	R_8	R_1	OI6	R_1	R_7	R_8	R_6
O7	R_6	R_7	R_4	R_5	OI7	R_6	R_2	R_5	R_1
O8	R_6	R_7	R_8	R_5	OI8	R_6	R_2	R_5	R_5
I1	R_1	R_2	R_5	R_6	I2	R_1	R_2	R_5	R_6

Table 6.2: Equilibrium points position in each case.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
p_-								p_+			R_5	R_6	R_1	R_7

Equilibrium points are out from their corresponding region, so stable nodes of the outer regions can not be fixed-points for the system. As in the antisymmetric case, index theory assures that there are no fixed-points in the boundaries of the unit square. Therefore, there exist a limit cycle.

To construct curve C_2 as in the antisymmetric case, repulsive principal directions of saddle points are connected through lines in the outer regions R_5, R_6, R_7 and R_8.

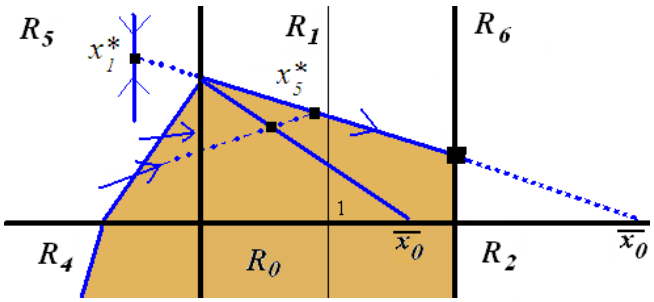


Figure 6.2: Positions for saddle point x_1.

In region R_1, saddle point x_1^* is out from its region. Its repulsive principal direction l_1 can reach the unit square or the neighboring outer region R_6 as can be seen in Figure 6.2.

$$l_1 : p_- x_0 = s x_1 - s^2 + p_+ p_-$$

To study it, we intersect l_1 with the border of the unit square $x_1 = 1$ obtaining

$$\bar{x_0} = \frac{s(1-s) + p_+ p_-}{p_-}$$

If $\bar{x_0} > 1$, l_1 does not reach the unit square. Parameters in this case fulfill $L_1 : s(s-1) + p_-(1-p_+) > 0$. Otherwise, the line reach the unit square and parameters fulfill $L_1 : s(s-1) + p_-(1-p_+) < 0$.

The same study can be done in region R_2. In this case, repulsive principal direction across x_2^* is

$$l_2 : sx_0 - s^2 + p_+ p_- = p_+ x_1$$

Again we intersect this line with the border of the unit square $x_1 = -1$ obtaining $\tilde{x_0}$. If $\tilde{x_0} > 1$, line l_2 not reach the unit square and parameters fulfill $L_2 : s(s-1) - p_+(1+p_-) > 0$. Otherwise, it reaches the unit square. However for the parameter range we are working on, $|p_+ - s| < 1, p_- < -1-s$ only first condition is fulfilled and so line l_2 pass across region R_2 connecting its neighboring regions R_5 and R_6.

We can conclude that we have two possibilities for C_2 construction depending on the parameter range of the cloning template as can be seen in Figures 6.3−6.4.

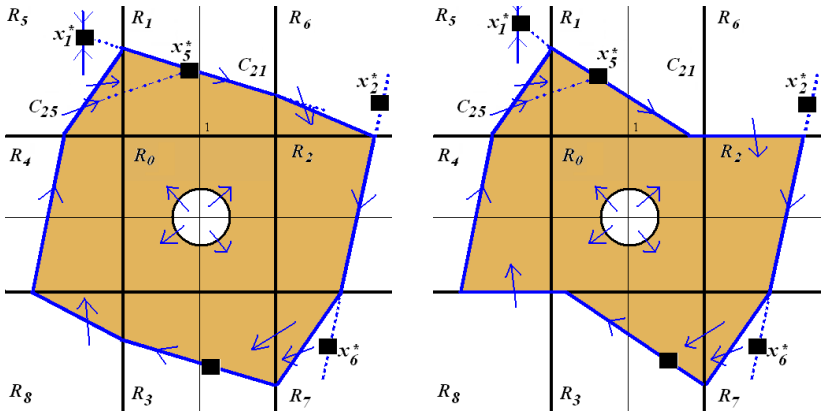


Figure 6.3: Case 1 : $L_1 > 0, L_2 > 0$ and Case 2 : $L_1 < 0, L_2 > 0$.

There can be different limit cycles here like for example those passing around the unit square in Case 1 where curves L_1 and L_2 are both positive or passing across all the regions except R_6 and R_8 in Case 2 where L_1 is negative and L_2 is positive.

Another example can be case OI1 where not all the equilibrium points are out from their region.

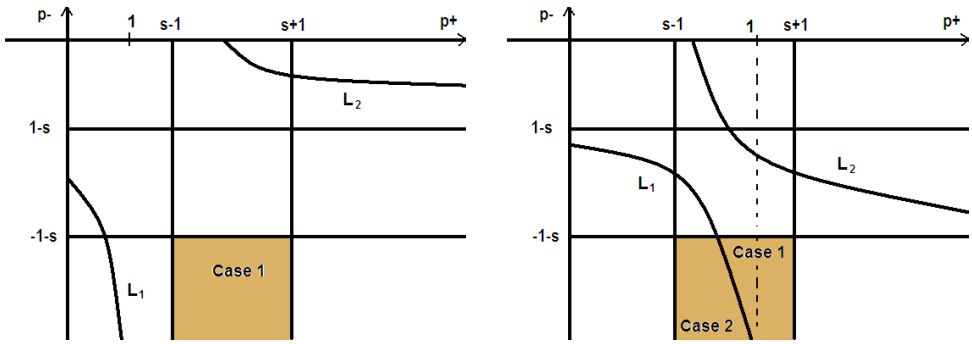


Figure 6.4: Parameter range for $s > 2$ and $1 < s < 2$.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
				p_-				p_+			R_5	R_2	R_1	R_6

Saddle point x_2^* and stable node x_6^* are inside their regions, and so the stable node becomes a fixed-point where trajectories can converge. In Figures 6.5−6.6 we see that for this parameter range, curve C_2 can not be constructed as we have done in the previous case.

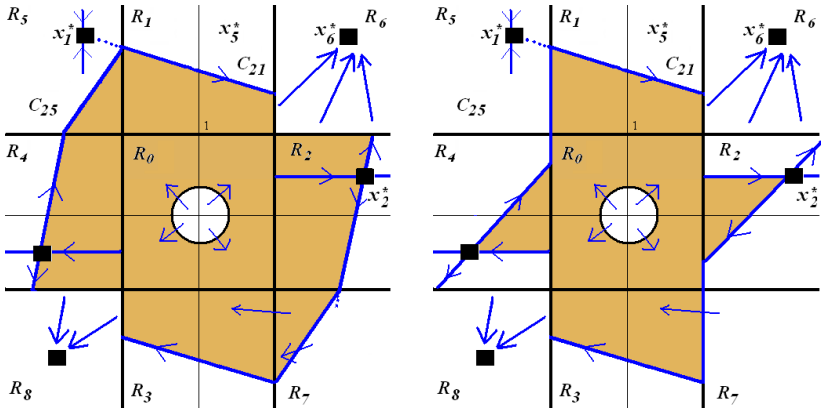


Figure 6.5: Case 1: $L_1 > 0, L_2 > 0$ and Case 3: $L_1 > 0, L_2 < 0$.

In this case, the system can have two dynamic behavior: converge to one of the stable nodes x_6^* or x_8^* located in their region or converge to a limit cycle different from those found before.

All other cases are detailed in Appendix E. The overlap of the limiting parameter regions for the different cases finally boils down to three topological distributions according to the value of s. These different distributions depend on the existence of limit cycles, limit cycles coexisting with two fixed-points, and the converge to one of the four stable nodes.

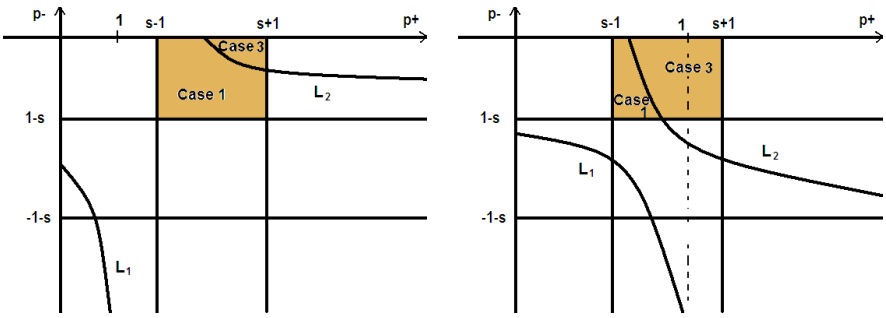


Figure 6.6: Parameter range for $s > 2$ and for $1 < s < 2$.

As a summary we show in Figures 6.7−6.9 C_2 construction in each different parameter region.

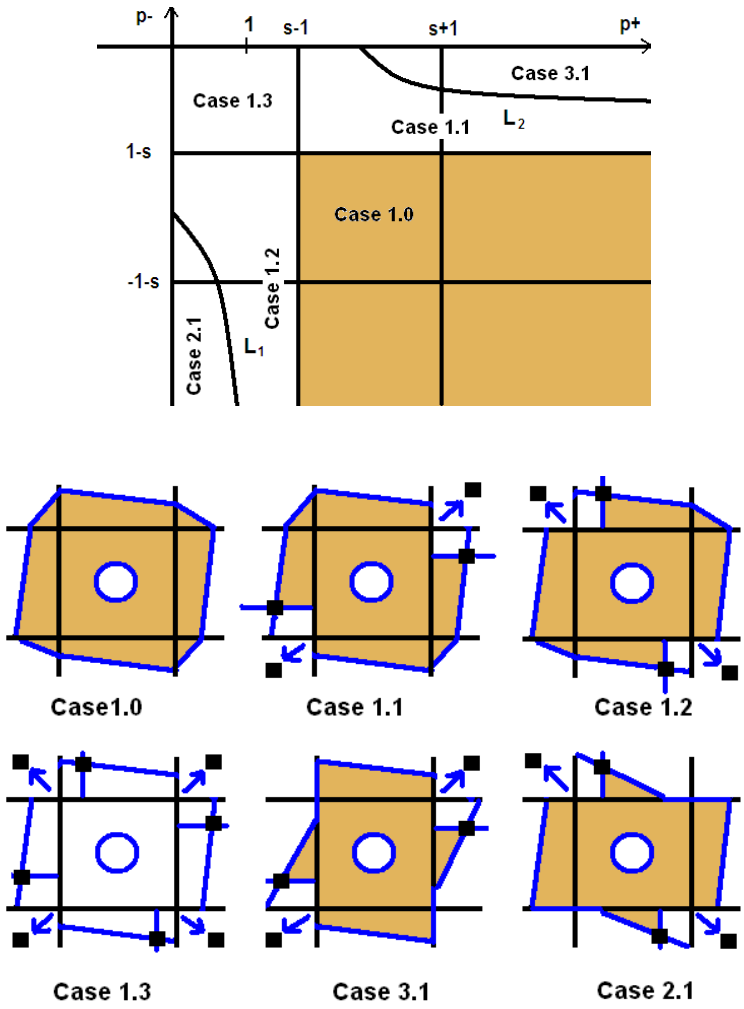


Figure 6.7: Convergence map for $s > 2$.

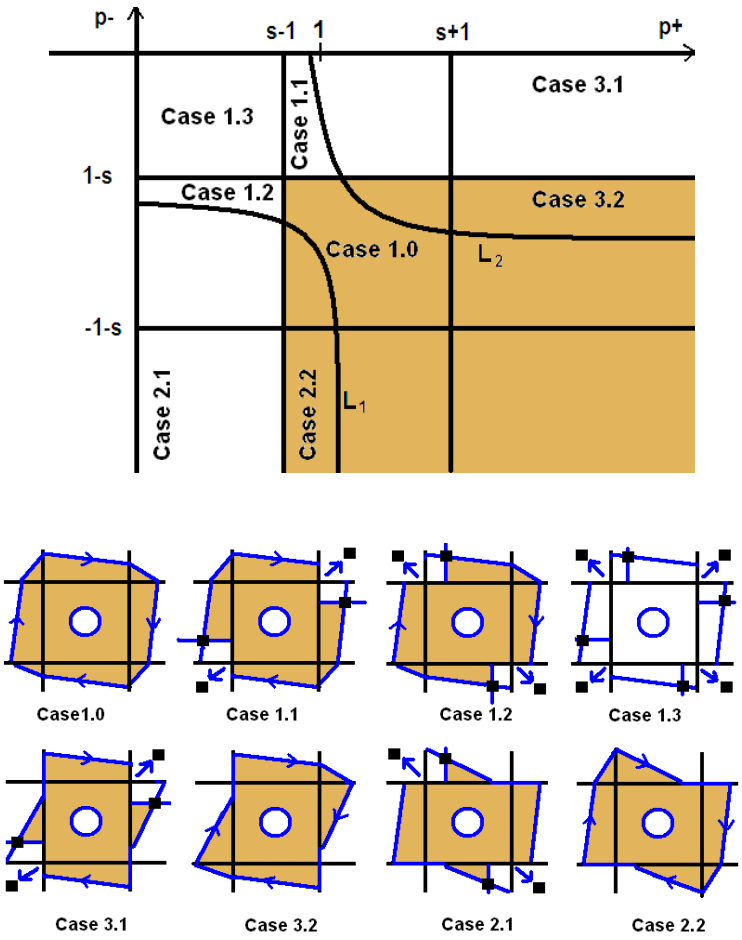
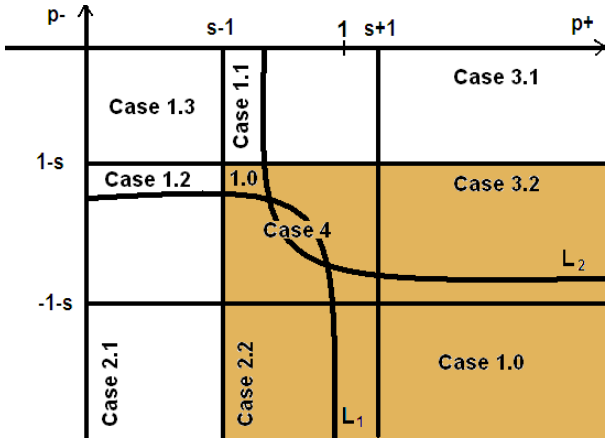


Figure 6.8: Convergence map for $\frac{1+\sqrt{2}}{2} < s < 2$.



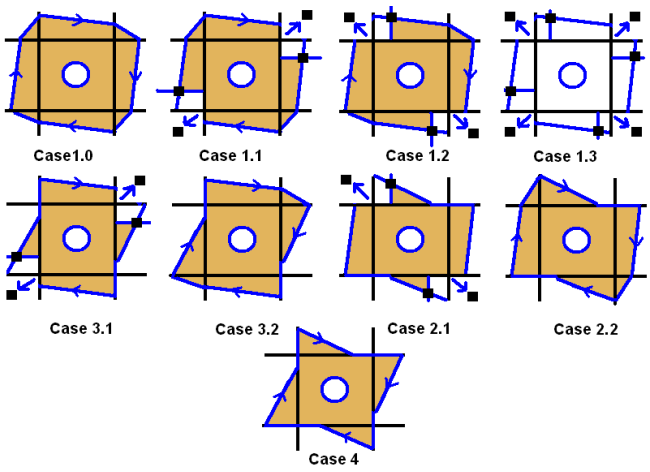


Figure 6.9: Convergence map for $1 < s < \frac{1+\sqrt{2}}{2}$.

6.2 Conclusions

From the results obtained in previous sections we may now discuss the existence of limit cycles depending on the equilibrium points position on the plane and therefore depending on parameter values s, p_+ and p_-.

For $s > 1$, equilibrium points in the nine regions are a spiral source in the middle with, saddle points and stable nodes all around. Their layout establishes in which cases a limit cycle exists, because it determines the geometry of the necessary limiting curve C_2. The CNN system can then be classified into three cases:

- Out cases: $Oi, i = 1, \ldots 8$ where all equilibrium points are out from their corresponding region,
- Out-In cases: $OIi, i = 1, \ldots 8$ where four equilibrium points are inside their region,
- In cases: $I1, I2$ where all of them are inside their region.

In Out regions $Oi, i = 1, \ldots 8$, parameter range is

$$\{p_+ > s-1, p_- < 1-s\} \cup \{p_+ < 1-s, p_- > s-1\},$$

and the system converges to a limit cycle. There are no fixed-points for the system because all stable nodes are out from their region so, sufficient conditions for the existence of limit cycles can be proven by defining curves C_1 and C_2 which delimit a region where trajectories remain as t tends to ∞. Repulsive principal directions of saddle points are the bases to define curve C_2 which trajectories will not cross. Moreover, we have also found different types of limit cycles according to the regions $R_i, i = 1, \ldots 8$ they cross. For example, for $s = 1.1$,$p_+ = 0.2$,$p_- = -1.3$ there exist a limit cycle crossing regions $R_0 \rightarrow R_1 \rightarrow R_3$ and for $s = 1.1$,$p_+ = 0.9$,$p_- = -1.3$, the limit cycle pass across regions $R_0 \rightarrow R_3 \rightarrow R_0 \rightarrow R_4 \rightarrow R_0 \rightarrow R_1 \rightarrow R_0 \rightarrow R_2 \rightarrow R_0$ (Figure 6.10).

On the other hand, for cases $OIi, i = 1, \ldots 8$, parameter range is

$$\{|p_+| < s-1, |p_-| > s-1, p_+p_- < 0\} \cup \{|p_-| < s-1, |p_+| > s-1, p_+p_- < 0\}$$

Four of the equilibrium points, two saddle points and two stable nodes are inside their corresponding region so the system has two possible dynamical behaviors: converge to a fixed point or converge to a limit cycle crossing the unit square and two saddle regions as can be seen in Figures 6.13−6.14.

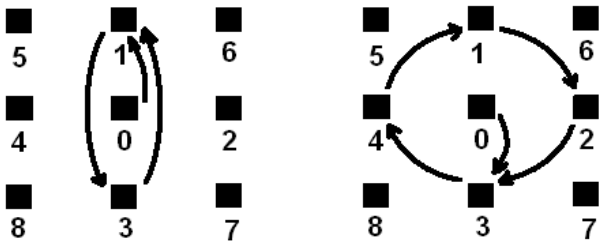


Figure 6.10: Examples of a limit cycle for an Out-In region and a limit cycle for an Out region.

This parameter range has been studied in [24], where a heteroclinic orbit connecting the two repulsive principal directions which reach the unit square exist if and only if CNN parameters fulfill :

$$\ln\left(\frac{\sqrt{|p_+p_-|}}{(s-1)\sqrt{(s-1)^2-p_+p_-}}\right)-\frac{s-1}{\sqrt{|p_+p_-|}}\arctan\left(\frac{s}{\sqrt{|p_+p_-|}}\right)=0. \quad (6.1)$$

Parameter space can then be separated by this condition in two regions with different dynamic behavior. A region where CNN converges to a fixed-point if the left hand side of ([24]) is negative and a region where converge to a limit cycle if the right hand side is positive. Figure 6.11 shows an example of these different dynamic behaviors. Let us note that in this case, limit cycles will pass only through three different regions because of the geometry of the saddle points.

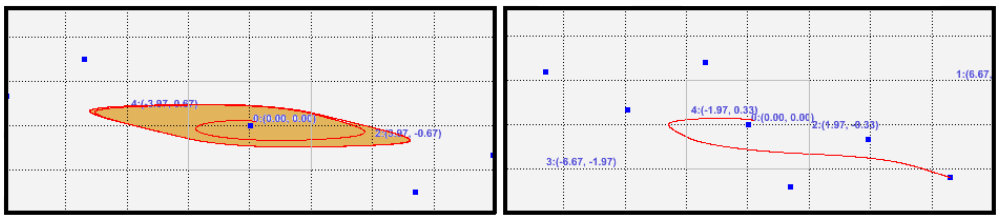


Figure 6.11: Examples for $s = 1.3, p_+ = -2; -4, p_- = 0.1; 0.2$ respectively.

Moreover, in [60] necessary and sufficient conditions for a CNN to converge to a fixed-point are found. For any s value different than 2, and $s-1 \geq \sqrt{|p_+p_-|}$, the system converges to a fixed-point.

For cases 2.1 and 3.1 where $s-1 \leq \sqrt{|p_+p_-|}$, a function g is defined

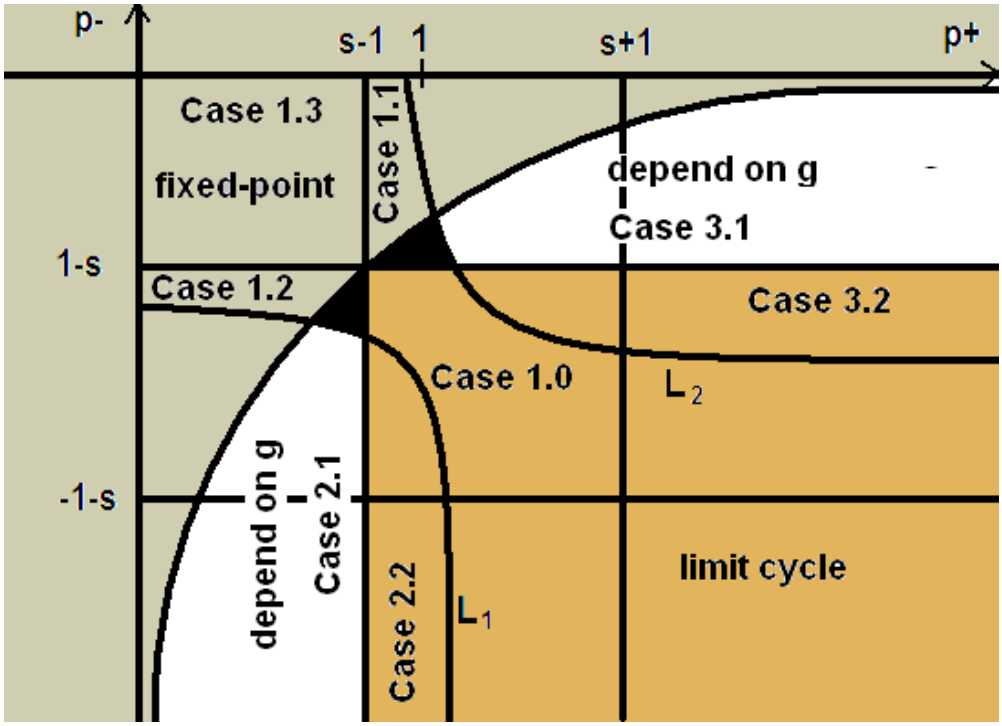


Figure 6.12: Convergence map for $p_+ > 0, p_- < 0$ and $s < 2$.

$$g(\sqrt{|p_+p_-|}, s) = \exp\left(\frac{2(s-1)}{\sqrt{|p_+p_-|}}\arctan\left(\frac{s}{\sqrt{|p_+p_-|}}\right)\right) + \frac{p_+p_-}{(s-1)^2(s^2-p_+p_-)}$$

As on (6.1), the function sign determines sufficient conditions for the existence of limit cycle. If $g(\sqrt{|p_+p_-|}, s) \leq 0$, the system converge to a limit cycle, else converge to a fixed-point.

In dark zones of cases 1.2 and 1.1 in Figure 6.12, repulsive principal directions does not reach the unit square. This fact avoid the construction of the heteroclinic orbit. Hence, sufficient conditions for the existence of a limit cycle has not been found. Finally, for In cases: $Ii, i = 1, 2$, all equilibrium points are inside their corresponding region and the system converge to one of the stable nodes becoming fixed-points for the system.

In Figures 6.13-6.14 we can see six different limit cycles that can be found in the autonomous case. Limit cycles passing across more than three regions (B,D,E,F) corresponds to Out cases while those passing across only three regions (A,C) correspond to Out-In cases .

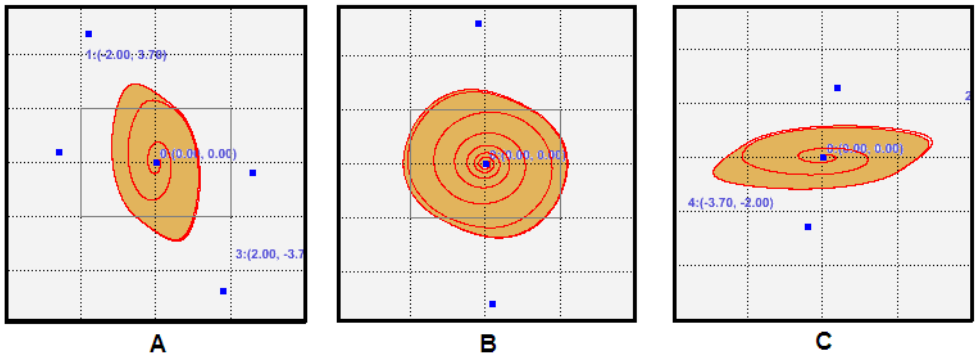


Figure 6.13: Examples for $s = 1.1$, $p_+ = 0.2; 0.9; 1.3$, $p_- = -1.3; -1.3; -0.2$ respectively.

These features may prove useful on classification problems. Each type of limit cycle is related to an area in parameter space. This allows us to design CNNs whose outputs are only limit cycles choosing the parameters of the cloning template adequately.

These different limit cycles let us think on classification problems. Parameter regions are related with each one of these different limit cycles, and let us found a mapping relation between them and the parameters of the CNN. This allows us to design CNNs whose outputs are only limit cycles choosing the parameters of the cloning template adequately. Furthermore,

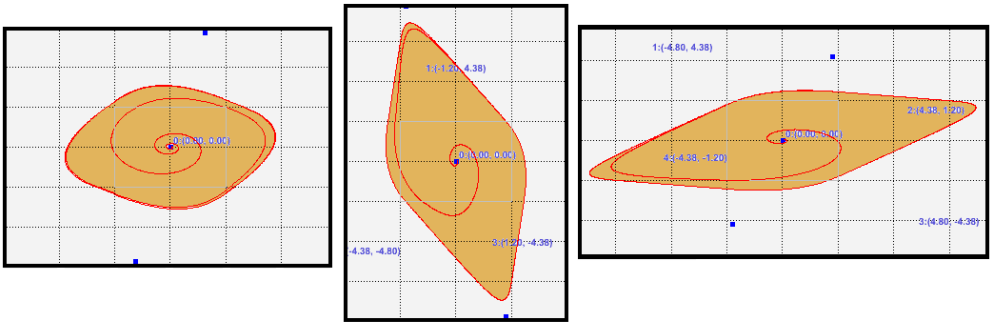


Figure 6.14: Examples for $s = 1.37, p_+ = 2, p_- = -1.5$, $s = 1.5, p_+ = 0.6; 2.4, p_- = -2.4; -0.6$ respectively.

adding points $\{(\pm 1, \pm 1)\}$ where the system converge in the complete stable cases, generic CNN can show up to ten possible outputs namely six limit cycles and four fixed points.

Classification problems with ten or less classes, might be solved by mapping a general CNN parameter space to classification problem variables. Although, a careful analysis of initial conditions would be required. In particular for Out-In cases where the system can move from a stable fixed point to a limit cycle.

Chapter 7

Limit cycle period

The non linearity of Cellular Neural Networks allows the CNN system to show different dynamic behavior such as converge to a fixed-point, to a closed curve or to a limit cycle. In the two neuron CNN case, limit cycles have been found for example in [24], [59], [60] and their existence and classification in the autonomous case has been studied in previous sections. Moreover, for circular CNNs with periodic boundary conditions the period of a limit cycle was computed in [54].

In this chapter, we are going to address the problem of finding the period value in the simplest case where a limit cycle exist : the autonomous antisymmetric two neuron CNN. In this case, we have found that there are two different limit cycles kinds depending on parameter range where they exist. Firstly, those passing around the unit square and secondly, limit cycles passing across the unit square. We will start focusing our study about the period value in the first case.

7.1 Computing the period function

As limit cycles kind depend on the CNN parameters, their period P will be found as a function of them $P(p_a, s)$. Working in the parameter range,

$$R = \{(p_a, s) \in \mathbb{R}^2 | s - 1 < |p_a|, s > 1, (p_a - \frac{1}{2})^2 + (s - \frac{1}{2})^2 > \frac{1}{2}\}, \qquad (7.1)$$

the CNN system in the autonomous antisymmetric case, converges to a limit cycle around the unit square. To find the correspondent period we are going to use the problem symmetry, and the CNN system dynamic behavior in the different regions. In the middle regions R_i, $i = 1, 2, 3, 4$, local equilibrium points are saddle points (m_i, k_i) and they are symmetric with respect the

origin. In fact, there is a $\pi/2$ rotation between them. A similar symmetric behavior appears in the out regions R_i, $i = 5, 6, 7, 8$ where equilibrium points are attractive points but none of them are located inside their region. This symmetric property is also valid for the limit cycle as can be seen in Figure 7.1.

As the dynamic behavior of the limit cycle is different depending on the region it crosses, we will compute the period locally inside each one of these regions. Let us name P_i, $i = 1, \dots 8$ the local part of the period in each region R_i, $i = 1, \dots, 8$.

From the symmetric property mentioned before, the local part of the period in the middle regions are the same, $P_1 = P_2 = P_3 = P_4$ just like in the out regions where $P_5 = P_6 = P_7 = P_8$. So the limit cycle period is:

$$P(p_a, s) = \sum_{i=1}^{i=8} P_i(p_a, s) = 4P_1(p_a, s) + 4P_5(p_a, s) \tag{7.2}$$

For instance, Figure 7.1 shows a limit cycle for parameters $s = 2$ and $p_a = 2$. Using the symmetry of this curve, we can see that computing P_1 and P_5 we can obtain the limit cycle period, and so the period of each output cell $y_i(t)$. To do it, we will use points Q_0, Q_1 and Q_2 connecting a closed curve between middle regions and out regions. Let Q_0 be the connecting point between regions R_4 and R_5, Q_1 the corresponding one connecting regions R_5 and R_1, and point Q_2 connecting regions R_1 and R_6.

Limit cycles directions depends on p_a sign. For a positive parameter p_a, it will follows a clock direction $R_6, R_2, R_7, R_3, R_8, R_4, R_5, R_1, \dots$ (7.3) otherwise, limit cycle goes counterclockwise.

$$\begin{bmatrix}1\\1\end{bmatrix} \to \begin{bmatrix}1\\y_1\end{bmatrix} \to \begin{bmatrix}1\\-1\end{bmatrix} \to \begin{bmatrix}y_0\\-1\end{bmatrix} \to \begin{bmatrix}-1\\-1\end{bmatrix} \to \begin{bmatrix}-1\\y_1\end{bmatrix} \to \tag{7.3}$$

$$\to \begin{bmatrix}-1\\1\end{bmatrix} \to \begin{bmatrix}y_0\\1\end{bmatrix} \to \begin{bmatrix}1\\1\end{bmatrix} \to \dots$$

To compute P_1 corresponding to region R_1, let us remind the CNN parametric solution (2.11) found on the stability chapter. For sake of clarity, let us name $x_0^1(t)$ and $x_1^1(t)$ the parametric solution of region R_1,

$$\begin{cases} x_0^1(t) = \alpha_0^1 s e^{(s-1)t} + m_1 \\ x_1^1(t) = -\alpha_0^1 p_a e^{(s-1)t} + \alpha_1^1 e^{-t} + k_1 \end{cases}$$

where $(m_1, k_1) = \left(-\frac{p_a}{s-1}, s + \frac{p_a^2}{s-1}\right)$ is the local equilibrium point, and index $k = 1$ of x_i^k corresponds to region R_1 (Table 5.3). Limit trajectory will start

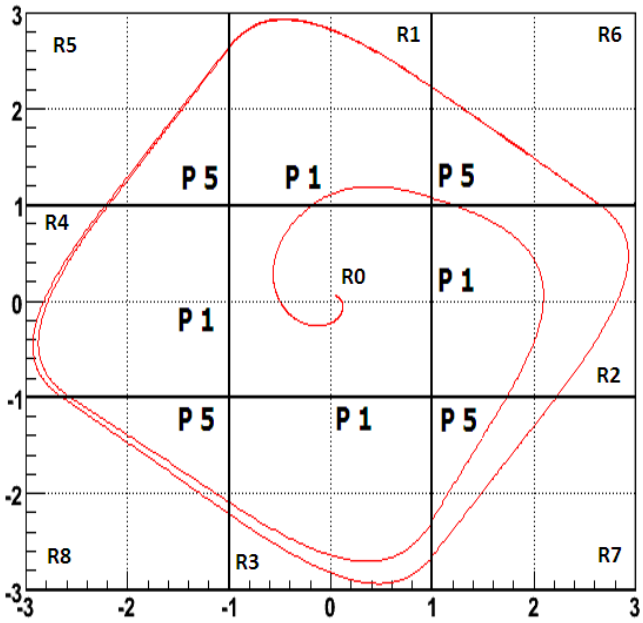


Figure 7.1: Limit cycle for parameters $s = 2$ and $p_a = 2$ with periods P_1 and P_5.

at

$$Q_1 = (x_0^1(0), x_1^1(0)) = (-1, x_1^1(0))$$

and will end at

$$Q_2 = (x_0^1(P_1), x_1^1(P_1)) = (+1, x_1^1(P_1))$$

as can be seen in Figure 7.2. Imposing these initial conditions, we obtain the system equations

$$\begin{cases} x_0^1(0) = \alpha_0^1 s e^0 + m_1 = -1 \\ x_0^1(P_1) = \alpha_0^1 s e^{(s-1)P_1} + m_1 = 1 \end{cases}$$

This system solution let us obtain P_1 corresponding to middle region R_1.

$$P_1(p_a, s) = \frac{1}{s-1} \ln\left(\frac{p_a + s - 1}{p_a - (s-1)}\right). \tag{7.4}$$

Now, we are going to do a similar study in out region R_5 in order to find P_5. Again, using the parametric solution of the CNN system (2.12). Let us name $x_0^5(t)$ and $x_1^5(t)$ the parametric solution of region R_5,

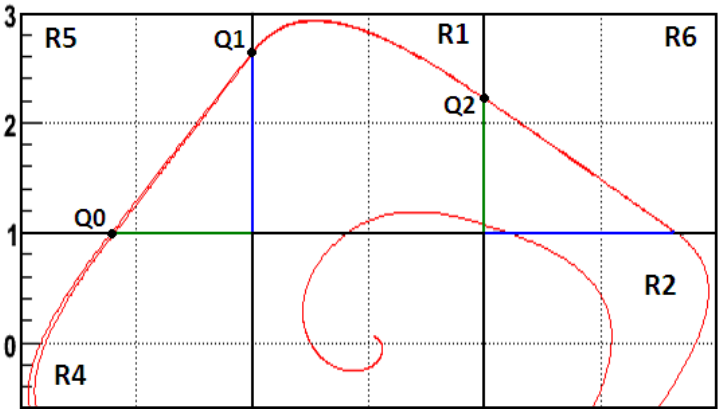


Figure 7.2: Initial and final points of limit cycle across region R_1.

$$\begin{cases} x_0^5(t) = \alpha_0^5 e^{-t} + m_5, \\ x_1^5(t) = \alpha_1^5 e^{-t} + k_5, \end{cases}$$

where $(m_5, k_5) = (p_a - s, p_a + s)$ (Table 5.3) is the local equilibrium point, we will find P_5. The limit trajectory in this region starts at the boundary point

$$Q_0 = (x_0^5(0), x_1^5(0)) = (x_0^5(t_0), +1)$$

and ends at

$$Q_1 = (x_0^5(P_5), x_1^5(P_5)) = (-1, x_1^5(P_5))$$

as can be seen in Figure 7.2. From this, we obtain the system equations which solution in this out region is P_5 (7.5) .

$$\begin{cases} x_1^5(0) = \alpha_1^5 e^0 + k_5 = 1, \\ x_0^5(P_5) = \alpha_0^5 e^{-P_5} + m_5 = -1. \end{cases}$$

$$P_5(p_a, s) = \ln\left(\frac{\alpha_0^5}{s - 1 - p_a}\right) \tag{7.5}$$

Nevertheless P_5 depends on parameter α_0^5 of the parametric solution. To find it, we use the limit cycle symmetry, and the fact that limit cycle is a closed curve in the limit. Boundary points connecting limit cycle curves of neighboring regions must then fulfill equations (7.6).

$$\begin{cases} x_1^5(0) = 1 \\ x_0^1(0) = -1 \\ x_1^1(0) = x_1^5(P_5) \\ x_1^1(P_1) = -x_0^5(0) \end{cases} \tag{7.6}$$

Solving these conditions we find α_0^5 value as,

$$\alpha_0^5 = -\frac{1}{2s}\left(p_a[(s-1-p_a)A + s - 1 + p_a] + \right.$$
$$\left. + \sqrt{p_a^2(s-1-p_a)^2A^2 + p_a^2(s-1+p_a)^2 - A(2p_a^2+4s^2)(p_a^2-(s-1)^2)}\right)$$

where $A = \left(\frac{p_a-(s-1)}{p_a+s-1}\right)^{\frac{1}{s-1}}$.

Therefore using (7.4) and (7.5), the period of this kind of limit cycle is $P(p_a, s)$,

$$P(p_a,s) = 4\frac{1}{s-1}\ln\left(\frac{s-1+p_a}{p_a-(s-1)}\right) + 4\ln\left(\frac{-\alpha_0^5}{p_a-(s-1)}\right) = \\ 4\ln\left(\left(\frac{s-1+p_a}{p_a-(s-1)}\right)^{\frac{1}{s-1}} \cdot \frac{-\alpha_0^5}{p_a-(s-1)}\right) \tag{7.7}$$

In Figure 7.3, we can see the period function (7.7) depending on parameters s and p_a.

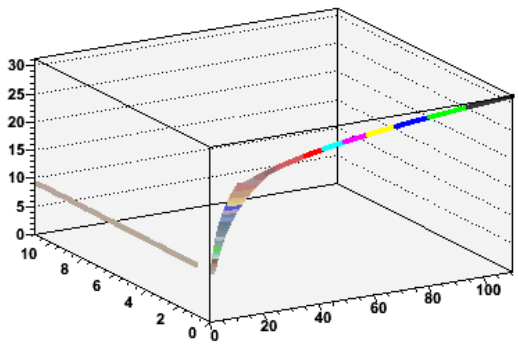


Figure 7.3: Period function $P(p_a, s)$.

Yet complex, this formula is analytical, and so allows us to evaluate beforehand the limit cycle frequency obtained from a given set of parameters s and p_a. A quick analysis shows that periods become infinite in some limits, and hints the fact that a minimum period P_a or maximum frequency should exist.

7.2 Generating clock signals using CNNs

The results obtained from the period study drive us into the possible applications of limit cycles like for example, use them to generate clock signals. We propose using a simple continuous time two neuron CNN with the

standard piecewise linear output function, to tackle this problem. We will work with fully antisymmetric weights so that the system is in particular limit cycle condition. By the nature of the piecewise linear function, neuron outputs can be used as highly stable and well shaped clock signals. The clock frequency is actually a direct function of the connection values and might be modified by reprogramming the weights to cover a very wide range of values.

There are many different options to implement CNN weights. For instance, based on reference [40] where a model of neuron cell with programmable memristor connections is described, CNN weights can be described in such a way. The Memristor, proposed by L.O.Chua in 1971 [15] as a missing linear response circuit element, beyond the resistor, the capacitor and the inductor, was finally found in a nanoscale implementation acting as a resistive memory [56]. Among the very many applications of such a device, we find synapse implementation for artificial neural networks [55]. Based on this, reference [40] proposes a cell structure for a Cellular Neural Network where the interconnecting weights are memristors whose value is programmable to any continuous value in continuous time, with circuitry implemented in the cell structure.

By looking at Figure 7.1, we get a qualitative description of the behavior of the cell outputs. If we take, for instance, neuron 0. When the system is in regions R_6, R_2 and R_7, which are run consecutively, neuron state x_0 is larger than one and thus, output value y_0 is 1. Reversely, when the system is in region R_8, R_4 and R_5, x_0 is lower than one and so y_0 is -1. In region R_3, y_0 transients from 1 to -1 while in region R_1 rises from -1 to 1. This is, in fact, the behavior of a clock signal. Naturally, neuron 1 output behaves just the same, delivering an identical signal because of the system symmetry but, delayed. The delay can be estimated to be the transit time in region R_5, which, by symmetry is the same as the transit time in regions R_6, R_7 or R_8.

Quantitatively, we may observe the signal shapes from the exact solution of the differential equations in Figure 7.4, which confirms the fact that both outputs generate very clear delayed clock signals.

If we now look at the clock parameters, frequency shall be given, by the inverse of the period P, and as such, can be tuned by modifying the CNN parameters. The rise time is the transit time through region R_1.

$$t_{rise} = P_1(p_a, s) = t_{fall}$$

By symmetry, it is equal to the fall time. The time spent at maximum and minimum values is then,

$$t_1 = P_1(p_a, s) + 2P_5(p_a, s),$$

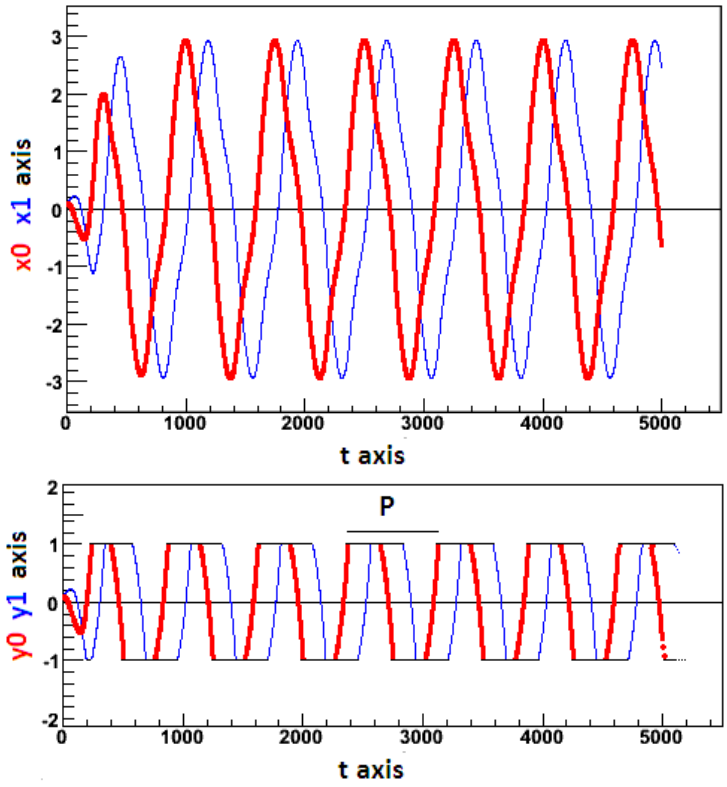


Figure 7.4: Periodic states and outputs for parameters $s = 2$ and $p_a = 2$ where the system converges to a limit cycle around the unit square.

and the delay between the clock signals is $P_5(p_a, s)$. Since the period is a bivariate continuous function, we may consider tuning the parameters in order to obtain a given period, and then play to set the rise time or the delay between both clocks.

The tuning possibilities are, in principle not unlimited. Since, to reproduce a periodic behavior on the CNN outputs, we must work in the parameter range (7.1) where symmetric limit cycles exist. At first sight, this parameter range is not finite but we can bound it in each particular case. This is important because to use memristors as weight parameters, we need to work in a bounded range $[s_1, s_2] \in [0, 1]$ of the window function f [40]. In our case, for certain parameter s, antisymmetric parameter p_a belongs to $[P_a(s), k]$, $P_a < k < \infty$. Rescaling this range into $[s_1, s_2]$ where $s_1 = P_a/k + 1$ and $s_2 = k/k+1$, the weight state belongs to $[0, 1]$. This means that we will work in a bounded parameter range, and so a minimum and a maximum value of the period function and therefore frequency shall exist.

7.3 Conclusions

In this section we have studied the period function in the particular case in which two neurons have opposite sign connection values, also known as the antisymmetric case. The limit cycle period as a function of the self-feedback parameter s and the antisymmetric one p_a, has been obtained using the local study made in each of the nine regions where the CNN system is linear. The symmetry of the problem has allowed to work only in two different regions: middle region R_1 and out region R_5. On the rest, system dynamics is the same and so, the local period functions of middle regions are all equal to the local one P_1 while in the out ones, are equal to P_5. The computation of this two local parts of the period function, gives us the key point to determine the general one $P_a = 4P_1 + 4P_5$.

From the parametric solutions, we have found a curve starting at a point Q_1 of the boundary line $x_0 = -1$ of middle region R_1 and ending at Q_2 on line $x_0 = 1$. Similar arguments are used to find the local period in out region R_5 but additional conditions of symmetry are necessary to find the necessary arbitrary constants of the parametric solutions.

As we can see on the period figure (Figure 7.7), $P(p_a, s)$ is an increasing function. Working inside the period domain one can think that it should have a minimum value representing a break point between periods of limit cycles surrounding the unit square and those passing across the unit square. Nevertheless, the study of the different period properties such as the existence of a minimum value or its behavior on the boundary curves of its domain is

still lacking.

At last, we may discuss about some possible uses of this result like for example use limit cycles to store individual memories or use it to reproduce a clock signal. As CNNs have a periodic behavior for certain range of their weights, states of each neuron behaves then like a periodic signal and so the final outputs. On this grounds, each of the saturation regions (out regions) where the limit cycle pass across (P_5) could be thought as memory state. Transitions from one memory state to another are done when it passes across the middle regions (P_1).

On the other hand, the combination of the cell model with memristor connections proposed by Lehtonen and Laiho [40] using the limit cycles characteristics of the two neuron Cellular Neural Network with fully antisymmetric connections allows us to generate digital clocks with almost arbitrary frequency.

Part IV

Convergence comparison

Chapter 8

Topological equivalence

The original model of Cellular Neural Networks (CNN) introduced by Chua and Yang [5] use the piecewise linear function (1.2) as activation function which is in fact non bijective and non differentiable at $x = \pm 1$. To avoid the difficulties of these features, for both theoretical and practical considerations, many authors [12],[5],[51] resort to a sigmoidal function, typically hyperbolic tangent as activation function. The core argument for this substitution is the similar behavior of both functions around three important domains, namely the origin and the two asymptotic limits $x \to \pm\infty$, as shown in Figure 8.1. We might summarize the strategy saying that hyperbolic tangent is a differentiable and bijective version of piecewise linear function. As such, dynamic behavior of these two systems should be the same. On this grounds, results established for hyperbolic tangent are extended to the piecewise linear case.

However, in our analysis of the dynamic behavior of the two neuron CNN, we have found that the non-bijective nature of the piecewise linear function plays an important role. For this reason we decided to peruse this equivalence. For an autonomous two neuron CNN, we will compare the stability of these two dynamical systems [22]. Both systems show the same dynamic behavior except for some regions of parameter space.

8.1 The Chua-Yang and $\tanh x$ models

As we have seen in the stability chapter, a CNN can be described by a system equations like $\dot{x} = -x + Ay + Bu + z$, where each cell is a dynamical system which has an external input, an output and a state: u, y, x. Output y may be any function of the input or the threshold but, for a standard CNN the output is defined by the piecewise linear function of the states

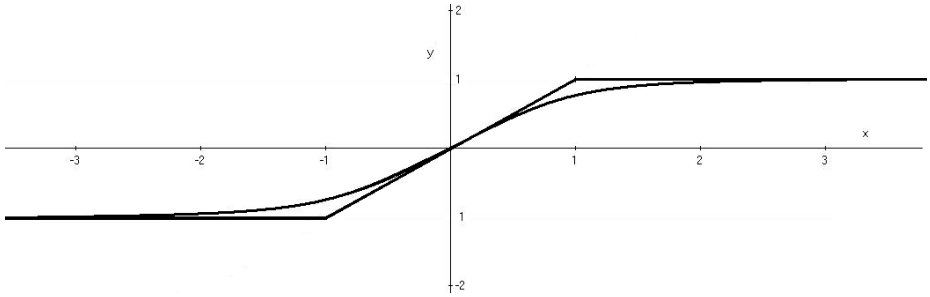


Figure 8.1: Hyperbolic tangent and piecewise linear function.

$y = f(x) = pwl(x)$ (1.2), as we have been using along the different chapters. For the hyperbolic one, it is defined by the hyperbolic tangent $y = f(x) = \tanh(x)$.

To compare both systems we first study some of the main differences between them. First, if output is defined by the hyperbolic tangent, then the system is differentiable along the plane while in the piecewise linear one, we loose the differentiability on lines $x_i = \pm 1$.

The state space for the piecewise linear system is classified into nine regions where the dynamic behavior is different. Namely, the central region where $|x_i| \leq 1$, four saturation regions where $|x_i| > 1$ and four partial saturation regions where $|x_i| > 1$ and $|x_j| \leq 1$ for $i, j = 0, 1$. On the border of these regions, there is a significant change on the CNN dynamics. On the other hand, the differentiable nature of the hyperbolic tangent does not allow for such a partition of phase space.

In general, mathematical criteria to study the stability of both systems is similar. For example the State-Boundedness Criterion, which assures that the state of each cell is bounded, is valid for $f(x) = \tanh(x)$ and for $f(x) = pwl(x)$ because they both are bounded functions.

However, original CNN system is not a differentiable model at the border of the different linear regions, so some standard dynamical system theorems such as Bendixson's criterion or the Lyapunov theory, which require this hypothesis can not be directly applied. Meanwhile, the hyperbolic model is a differentiable one, so these tools can be used as follows.

Theorem 9 (Bendixon criterion)**.** Given a simply connected region D in the plane. If the divergence of the vector field is always positive or is always negative inside D, then there cannot be a periodic orbit inside D.

Hyperbolic vector field is

$$F(x_0, x_1) = (F_0(x_0, x_1), F_1(x_0, x_1)) =$$

$$(-x_0 + s\tanh x_0 + p_+ \tanh x_1, -x_1 + s\tanh x_1 + p_- \tanh x_0)$$

Hence, the vector field divergence is:

$$divF = \left(\frac{\partial F_0}{\partial x_0} + \frac{\partial F_1}{\partial x_1}\right) =$$

$$-1 + s(1 - \tanh^2 x_0) - 1 + s(1 - \tanh^2 x_1) =$$
$$= -2 + s(1 - \tanh^2 x_0 + 1 - \tanh^2 x_1) < 2(s - 1)$$

For $s < 1$, the divergence of the vector field is always negative, and so the system does not converge to a periodic orbit. Therefore, both models show exactly the same behavior in the symmetric case. For a symmetric template, theorem used in [5] to assure the stability of a CNN is fulfilled.

Theorem 10 (Complete Stability Criterion)**.** All trajectories of the hyperbolic CNN converge to an equilibrium state, which in general depends on initial states, if:

1. The cloning template is symmetric, $A^T = A$,
2. The scalar function $f(x_i)$ is differentiable with positive slopes and bounded,
3. All equilibrium points are isolated.

In the hyperbolic case, function $f(x) = \tanh x$ is differentiable with positive slopes and bounded, so there exist a Lyapunov function

$$V(x_0, x_1) = -\frac{1}{2}(y_0, y_1)^T A(y_0, y_1) + \sum_{i=0}^{1} \int_{\theta}^{y_i} f^{-1}(\tau) d\tau$$

The system then, converges to an attractive equilibrium point. In the piecewise linear case, function $f(x) = pwl(x)$ is not differentiable and it is not injective in the saturation regions. Therefore, theoretically this criterion can not be applied. Many times, the argument of being arbitrarily closely to an injective function such as the hyperbolic one has been used in the literature. This would only hold if both systems are topologically equivalent.

8.2 Topological equivalence

In order to establish if both systems are topologically equivalent, let us recall first the definition of the concept.

Definition 3. *A function between two topological spaces is called a homeomorphism if it has the following properties:*

1. *f is a bijection,*
2. *f is continuous,*
3. *the inverse function f^{-1} is continuous.*

Homeomorphisms are the isomorphisms in the category of topological spaces, that is, they are mappings which preserve all topological properties of a given space.

Definition 4. *Two vector fields X_0 and X_1 are said to be topologically conjugate if there exist an homeomorphism $h : \Delta_0 \subset \mathbb{R}^2 \to \Delta_1 \subset \mathbb{R}^2$ such that*

$$Dh(X_0(p)) = X_1(h(p)).$$

Definition 5. *Let X and Y be topological spaces. A flow φ on X is topologically semiconjugate to a flow ψ on Y if there is a continuous surjection $h : Y \to X$ such that*

$$\varphi(\ h(y), t\ ) = h(\ \psi(y, t)\ )$$

for each $y \in Y, t \in \mathbb{R}$. If h is a homeomorphism then ψ and φ are topologically conjugate.

This means that if there exist a homeomorphism between two dynamical systems, every trajectory from the first one corresponds to a trajectory from the second. If both systems are topologically equivalent, there should be a one-to-one correspondence between the equilibrium points of the two vector fields and a one-to-one correspondence between limit cycles.

Equilibrium points of the hyperbolic model, can be found solving

$$(F_0(x_0, x_1), F_1(x_0, x_1)) = (0, 0)$$

From this, we obtain equations

$$\begin{aligned} C_0 &: p_- x_0 = s x_1 - (s^2 - p_+ p_-) \tanh x_1 \\ C_1 &: p_+ x_1 = s x_0 - (s^2 - p_+ p_-) \tanh x_0 \end{aligned} \tag{8.1}$$

The number of intersection points of these curves is the number of equilibrium points. In many particular cases, both systems have the same number of equilibrium points located in the same regions of the plane. Let us note that the origin is always a solution of (8.1), and so $(0, 0)$ is an equilibrium point just like in the piecewise linear system. Furthermore, using a linearisation

technique we can see that for parameter range $s > 1$ and $p_+p_- < 0$ the origin is a repulsive equilibrium point like in the piecewise linear system (2.1).

$$\begin{aligned}
&\dot{x_0} = F_0(0,0) + \tfrac{\partial F_0(0,0)}{\partial x_0}x_0 + \tfrac{\partial F_0(0,0)}{\partial x_1}x_1 = \\
&= (-1+s)x_0 + p_+x_1 \\
&\dot{x_1} = F_1(0,0) + \tfrac{\partial F_1(0,0)}{\partial x_0}x_0 + \tfrac{\partial F_1(0,0)}{\partial x_1}x_1 = \\
&= (-1+s)x_1 + p_-x_0
\end{aligned}$$

Eigenvalues of this CNN matrix are $\lambda = (s-1) \pm j\sqrt{p_+p_-}$. Its real part is positive for $s > 1$, and so it is a repulsive point. Solving system (8.1) can be hard, so to find the number of solutions we intersect the asymptotic lines of a curve with the other one. For example, an asymptotic line of curve C_0 with curve C_1,

$$\begin{cases} C_0 & \text{asymptotic line: } p_-x_0 = sx_1 - (s^2 - p_+p_-) \\ C_1 & \text{curve: } p_+x_1 = sx_0 - (s^2 - p_+p_-)\tanh x_0 \end{cases}$$

which can be written as:

$$\left\{ x_1 = \frac{p_-}{s}x_0 - \frac{s^2 - p_+p_-}{s}, \quad \tanh x_0 = \frac{x_0}{s} - \frac{p_+}{s} \right\} \tag{8.2}$$

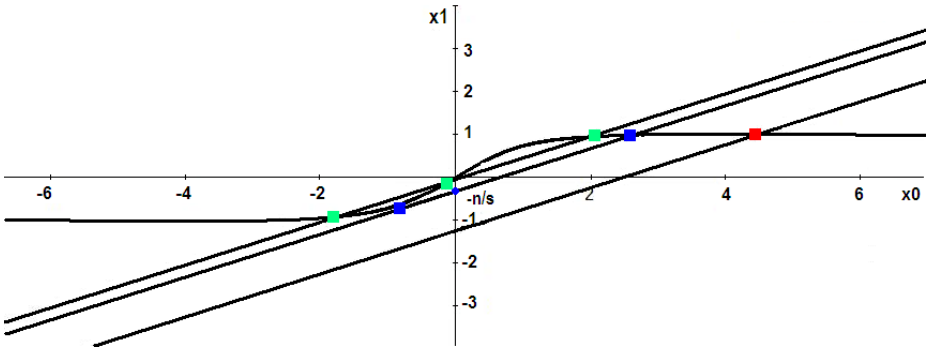


Figure 8.2: Intersection between curve $C_1 : p_+x_1 = sx_0 - (s^2 - p_+p_-)\tanh x_0$ and the asymptotic line $p_-x_0 = sx_1 - (s^2 - p_+p_-)$ of curve C_0. .

Solving equation (8.2) means to intersect line $x_1 = \frac{x_0}{s} - \frac{p_+}{s}$ with hyperbolic tangent function. There can be one, two or three intersection points of these curves depending on the line incline and position. In fact, solution will depend on parameter s. Finding the tangent line of function $\tanh x_0$ with incline $\frac{1}{s}$ for parameter $s > 1$, we can study the number of solutions of system (8.2).

$$x_1 = \frac{1}{s}x_0 - \sqrt{1 - \frac{1}{s}} + \frac{1}{s}\tanh^{-1}\left(\sqrt{1 - \frac{1}{s}}\right) = \frac{x_0}{s} - \frac{n}{s}$$

This tangent line is equal to line (8.2) if and only if

$$p_+ = s\sqrt{1-\frac{1}{s}} - \tanh^{-1}\sqrt{1-\frac{1}{s}} := n$$

Let us note that for $s < 1$, do not exist such a line. In this case, there is only one intersection point but for $s > 1$ and $p_+ > 0$,

$$\begin{aligned} &p_+ > n > 0 \Rightarrow \text{ there exist only one intersection point,} \\ &p_+ = n \Rightarrow \text{ there exist two intersection points,} \\ &0 < p_+ < n \Rightarrow \text{there exist three intersection points.} \end{aligned}$$

The same study can be done for the other tangent line (Figure 8.2). For $p_+ < 0$,

$$\begin{aligned} &p_+ < -n \Rightarrow \text{ there exist only one intersection point,} \\ &p_+ = -n \Rightarrow \text{ there exist two intersection points,} \\ &0 > p_+ > -n \Rightarrow \text{there exist three intersection points.} \end{aligned}$$

From the intersection of the two asymptotic lines of C_0 with curve C_1, and doing the same study for the asymptotic lines of C_1 with curve C_0, we obtain the number of intersection points. Hence, we can deduce the number of equilibrium points of the system. If parameters fulfill $0 > n > p_+$ and $p_- > |n|$ for $s > 1$, there is one point from each intersection. This corresponds to one equilibrium point for the system: the origin (first graph in Figure 8.3), or three equilibrium points (first graph in Figure 8.4). We obtain three points from each intersection of the tangent lines with the curve if $n < p_+ < 0$, which means that there are nine equilibrium points for $0 > p_+ > n$ and $0 < p_- < |n|$ (second graph in Figure 8.4).

Finally, if we obtain three points from the intersection of the asymptotic lines of C_0 with curve C_1, and one point from the other intersection or vice versa, there are five equilibrium points for the system(second graph in Figure 8.3). With these results, we can print an equilibrium point map which is symmetric with respect the origin.

Let's note that this map is different from the piecewise linear map because in general $n \neq s - 1$. This means that there are some parameter regions where the systems exhibit different dynamic behavior because they have a different number of equilibrium points (Figure 8.5). Therefore, they are not topologically equivalent.

For instance, for $s = 1.1$, $p_+ = -0.0.5$ and $p_- = 3$, piecewise linear system has five equilibrium points while there is a single equilibrium point for the hyperbolic one. The original system can converge to one of the two

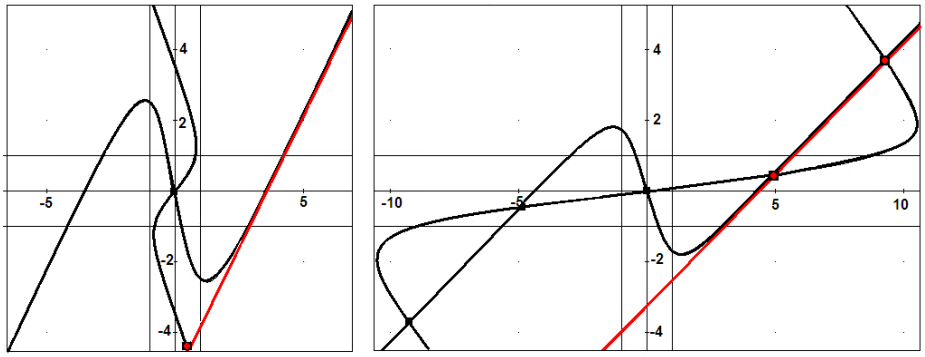


Figure 8.3: Examples of one and five equilibrium points. Black points are the equilibrium points and the red ones, are the intersection points between the asymptotic line and the hyperbolic tangent function.

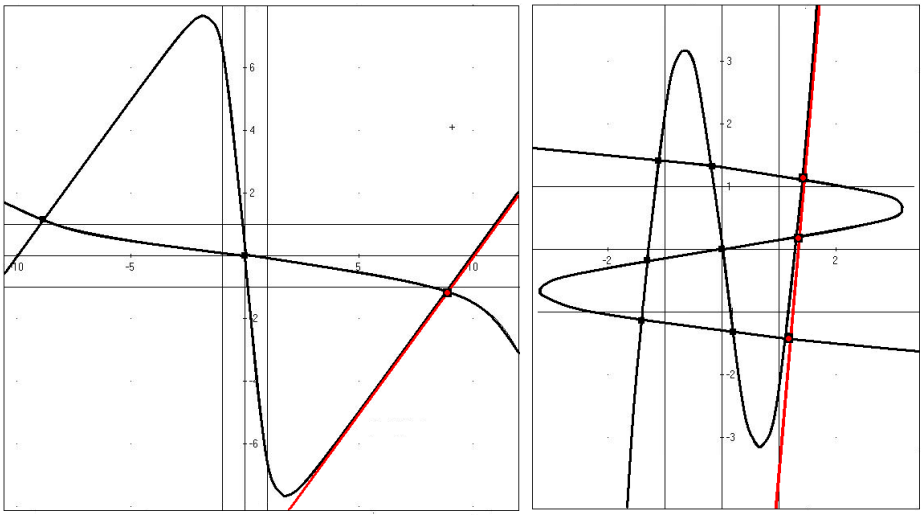


Figure 8.4: Examples of three and nine equilibrium points. Let us note that in the first graph, there is only one intersection point between the red asymptotic line and the hyperbolic tangent just like in the first example of Figure 8.3 but corresponding to three equilibrium points of the system instead of one.

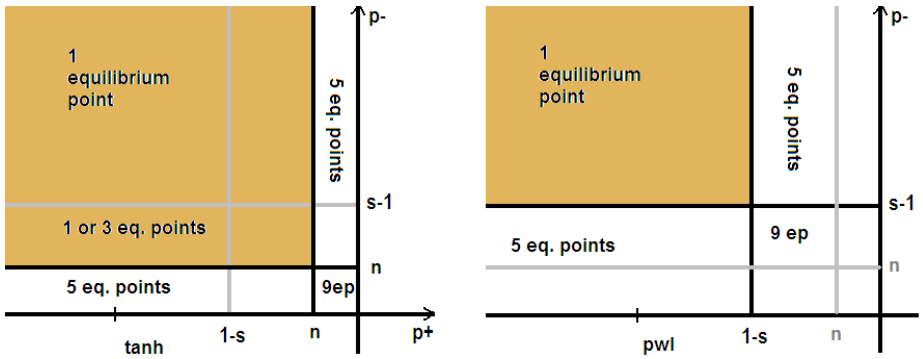


Figure 8.5: Equilibrium points regions for the hyperbolic model and for the piecewise linear model for $p_+ < 0$ and $p_- > 0$.

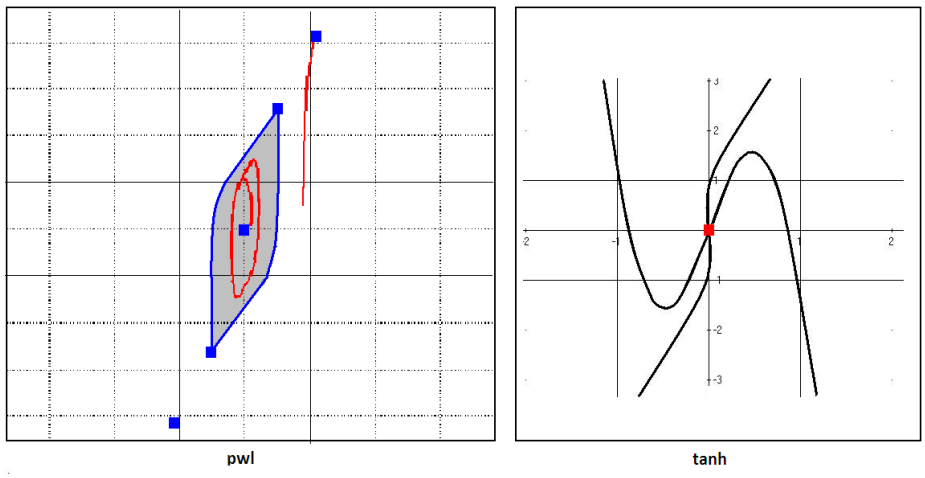


Figure 8.6: Example of different number of equilibrium points for parameters $s = 1.1$, $p_+ = -0.0.5$ and $p_- = 3$. There are five equilibrium points for the piecewise linear system and so depending on the initial conditions (inside or outside the gray region defined by an heteroclinic orbit) the system can converge to a limit cycle or to an attractive fixed-point in a saturation region. For the hyperbolic one, there exist only one repulsive equilibrium point $(0, 0)$ and so a trajectory can not converge to a fixed-point.

attractive fixed-points or to a limit cycle depending on the initial conditions (Figure 8.6). On the other hand, hyperbolic system will converge to a limit cycle without any dependence on initial conditions.

Nevertheless, in some regions the number of equilibrium points is the same, and so they are likely to show an equivalence while in others they are manifestly not. Equilibrium points of the hyperbolic system should then be close the piecewise linear ones. To study it we use for example the equilibrium point $x^* = (p_+ - s, -p_- + s)$ from one of the saturation regions $\{(x_0, x_1) \in \mathbb{R}^2 | x_0 < -1; x_1 > 1\}$. Evaluating this point in the vector field of system (2.1) using the hyperbolic output, we can approximate the distance between equilibrium points of both systems.

$$F_0(x^*) = -(p_+ - s) + s\tanh(p_+ - s) + p_+\tanh(-p_- + s)$$
$$F_1(x^*) = -(s - p_-) + s\tanh(-p_- + s) + p_-\tanh(p_+ - s)$$

As we can not evaluate the hyperbolic tangent on this points, we use the asymptotic expansion $\tanh x_1 \sim 1 - e^{-2x_1}$ and $\tanh x_0 \sim -1 + e^{2x_0}$.

$$F_0(x^*) \sim -(p_+ - s) + s(e^{2(p_+ - s)} - 1) + p_+(1 - e^{-2(-p_- + s)})$$
$$F_1(x^*) \sim -(s - p_-) + s(1 - e^{-2(-p_- + s)}) + p_-(e^{2(p_+ - s)} - 1)$$

For example, first component of vector field $F_0(x^*)$,

$$-(p_+ - s) + s(-1 + e^{2(p_+ - s)}) + p_+(1 - e^{-2(-p_- + s)}) =$$

$$= se^{2(p_+ - s)} - p_+e^{-2(-p_- + s)} \to 0 \text{ if } x^* \to (-\infty, +\infty)$$

If equilibrium point of the hyperbolic system is far from the boundary of this saturation region, the distance tends to 0. If this point is close to the boundary, which means that value $s - p_+$ is near to -1 and value $-p_- + s$ is near to $+1$,

$$F_0(x^*) \sim \frac{s}{e^{2(-p_+ + s)}} - \frac{p_+}{e^{2(-p_- + s)}} < \frac{s}{e^2} - \frac{s-1}{e^2} = \frac{1}{e^2}$$

$$F_1(x^*) \sim \frac{-s}{e^{2(p_+ - s)}} + \frac{p_-}{e^{2(-p_- + s)}} < \frac{s}{e^2} - \frac{s-1}{e^2} = \frac{1}{e^2}$$

the distance is bounded by $\frac{1}{e^2} \sim 0.1353352$. The same value is obtained for the other equilibrium points located in saturation regions. They can be attractive fixed-points for the piecewise linear system if parameter s is larger than 1, and $p_+p_- < 0$. The distance between the equilibrium points of both systems is lower than $\frac{1}{e^2}$. If they are close to the boundaries of the saturation regions, this means that they can be in different regions depending on the system we use. This is important because in the piecewise linear system,

equilibrium points kind depend on the region where they are located. For instance, in the partial saturation regions they are saddle points, while in the saturation ones they are attractive points.

8.3 Conclusions

We have compared two dynamical systems with different output functions, the piecewise linear and the hyperbolic one. Both systems seem to be topologically equivalent and so the hyperbolic one has been used many times instead of the original because it is differentiable along the plane. From the results obtained, we have seen that in general this is not true. There are some particular regions in the parameter space where they exhibit a different number of equilibrium points, so for theoretical results the hyperbolic tangent should not be used instead of the piecewise linear function. Furthermore, for future applications of the CNN different limit cycles, this result could be relevant because both systems exhibit a different dynamic behavior in some particular regions. On the other hand, there are other regions where both systems has a similar dynamic behavior. The number of equilibrium points in these regions is equal, so for the actually known practical applications it can be valid to use the hyperbolic model instead of the piecewise linear one.

Part V

Conclusions

Chapter 9

Conclusions and Future work

9.1 Conclusions

In this dissertation we have presented different aspects related to the two neuron CNN. Partial conclusions are given at the end of each Chapter, however we now summarize the main results obtained and outline the future work. First of all, we have studied the system stability from two different points of view, using the Lyapunov theory and doing a local equilibrium points analysis. From both methods, we have seen that in a symmetric parameter range $p_+ = p_-$, the system converges to a fixed-point. Moreover, for $s > 1$, fixed-point is located in a saturation region (out region) and so output values are restricted to ± 1. This is important because many applications such as image processing, classification problems or reproduce probability distributions use these bi-valuated output states.

In a non symmetric parameter range, the CNN system has basically two different dynamic behavior: converge to a fixed-point or converge to limit cycles. At this point, we divide the rest of the CNN study into two main parts. In the first one we focus our efforts on some of the different problems and applications for the CNN system converging to a fixed-point. In the second, we reproduce a similar scheme but in the case where the CNN system convergence to limit cycles.

The Lyapunov function has demonstrated to be an effective approach to tackle the dependence on initial conditions problem. We have found the parameter range where the system must work in order to converge to a final output without any dependence, using the Lyapunov function as a quadratic form of the outputs (y_0, y_1) while working inside the unit square.

The dynamic behavior of the entire CNN system is defined basically by its initial conditions, its inputs, and the cell interconnect (weights). Due to

the CNN system architecture, there are two different choices for the CNN inputs: the external ones u_i or the internal ones at the beginning of the process $x_i(0)$. We have seen that using the internal ones, we can reproduce Bernoulli probability distributions on the output variables derived by the Lyapunov function geometry. Yet, commonly external inputs are usually chosen to find CNN applications like reproduce linear functions, Boolean functions or solve classification problems.

We have studied the CNN system convergence in the two dimensional case constructing a map relating the external inputs $\{u_0, u_1\}$, the final outputs $\{y_0, y_1\}$ and the parameters of the CNN cloning template. This result let us design templates performing an specific functional task, and let us understand the convergence problem. Nevertheless, we have found that there are some restrictions on the CNN parameters such as the symmetry of the cloning template, $p_+ = p_- = p$. Furthermore, from this study we have also seen that not all the CNN parameters plays a direct role in the geometry of the convergence map. From six parameters $\tau = (I, b_0, b_+, b_-, p, s)$ needed to design a symmetric CNN template, only four are significant: I, b_+, b_-, s. Parameter b_0 is only a scalar factor, and parameter s bigger than one plays an indirect role on the network output.

Usually, there are two methods by which to select a template or weights. Training the processor using back-propagation or genetic algorithms does not work in this particular CNN case, so weights has to be determined off line. Convergence map has given us a recipe to define templates performing an specific task based on its geometry: lines inclines and intersection points.

To solve classification problems, sometimes a single template is not enough, so the composition of different templates gives us the key point to tackle this problem. We have seen that one template shall drive the system to one of the four possible outputs of the system

$$\mathcal{S} = \{(1,1), (-1,1), (1,-1), (-1,-1)\}.$$

Using these points as new inputs, a second template can be found in order to obtain the desired outputs designing a new convergence map. The convergence map will be useful in order to design template libraries to solve different classification problems. Moreover, we have seen that the problems that can be solved using the convergence map, are slightly more complex than the linearly separable ones. In fact, we may speak of piecewise-linear separable problems.

From the convergence map, we have found CNN parameters performing an input-output functional relation. Due to the fact that the final states set is discrete $\mathcal{S}$, this correspondence can be thought as a classification problem

where each one of the different classes is defined by the different final states, which will depend on the parameters of the CNN system. Given four subsets, they can be classified into one, two, three or four different classes. From this analysis we have established which input-output combinations are possible and which are not. A a new recipe to design CNN templates performing this task depending on only three parameters b_0, p, s has been found, and also the minimum number of necessary templates to solve a particular classification problem.

However, like other simple neural networks, functions that can be implemented are limited. We have seen that that a two neuron CNN is able to reproduce any Boolean function defined as $F(u_0, u_1) = y_i$, even the non-linear function XOR by composing two different templates. We may say then that using these functions, a two neuron CNN is a universal machine. This universal property is missing when we try to reproduce Boolean functions of the form $F(u_0, u_1) = (y_0, y_1)$.

Moreover, we have tried to reproduce the header action of a universal Turing machine without success. On higher dimensions, CNN processors has been one of the simplest realization of Conway's Game of Life and Wolfram's Rule 110, known universal Turing machine. For further investigations, one may ask which should be the minimal number of neurons needed in a CNN to be a universal Turing machine.

On the other hand, complex dynamics of Cellular Neural Networks let us show all forms of stable dynamical behavior. They may converge to a fixed-point or a limit cycle or evolve along a chaotic trajectory. In the autonomous two neuron CNN case, there are no chaotic trajectories yet there exist different limit cycles. We have done an exhaustive study relating CNN parameter values to dynamical behavior. A systematic classification of this dynamics has been studied, and we have found sufficient conditions for the existence of limit cycles in the general case.

Six different kinds of limit cycles had been found depending on the phase plane regions where they pass across. This result may prove useful on classification problems. Each limit cycle kind is related to a region in parameter space. This let us design CNNs whose outputs are only limit cycles, choosing the cloning template parameters adequately. Furthermore, adding points $\mathcal{S}$ where the system converge in the complete stable cases, generic CNN can show up to ten possible outputs namely six limit cycles and four fixed points. Classification problems with ten or less classes might be solved by mapping a general CNN parameter space to each one of these closed curves. However, a careful analysis on initial conditions would be required. Moreover, the existence and kind of limit cycles in the non autonomous case should also be studied.

Periodic behavior related with limit cycles, and the relation between limit cycles period and CNN parameters, is an interesting subject to study. For limit cycles around the central region (unit square), in the antisymmetric case, we have found the period formula depending on CNN parameters. We may think then on the possible uses of this result for further investigations just like for example, use limit cycles to reproduce a clock signal, or use it to store individual memories. We have started to study the problem to reproduce clock signals for a particular parameter range where the limit cycle period can be computed.

Finally, we have compared the CNN dynamic behavior using different output functions, hyperbolic tangent and piecewise linear function. Many times in the literature, hyperbolic tangent is used instead of piecewise linear function because of its differentiability along the plane. In some particular regions in the parameter space, they exhibit a different number of equilibrium points. Hence, for theoretical results, hyperbolic tangent should not be used instead of piecewise linear function. A priori, this result seems to be irrelevant because working in certain parameter range, both systems exhibit similar dynamic behavior. Nevertheless, for future limit cycles applications, this result could be relevant because both systems exhibit different convergence sets in some particular regions.

9.2 Final discussion: The present and the future

In this theoretical dissertation, a detailed study of the two neuron Cellular Neural Network has been attempted. Initially, we would not expect singular results from the analysis of such a simple system, since, for instance, the use of constant inputs forbids the presence of chaotic behaviors. However, the system has pleasantly shown a rich set of features so as to exhaust the present thesis.

A Cellular Neural Network is a biologically inspired system where computation emerges from a collection of simple nonlinear locally coupled cells [20]. The system complexity lies on its non linearity but also on the large number of freedom of freedom. The study of a simple case, such as the two neuron one, sheds the focus on set one of those aspects, the non linearity, and allows the application of analytical techniques. The results should contain the essence of the system nonlinearity and points towards the dues to understanding of the full real system. As such, this thesis is a piece of a large program, a ground state of first step. In this case we are not expecting to

provide a large number of applications out of the thesis results. We want to completely explore and understand two neuron Cellular Neural Networks.

Mathematics behind the CNN system is one of the main tools we have investigated. They establish the fundamentals of the network dynamic behavior. Moreover, they gives us the basis to find some possible CNN applications. For instance, the convergence map obtained from the stability analysis and the B-transformation, offers an alternative way to find cloning templates, while having an input-output functional relation or a classification problem. Other interesting applications derived from limit cycles, are the possibility to generate clock signals with a wide range of frequencies, or use them to increase the number of classification sets. These results may be useful for practical applications or electronics devices yet we haven't found them explicitly.

On these grounds, future work should be the generalization of some results obtained either to one dimensional CNN systems with n cells or to higher dimensional Cellular Neural Networks. The idea of solving the CNN system in each region where it is linear, can also be used on higher dimensions. In the symmetric case, a similar Lyapunov function can be found. Diagonalize this system and scrutinize it in each of these regions, can show up different local dynamic behaviors. Stability study on this way, could then be extended. Another main idea that can be used in a a general case, is the B-transformation. CNN system analysis is simplified if we group its external influence. Therefore, the convergence map may be useful for a similar study either on higher dimensions or on non constant inputs system. The consequences of this idea will let us tackle different problems like reproducing Boolean functions or solve classification problems. We have seen that not every Boolean function can be performed, nor any classification problem can be solved by a two neuron CNN. Study these applications on higher dimensions can show up the characteristic limitations of CNNs like for example study the universality problem. Yet one must be very careful on universal definitions, a deep study on this grounds would also be really interesting in order to describe intrinsic CNN properties.

On the other hand, limit cycles study using external inputs is a concrete problem that can be extended. In this study, they have been found only for the autonomous case. If external inputs are different from zero, general dynamic behavior of the CNN system should be almost the same because they do not change equilibrium points kind but only its position on the plane. This should be done in order to completely close the limit cycles classification problem. Moreover, limit cycles applications like reproducing clock signals or use it as an individual memory is a very interesting subject to study on further investigations. A deeper study of period function in order to

locate its minima position, and map the period sensitivity against parameter variations should be done.

Nevertheless, some other results may not be extended because they are only valid in the plane. As an example, the way we have found limit cycles using the idea of Poincaré-Bendixson theorem does not hold on higher dimensions. We should then think another way to demonstrate limit cycles existence yet the essence of this work should help to find them.

In addition, the two neuron CNN study is not completely closed yet. The system response using non constant external inputs, can open new problems to take on as a challenge. Chaotic behavior may be expected and all the chaos consequences [77]. General CNN cases, CNNs with functional external inputs, or CNN limits from a universal point of view, are attractive candidates for future work. We hope the results explained in this thesis to be useful in order to guide us traveling along CNNs roads.

"*Roads go ever ever on,*
under cloud and under star,
yet feet that wandering have gone,
turn at last to home afar"
J.R.R.Tolkien

Part VI

Appendix

Appendix A

Convergence map

The Lyapunov function obtained while studying CNN stability has let us construct a map relating the CNN the external inputs, the outputs, and weight parameters. Working with variables u', which concentrate the external influence of each neuron, we have found a simple model of this map. Via the B-transformation, the original convergence map in the u-plane can be found by mapping each one of the simple model principal elements. These principal elements are basically two intersection points P, and five boundary lines (Figure 4.3). There are four convergence regions limited by two or three of these boundary lines.

In this section we are going to study the consistence convergence regions in u-plane. To do it, we will use B-transformation (2.3) defined as $T(u_0, u_1) = (u'_0, u'_1)$ in order to study the intersection points preimage $T^{-1}(P') = P$. For sake of clarity we have used notation $R(i)$, $i = 1, 2, 3, 4$ to denote the different convergence regions $L(i, j)$, for $(i, j) \in \mathcal{S}$. These points $(\pm 1, \pm 1)$ are renamed as $(+1, +1) \equiv 1$, $(+1, -1) \equiv 2$, $(-1, -1) \equiv 3$, and $(-1, +1) \equiv 4$.

Convergence regions for $p < 0$

Convergence region $R(1) = L(+1, +1)$:

$$T^{-1}: \quad \mathbb{R}^2_{\{u'_0,u'_1\}} \quad \rightarrow \quad \mathbb{R}^2_{\{u_0,u_1\}}$$

$$\begin{cases} u'_0 \geq -p \\ u'_1 \geq -p \\ u'_0 + u'_1 \geq 0 \end{cases} \qquad \begin{cases} b_0 u_0 + b_+ u_1 + I \geq -p \\ b_- u_0 + b_0 u_1 + I \geq -p \\ (b_0 + b_-)u_0 + (b_+ + b_0)u_1 + 2I \geq 0 \end{cases}$$

To study the intersection point position in u-plane, we intersect boundary lines $b_0 u_0 + b_+ u_1 + I = -p$ and $b_- u_0 + b_0 u_1 + I = -p$ obtaining

$\left(-\frac{(I+p)(b_0-b_+)}{detB}, -\frac{(I+p)(b_0-b_-)}{detB}\right)$.

For this correspondence to be consistent, this point should be inside region defined by $(b_0 + b_-)u_0 + (b_+ + b_0)u_1 + 2I \geq 00$ just like the correspondent one in $\{u'_0, u'_1\}$-plane.

$$(b_0+b_-)\left(-\frac{(I+p)(b_0-b_+)}{detB}\right)+(b_++b_0)\left(-\frac{(I+p)(b_0-b_-)}{detB}\right)+2I=$$

$$=-\frac{(I+p)}{detB}(2b_0^2-2b_+b_-)+2I=-2(I+p)+2I=-2p>0$$

because we work with a negative parameter p.

Convergence region $R(3) = L(-1,-1)$**:** A similar study can be done in this case. Intersection point obtained from boundary lines $b_0u_0+b_+u_1+I = p$ and $b_-u_0 + b_0u_1 + I = -p$, must lay on region defined by $(b_0 + b_-)u_0 + (b_+ + b_0)u_1 + 2I \leq 0$.

$$\begin{array}{ccc} T^{-1} & \mathbb{R}^2_{\{u'_0,u'_1\}} & \rightarrow \\ & \begin{cases} u'_0 \leq p \\ u'_1 \leq p \\ u'_0 + u'_1 \leq 0 \end{cases} & \end{array} \quad \begin{array}{c} \mathbb{R}^2_{\{u_0,u_1\}} \\ \begin{cases} b_0u_0 + b_+u_1 + I \leq p \\ b_-u_0 + b_0u_1 + I \leq p \\ (b_0 + b_-)u_0 + (b_+ + b_0)u_1 + 2I \leq 0 \end{cases} \end{array}$$

$$(b_0+b_-)\left(-\frac{(p-I)(b_+-b_0)}{detB}\right)+(b_++b_0)\left(-\frac{(p-I)(b_--b_0)}{detB}\right)+2I=$$

$$=-\frac{(p-I)}{detB}(2b_+b_--2b_0^2)+2I=2(p-I)+2I=2p<0$$

because $p < 0$.

To continue studying the consistence of the convergence regions, let us tackle now this problem by studying some points position. Taking an interior point inside one region in u'-plane, via using the B-transformation, we find its preimage. The point obtained, should be inside the correspondent region in the u-plane. From the boundary lines conditions, we can check this consistence for any point.

For instance, let us take point $(-p+1, -p+1) \in L(+1,+1)$ in u'-plane. Its image $T(-p+1, -p+1)$ must be inside the corresponding convergence region $R(1)$ in u-plane.

$$\begin{pmatrix} -p+1 \\ -p+1 \end{pmatrix} = \begin{pmatrix} b_0 & b_+ \\ b_- & b_0 \end{pmatrix} \begin{pmatrix} x_{(-p+1,-p+1)} \\ y_{(-p+1,-p+1)} \end{pmatrix} + \begin{pmatrix} I \\ I \end{pmatrix}$$

$$\begin{pmatrix} x_{(-p+1,-p+1)} \\ y_{(-p+1,-p+1)} \end{pmatrix} = \frac{p+I-1}{\det B} \begin{pmatrix} b_+ - b_0 \\ b_- - b_0 \end{pmatrix}$$

To see the convergence map consistence, we will see that $T(-p+1, -p+1)$ fulfill the necessary conditions to be placed inside convergence region $R(1)$.

1. $u'_0 + u'_1 \geq 0$,

$$(b_0 + b_-)u_0 + (b_+ + b_0)u_1 + 2I =$$

$$\frac{1}{\det B}(b_0+b_-)(b_+-b_0)(I+p-1)+\frac{1}{\det B}(b_++b_0)(b_--b_0)(I+p-1)+2I =$$

$$= -2p + 2 \geq 0 \quad \text{because} \quad p < 0 \surd$$

2. $u'_0 \geq -p$,

$$b_0 u_0 + b_+ u_1 + I =$$

$$\frac{b_0}{\det B}(b_+ - b_0)(p + I - 1) + \frac{b_+}{\det B}(b_- - b_0)(p + I - 1) + I =$$

$$= -p + 1 \geq -p \surd$$

3. $u'_1 \geq -p$,

$$b_- u_0 + b_0 u_1 + I =$$

$$\frac{b_-}{\det B}(b_+ - b_0)(p + I - 1) + \frac{b_0}{\det B}(b_- - b_0)(p + I - 1) + I =$$

$$= -p + 1 \geq -p \surd$$

Let us take now point $(p-1, p-1) \in L(-1, -1)$ in order to study if its image $T(p-1, p-1)$ is inside the corresponding convergence region $R(3)$.

$$\begin{pmatrix} p-1 \\ p-1 \end{pmatrix} = \begin{pmatrix} b_0 & b_+ \\ b_- & b_0 \end{pmatrix} \begin{pmatrix} x_{(p-1,p-1)} \\ y_{(p-1,p-1)} \end{pmatrix} + \begin{pmatrix} I \\ I \end{pmatrix}$$

$$\begin{pmatrix} x_{(p-1,p-1)} \\ y_{(p-1,p-1)} \end{pmatrix} = \frac{p-I-1}{\det B} \begin{pmatrix} b_0 - b_+ \\ b_0 - b_- \end{pmatrix}$$

1. $u'_0 + u'_1 \leq 0$,

$$(b_0 + b_-)u_0 + (b_+ + b_0)u_1 + 2I =$$

$$\frac{1}{\det B}(b_0+b_-)(b_0-b_+)(p-I-1)+\frac{1}{\det B}(b_++b_0)(b_0-b_-)(p-I-1)+2I =$$

$$= 2p - 2 \leq 0 \quad \text{because} \quad p < 0 \surd$$

2. $u_0' \leq p$,

$$b_0 u_0 + b_+ u_1 + I =$$

$$\frac{b_0}{\det B}(b_0 - b_+)(p - I - 1) + \frac{b_+}{\det B}(b_0 - b_-)(p - I - 1) + I =$$

$$= p - 1 \leq p \surd$$

3. $u_1' \leq p$,

$$b_- u_0 + b_0 u_1 + I =$$

$$\frac{b_-}{\det B}(b_0 - b_+)(p - I - 1) + \frac{b_0}{\det B}(b_0 - b_-)(p - I - 1) + I =$$

$$= p - 1 \leq p \surd$$

Let us take point $(p, -p) \in L(-1, +1)$ in order to study if its image $T(p, -p)$ is inside the new convergence region $R(4)$ in u-plane.

$$\begin{pmatrix} p \\ -p \end{pmatrix} = \begin{pmatrix} b_0 & b_+ \\ b_- & b_0 \end{pmatrix} \begin{pmatrix} x_{(p,-p)} \\ y_{(p,-p)} \end{pmatrix} + \begin{pmatrix} I \\ I \end{pmatrix}$$

$$\begin{pmatrix} x_{(p,-p)} \\ y_{(p,-p)} \end{pmatrix} = \frac{1}{\det B} \begin{pmatrix} p(b_+ + b_0) + I(b_+ - b_0) \\ -p(b_- + b_0) + I(b_- - b_0) \end{pmatrix}$$

Now let us see the consistence:

1. $u_0' - u_1' \leq 0$,

$$(b_0 - b_-)u_0 + (b_+ - b_0)u_1 =$$

$$= \frac{(b_0 - b_-)}{\det B}\left(p(b_+ - b_0) + I(b_+ - b_0)\right) + \frac{(b_+ - b_0)}{\det B}\left(-p(b_- + b_0) + I(b_- - b_0)\right) =$$

$$= 2p \leq 0 \quad \text{because} \quad p < 0 \surd$$

2. $u_0' \leq -p$,

$$b_0 u_0 + b_+ u_1 + I =$$

$$= \frac{b_0}{\det B}\left(p(b_+ + b_0) + I(b_+ - b_0)\right) + \frac{b_+}{\det B}\left(-p(b_- + b_0) + I(b_- - b_0)\right) + I =$$

$$= p \leq -p \quad \text{because} \quad p < 0 \surd$$

3. $u_1' \geq p$,

$$b_- u_0 + b_0 u_1 + I =$$

$$\frac{b_-}{\det B}\left(p(b_+ + b_0) + I(b_+ - b_0)\right) + \frac{b_0}{\det B}\left(-p(b_- + b_0) + I(b_- - b_0)\right) + I =$$

$$= -p \geq p \surd$$

Finally, let us take point $(-p,p) \in L(+1,-1)$, to proceed as we have explained before in order to see the convergence map consistence.

$$\begin{pmatrix} -p \\ p \end{pmatrix} = \begin{pmatrix} b_0 & b_+ \\ b_- & b_0 \end{pmatrix} \begin{pmatrix} x_{(-p,p))} \\ y_{(-p,p)} \end{pmatrix} + \begin{pmatrix} I \\ I \end{pmatrix}$$

$$\begin{pmatrix} x_{(-p,p))} \\ y_{(-p,p)} \end{pmatrix} = \frac{1}{\det B} \begin{pmatrix} b_0 & -b_+ \\ -b_- & b_0 \end{pmatrix} \begin{pmatrix} -p-I \\ p-I \end{pmatrix} =$$

$$= \frac{1}{\det B} \begin{pmatrix} -p(b_+ + b_0) + I(b_+ - b_0) \\ p(b_- + b_0) + I(b_- - b_0) \end{pmatrix}$$

Now let us see the consistence:

1. $u_0' - u_1' \geq 0$,

$$(b_0 - b_-)u_0 + (b_+ - b_0)u_1 =$$

$$= \frac{(b_0 - b_-)}{\det B}(-p(b_+ - b_0) + I(b_+ - b_0)) + \frac{(b_+ - b_0)}{\det B}(p(b_- + b_0) + I(b_- - b_0)) =$$

$$= -2p \geq 0 \quad \text{because} \quad p < 0 \surd$$

2. $u_0' \leq p$,

$$b_0 u_0 + b_+ u_1 + I =$$

$$= \frac{b_0}{\det B}(-p(b_+ + b_0) + I(b_+ - b_0)) + \frac{b_+}{\det B}(p(b_- + b_0) + I(b_- - b_0)) + I =$$

$$= -p \geq p \quad \text{because} \quad p < 0 \surd$$

3. $u_1' \leq -p$,

$$b_- u_0 + b_0 u_1 + I =$$

$$\frac{b_-}{\det B}(-p(b_+ + b_0) + I(b_+ - b_0)) + \frac{b_0}{\det B}(p(b_- + b_0) + I(b_- - b_0)) + I =$$

$$= p \leq -p \surd$$

Convergence regions for $p > 0$.

Convergence region $R(4) = L(-1,+1)$**:** Inside convergence region $L(-1,+1)$, let us see if the intersection point between lines $b_0 u_0 + b_+ u_1 + I = -p$ and $b_- u_0 + b_0 u_1 + I = p$ is inside the convergence region $R(4)$ defined by $(b_0 - b_-)u_0 + (b_+ - b_0)u_1 \leq 0$. Intersection point of the first two lines is:

$$(x,y) = \left(\frac{-b_0(p+I) - b_+(p-I)}{detB}, \frac{b_0(p-I) + b_-(p+I)}{\det B} \right)$$

$$(b_0 - b_-)\frac{-b_0(p+I) - b_+(p-I)}{detB} + (b_+ - b_0)\frac{b_0(p-I) + b_-(p+I)}{\det B} =$$
$$= \frac{(p+I)(-b_0^2 + b_0 b_- + b_- b_+ - b_0 b_-) + (p-I)(-b_0 b_+ + b_+ b_- - b_0^2 + b_0 b_+)}{\det B} =$$
$$= \frac{-2p \det B}{\det B} = -2p \leq 0$$

because $p > 0$.

Convergence region $R(2) = L(+1, -1)$**:** Inside convergence region $L(+1, -1)$, let us study again if the intersection point of $b_0 u_0 + b_+ u_1 + I = p$ and $b_- u_0 + b_0 u_1 + I = -p$ is inside the convergence region $R(2)$ defined by $(b_0 - b_-)u_0 + (b_+ - b_0)u_1 \geq 0$. Intersection point in this case is:

$$(x, y) = \left(\frac{b_0(p-I) + b_+(p+I)}{\det B}, \frac{-b_0(p+I) - b_-(p-I)}{\det B}\right)$$

$$(b_0 - b_-)\frac{b_0(p-I) + b_+(p+I)}{\det B} + (b_+ - b_0)\frac{-b_0(p+I) - b_-(p-I)}{\det B} =$$
$$= \frac{(p+I)(b_0^2 - b_0 b_+ - b_- b_+ + b_0 b_+) + (p-I)(b_0 b_+ - b_+ b_- + b_0^2 - b_0 b_+)}{\det B} =$$
$$= \frac{2p \det B}{\det B} = 2p \geq 0$$

because $p > 0$.

To continue studying the convergence regions consistence, we will take any point P inside one region in order to see where does its image $T(P)$ goes under the B-transformation. Taking point $(p, p) \in L(+1, +1)$ in u'-plane, let us study if its image $T(p, p)$ is inside region $R(1)$.

$$\begin{pmatrix} p \\ p \end{pmatrix} = \begin{pmatrix} b_0 & b_+ \\ b_- & b_0 \end{pmatrix} \begin{pmatrix} x_{(p,p)} \\ y_{(p,p)} \end{pmatrix} + \begin{pmatrix} I \\ I \end{pmatrix}$$

$$\begin{pmatrix} x_{(p,p)} \\ y_{(p,p)} \end{pmatrix} = \frac{1}{\det B} \begin{pmatrix} b_0 & -b_+ \\ -b_- & b_0 \end{pmatrix} \begin{pmatrix} p-I \\ p-I \end{pmatrix} = \frac{p-I}{\det B} \begin{pmatrix} b_0 - b_+ \\ b_0 - b_- \end{pmatrix}$$

Now let us see the consistence:

1. $(b_0 - b_-)u_0 + (b_+ - b_0)u_1 + 2I \geq 0$,

$$(b_0 + b_-)u_0 + (b_+ + b_0)u_1 + 2I =$$
$$\frac{p-I}{\det B}(b_0 + b_-)(b_0 - b_+) + \frac{p-I}{\det B}(b_+ + b_0)(b_0 - b_-) + 2I =$$
$$= 2p \geq 0 \quad \text{because} \quad p > 0 \surd$$

2. $b_0u_0 + b_+u_1 + I \geq -p$,

$$b_0u_0 + b_+u_1 + I =$$

$$\frac{b_0}{\det B}(b_0 - b_+)(p - I) + \frac{b_+}{\det B}(b_0 - b_-)(p - I) + I =$$
$$= p \geq -p\surd$$

3. $b_-u_0 + b_0u_1 + I \geq -p$,

$$b_-u_0 + b_0u_1 + I =$$

$$\frac{b_-}{\det B}(b_0 - b_+)(p - I) + \frac{b_0}{\det B}(b_0 - b_-)(p - I) + I =$$
$$= p \geq -p\surd$$

A similar study can be done with point $(-p, -p) \in L(-1, -1)$ in u'-plane. Its image $T(-p, -p)$ must be inside convergence region $R(3)$.

$$\begin{pmatrix} -p \\ -p \end{pmatrix} = \begin{pmatrix} b_0 & b_+ \\ b_- & b_0 \end{pmatrix} \begin{pmatrix} x_{(-p,-p)} \\ y_{(-p,-p)} \end{pmatrix} + \begin{pmatrix} I \\ I \end{pmatrix}$$

$$\begin{pmatrix} x_{(-p,-p)} \\ y_{(-p,-p)} \end{pmatrix} = \frac{1}{\det B} \begin{pmatrix} b_0 & -b_+ \\ -b_- & b_0 \end{pmatrix} \begin{pmatrix} -p - I \\ -p - I \end{pmatrix} = \frac{p + I}{\det B} \begin{pmatrix} b_+ - b_0 \\ b_- - b_0 \end{pmatrix}$$

Now let us see the consistence:

1. $(b_0 - b_-)u_0 + (b_+ - b_0)u_1 + 2I \leq 0$,

$$(b_0 + b_-)u_0 + (b_+ + b_0)u_1 + 2I =$$

$$\frac{p + I}{\det B}(b_0 + b_-)(b_+ - b_0) + \frac{p + I}{\det B}(b_+ + b_0)(b_- - b_0) + 2I =$$
$$= -2p \leq 0 \quad \text{because} \quad p > 0\surd$$

2. $b_0u_0 + b_+u_1 \leq p$,

$$b_0u_0 + b_+u_1 + I =$$

$$\frac{b_0}{\det B}(b_+ - b_0)(p + I) + \frac{b_+}{\det B}(b_- - b_0)(p + I) + I =$$
$$= -p \leq p\surd$$

3. $b_- u_0 + b_0 u_1 + I \leq p$,

$$b_- u_0 + b_0 u_1 + I =$$
$$\frac{b_-}{\det B}(b_+ - b_0)(p+I) + \frac{b_0}{\det B}(b_- - b_0)(p+I) + I =$$
$$= -p \leq p \surd$$

Let us take point $(-p-1, p+1) \in L(-1,+1)$ in order to check its image $T(-p-1, p+1)$. It must be inside convergence region $R(4)$.

$$\begin{pmatrix} -p-1 \\ p+1 \end{pmatrix} = \begin{pmatrix} b_0 & b_+ \\ b_- & b_0 \end{pmatrix} \begin{pmatrix} x_{(-p-1,p+1)} \\ y_{(-p-1,p+1)} \end{pmatrix} + \begin{pmatrix} I \\ I \end{pmatrix}$$
$$\begin{pmatrix} x_{(-p-1,p+1)} \\ y_{(-p-1,p+1)} \end{pmatrix} = \frac{1}{\det B} \begin{pmatrix} b_0 & -b_+ \\ -b_- & b_0 \end{pmatrix} \begin{pmatrix} -p-I-1 \\ p-I+1 \end{pmatrix} =$$
$$= \frac{1}{\det B} \begin{pmatrix} -(p+1)(b_+ + b_0) + I(b_+ - b_0) \\ (p+1)(b_- + b_0) + I(b_- - b_0) \end{pmatrix}$$

Now let us see the consistence:

1. $(b_0 - b_-)u_0 + (b_+ - b_0)u_1 \leq 0$,

$$(b_0 - b_-)u_0 + (b_+ - b_0)u_1 =$$
$$= \frac{1}{\det B}\left((p+1)[-(b_+ + b_0)(b_0 - b_-) + (b_+ - b_0)(b_- + b_0)] + I\right) =$$
$$= -2(p+1) \leq 0 \quad \text{because} \quad p > 0 \surd$$

2. $b_0 u_0 + b_+ u_1 + I \leq -p$,

$$b_0 u_0 + b_+ u_1 + I =$$
$$= \frac{b_0}{\det B}\left(-(b_+ + b_0)(p+1) + I(b_+ - b_0)\right) +$$
$$+ \frac{b_+}{\det B}\left((b_- + b_0)(p+1) + I(b_- - b_0)\right) + I =$$
$$= -p - 1 \leq -p \quad \text{because} \quad p > 0 \surd$$

3. $b_- u_0 + b_0 u_1 + I \geq p$,

$$b_- u_0 + b_0 u_1 + I =$$
$$\frac{b_-}{\det B}\left(-(b_+ + b_0)(p+1) + I(b_+ - b_0)\right) +$$
$$+ \frac{b_0}{\det B}\left((b_- + b_0)(p+1) + I(b_- - b_0)\right) + I =$$
$$= p + 1 \geq p \surd$$

Finally, let us take point $(p+1,-p-1) \in L(+1,-1)$. Its image $T(p+1,-p-1)$ must be inside convergence region $R(2)$.

$$\begin{pmatrix} p+1 \\ -p-1 \end{pmatrix} = \begin{pmatrix} b_0 & b_+ \\ b_- & b_0 \end{pmatrix} \begin{pmatrix} x_{(p+1,-p-1))} \\ y_{(p+1,-p-1)} \end{pmatrix} + \begin{pmatrix} I \\ I \end{pmatrix}$$

$$\begin{pmatrix} x_{(p+1,-p-1))} \\ y_{(p+1,-p-1)} \end{pmatrix} = \frac{1}{\det B} \begin{pmatrix} b_0 & -b_+ \\ -b_- & b_0 \end{pmatrix} \begin{pmatrix} p+1-I \\ -p-1-I \end{pmatrix} =$$

$$= \frac{1}{\det B} \begin{pmatrix} (b_+ + b_0)(p+1) + I(b_+ - b_0) \\ -(b_- + b_0)(p+1) + I(b_- - b_0) \end{pmatrix}$$

Now let us see the consistence:

1. $(b_0 - b_-)u_0 + (b_+ - b_0)u_1 \geq 0$,

$$(b_0 - b_-)u_0 + (b_+ - b_0)u_1 =$$

$$= \frac{1}{\det B}\left([(b_+ + b_0)(b_0 - b_-) - (b_+ - b_0)(b_- + b_0)](p+1) + 0I\right) =$$

$$= 2(p+1) \geq 0 \quad \text{because} \quad p > 0 \surd$$

2. $b_0 u_0 + b_+ u_1 + I \geq p$,

$$b_0 u_0 + b_+ u_1 + I =$$

$$= \frac{b_0}{\det B}\left((b_+ + b_0)(p+1) + I(b_+ - b_0)\right) +$$

$$+ \frac{b_+}{\det B}\left(-(b_- + b_0)(p+1) + I(b_- - b_0)\right) + I =$$

$$= p+1 \geq p \quad \text{because} \quad p > 0 \surd$$

3. $b_- u_0 + b_0 u_1 + I \leq -p$,

$$b_- u_0 + b_0 u_1 + I =$$

$$\frac{b_-}{\det B}\left((b_+ + b_0)(p+1) + I(b_+ - b_0)\right) +$$

$$+ \frac{b_0}{\det B}\left(-(b_- + b_0)(p+1) + I(b_- - b_0)\right) + I =$$

$$= -p-1 \leq -p \surd$$

Appendix B

Input-output relations

In this Appendix we summarize the different convergence study made for parameter $p > 0$ and $p < 0$ respectively. We have classified the different possible CNN outputs into eight different cases. Case 1 where $B(1,1) = (1,1)$, $p > 0$ has been explained before. We describe now the remaining cases. To describe the input-output relations, we use the two row notation in order to clarify the results.

At last, we summarize the different convergence sets obtained with their correspondent parameter conditions (Tables B.11, B.12). We have classified them depending on the different number of output values where a given set can converge. Some of them can be obtained directly while the rest have been found from the composition of two elements. Of course, different element compositions let us find a particular input-output relation. We have described in Table B.13 each element using one possible element combination. The rest can be seen in different tables where we list all the different element compositions.

Case 2

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & * & 4 & * \end{pmatrix} \quad p > 0$$

For a positive parameter p, if $B(1,1) = (1,-1)$ then $B(-1,-1) = (-1,1)$ for parameters

$$I = 0, \quad b_+ = 1 - b_0, \quad b_- = -1 - b_0.$$

Point $(-1,-1)$ can not converge anywhere else because there are no parameters fulfilling the equations obtained. For instance, if $B(-1,-1) = (1,1)$, parameters must fulfill equations (B.1), and so parameter I must be equal

to 1 and 0.

$$\begin{cases} b_0 + b_+ + I = 1 \\ -b_0 - b_+ + I = 1 \\ b_- + b_0 + I = -1 \\ -b_- - b_0 + I = 1 \end{cases} \tag{B.1}$$

Possible convergence outputs for $(-1, 1)$ and $(1, -1)$ are found solving the system equations obtained by $B(-1, 1) = (\pm 1, \pm 1)$ and $B(1, -1) = (\pm 1, \pm 1)$. Using the convergence map for parameter $p > 0$, we find parameter conditions obtained imposing the image point to be inside a convergence region. For instance, let us take $B(1, -1) = (-1 + 2b_0, -1 - 2b_0)$.

- $B(1,-1) \in R(1) \Leftrightarrow \begin{cases} -1 + 2b_0 > -p \\ -1 - 2b_0 > -p \\ -1 - 2b_0 > 1 - 2b_0 \end{cases} \Leftrightarrow -1 > 1 \otimes$

- $B(1,-1) \in R(2) \Leftrightarrow \begin{cases} -1 + 2b_0 > p \\ -1 - 2b_0 < -p \end{cases} \Leftrightarrow p < \min\{\pm 1 + 2b_0\} = -1 + 2b_0$

- $B(1,-1) \in R(3) \Leftrightarrow \begin{cases} -1 + 2b_0 < p \\ -1 - 2b_0 < p \\ -1 - 2b_0 < 1 - 2b_0 \end{cases} \Leftrightarrow p > \max\{-1 \pm 2b_0\}$

- $B(1,-1) \in R(4) \Leftrightarrow \begin{cases} -1 + 2b_0 < -p \\ -1 - 2b_0 > p \end{cases} \Leftrightarrow p < \min\{\pm 1 - 2b_0\} = -1 - 2b_0$

In a similar way, we study $B(-1, 1) = (1 - 2b_0, 1 + 2b_0)$.

- $B(-1,1) \in R(1) \Leftrightarrow p > \max\{-1 - 2b_0, -1 + 2b_0\}$

- $B(-1,1) \in R(2) \Leftrightarrow p < \min\{-1 - 2b_0, 1 - 2b_0\} = -1 - 2b_0$

- $B(-1,1) \in R(3) \times$

- $B(-1,1) \in R(4) \Leftrightarrow p < \min\{-1 + 2b_0, 1 + 2b_0\} = -1 + 2b_0$

The results obtained are summarized in Table B.1.

$(1,-1)$	$(-1,1)$	parameter conditions	two row notation
$R(3)$	$R(1)$	$p > \max\{-1-2b_0, -1+2b_0\}$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$
$R(2)$	$R(4)$	$p < -1+2b_0$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 2 & 4 & 4 \end{pmatrix}$
$R(4)$	$R(2)$	$p < -1-2b_0$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 4 & 2 \end{pmatrix}$

Table B.1: Case 2. Convergence study for case where $B(-1,-1) = (-1,1)$ and $p > 0, I = 0, b_+ = 1 - b_0, b_- = -1 - b_0$.

Case 3

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & * & 1 & * \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & * & 3 & * \end{pmatrix} \quad p > 0$$

If $B(1,1) = (-1,-1)$ and $p > 0$, then $B(-1,-1) = (1,1)$ for parameters,

$$I = 0, \quad b_+ = -1 - b_0, \quad b_- = -1, -b_0,$$

or $B(-1,-1) = (-1,-1)$ for parameters

$$I = -1, \quad b_+ = -b_0, \quad b_- = -b_0.$$

In the first case, points are located on a boundary line so, we translate them $(\pm 1, \pm 1) + (\varepsilon, \varepsilon)$ obtaining the same parameter values but $I = \varepsilon$. Possible convergence outputs for $(-1,1)$ and $(1,-1)$ are summarized in Tables B.2, B.3.

$(1,-1)$	$(-1,1)$	parameter conditions	two row notation
$R(3)$	$R(3)$	$p > \max\{-1-2b_0, -1+2b_0\}$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 3 & 3 & 3 \end{pmatrix}$
$R(2)$	$R(4)$	$p < -1+2b_0$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 3 & 4 \end{pmatrix}$
$R(4)$	$R(2)$	$p < -1-2b_0$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 3 & 2 \end{pmatrix}$

Table B.2: Case 3. Convergence study for case where $B(-1,-1) = (-1,-1)$ and $p > 0, I = -1, b_+ = b_- = -b_0$.

$(1,-1)$	$(-1,1)$	parameter conditions	two row notation
$R(1)$	$R(1)$	$p > \max\{-1-2b_0-\varepsilon, 1+2b_0-\varepsilon\}$, $\varepsilon > 0$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 1 & 1 \end{pmatrix}$
$R(2)$	$R(4)$	$p < \min\{1+2b_0+\varepsilon, 1+2b_0-\varepsilon\}$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix}$
$R(3)$	$R(3)$	$p > \max\{-1-2b_0+\varepsilon, 1+2b_0+\varepsilon\}$, $\varepsilon < 0$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 3 & 1 & 3 \end{pmatrix}$
$R(4)$	$R(2)$	$p < \min\{-1-2b_0+\varepsilon, -1-2b_0-\varepsilon\}$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{pmatrix}$

Table B.3: Case 3. Convergence study for case where $B(-1,-1) = (1,1)$ and $p > 0, I = \varepsilon, b_+ = b_- = -1 - b_0$.

Case 4

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & * & 2 & * \end{pmatrix} \quad p > 0$$

If $B(1,1) = (-1,1)$ and $p > 0$, then $B(-1,-1) = (1,-1)$ for parameters,

$$I = 0, \quad b_+ = -1 - b_0, \quad b_- = 1 - b_0.$$

Possible convergence outputs for $(-1,1)$ and $(1,-1)$ are summarized in Table B.4.

$(1,-1)$	$(-1,1)$	parameter conditions	two row notation
$R(1)$	$R(3)$	$p > \max\{-1-2b_0, -1+2b_0\}$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{pmatrix}$
$R(2)$	$R(4)$	$p < -1+2b_0$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 2 & 4 \end{pmatrix}$
$R(4)$	$R(2)$	$p < -1-2b_0$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 4 & 2 & 2 \end{pmatrix}$

Table B.4: Case 4. Convergence study for case where $B(-1,-1) = (1,-1)$ and $p > 0, I = 0, b_+ = -1 - b_0, b_- = 1 - b_0$.

Case 5

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & * & 1 & * \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & * & 3 & * \end{pmatrix} \quad p < 0$$

For a negative parameter p, if $B(1,1) = (1,1)$ then $B(-1,-1) = (1,1)$ for parameters

$$I = 1, \quad b_+ = -b_0, \quad b_- = -b_0,$$

and $B(-1,-1) = (-1,-1)$ for parameters

$$I = 0, \quad b_+ = 1 - b_0, \quad b_- = 1 - b_0.$$

Now we study the possible convergence outputs for $(-1,1)$ and $(1,-1)$.

$(1,-1)$	$(-1,1)$	parameter conditions	two row notation
$R(1)$	$R(1)$	$p > \max\{-1-2b_0, -1+2b_0\}$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 1 & 1 \end{pmatrix}$
$R(2)$	$R(4)$	$p < -1+2b_0$, $b_0 > 0$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 1 & 4 \end{pmatrix}$
$R(4)$	$R(2)$	$p < -1-2b_0$, $b_0 < 0$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 1 & 2 \end{pmatrix}$

Table B.5: Case 5. Convergence study for case where $B(-1,-1) = (1,1)$ and $p < 0, I = 1, b_+ = b_- = -b_0$.

$(1,-1)$	$(-1,1)$	parameter conditions	two row notation
$R(2)$	$R(4)$	$p < -1+2b_0$, $b_0 > 2$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 2 \end{pmatrix}$
$R(4)$	$R(2)$	$p < -1-2b_0$, $b_0 < 2$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 2 \end{pmatrix}$

Table B.6: Case 5. Convergence study for case where $B(-1,-1) = (-1,-1)$ and $p < 0, I = 0, b_+ = b_- = 1 - b_0$.

Case 6

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & * & 4 & * \end{pmatrix} \quad p < 0$$

If $B(1,1) = (1,-1)$ and $p < 0$, then $B(-1,-1) = (-1,1)$ for parameters

$$I = 0, \quad b_+ = 1 - b_0, \quad b_- = -1 - b_0.$$

The possible convergence outputs for $(1,-1)$ and $(-1,1)$ are:

$(1,-1)$	$(-1,1)$	parameter conditions	two row notation
$R(2)$	$R(4)$	$p < \min\{\pm(1-2b_0), 1+2b_0\}$, $b_0 > 2$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 2 & 4 & 4 \end{pmatrix}$
$R(2)$	$R(2)$	$p < \min\{1-2b_0, \pm(1+2b_0)\}$, $b_0 < 2$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 2 & 4 & 2 \end{pmatrix}$
$R(3)$	$R(1)$	$p > \max\{-1+2b_0, -1-2b_0\}$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$

Table B.7: Case 6. Convergence study for case where $B(-1,-1) = (-1,1)$ and $p < 0, I = 0, b_+ = 1 - b_0, b_- = -1 - b_0$.

Case 7

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & * & 1 & * \end{pmatrix} \quad \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & * & 3 & * \end{pmatrix} \quad p < 0$$

If $B(1,1) = (-1,-1)$ and $p < 0$, then $B(-1,-1) = (1,1)$ for parameters

$$I = 0, \quad b_+ = -1 - b_0, \quad b_- = -1 - b_0,$$

and $B(-1,-1) = (-1,-1)$ for parameters

$$I = -1, \quad b_+ = -b_0, \quad b_- = -b_0.$$

The possible convergence outputs for $(1,-1$ and $(-1,1)$ are:

$(1,-1)$	$(-1,1)$	parameter conditions	two row notation
$R(2)$	$R(4)$	$p < -1-2b_0$, $b_0 < -2$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix}$
$R(4)$	$R(2)$	$p < 1+2b_0$, $b_0 > -2$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{pmatrix}$

Table B.8: Case 7. Convergence study for case where $B(-1,-1) = (1,1)$ and $p < 0, I = 0, b_+ = b_- = -1 - b_0$.

Case 8

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & * & 2 & * \end{pmatrix} \quad p < 0$$

$(1,-1)$	$(-1,1)$	parameter conditions	two row notation
$R(2)$	$R(4)$	$p < -1 - 2b_0,\ b_0 < 2$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 3 & 4 \end{pmatrix}$
$R(4)$	$R(2)$	$p < -1 + 2b_0,\ b_0 > 2$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 3 & 2 \end{pmatrix}$
$R(3)$	$R(3)$	$p > \max\{-1 \pm 2b_0\}$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 3 & 3 & 3 \end{pmatrix}$

Table B.9: Case 7. Convergence study for case where $B(-1,-1) = (-1,-1)$ and $p < 0, I = -1, b_+ = b_- = -b_0$.

If $B(1,1) = (-1,1)$ and $p < 0$, then $B(-1,-1) = (1,-1)$ for parameters

$$I = 0, \quad b_+ = -1 - b_0, \quad b_- = 1 - b_0.$$

The possible convergence outputs for $(1,-1)$ and $(-1,1)$ are:

$(1,-1)$	$(-1,1)$	parameter conditions	two row notation
$R(2)$	$R(4)$	$p < \min\{-1 + 2b_0, 1 + 2b_0\},\ b_0 > 2$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 2 & 4 \end{pmatrix}$
$R(4)$	$R(2)$	$p < \min\{1 - 2b_0, -1 - 2b_0\},\ b_0 < 2$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 4 & 2 & 2 \end{pmatrix}$
$R(1)$	$R(1)$	$p > \max\{-1 + 2b_0, -1 - 2b_0\}$	$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 1 \end{pmatrix}$

Table B.10: Case 8. Convergence study for case where $B(-1,-1) = (1,-1)$ and $p < 0, I = 0, b_+ = -1 - b_0, b_- = 1 - b_0$.

S_1	$(aaaa)$	(1111)	(2222)	(3333)	(4444)
S_2	$(aaba)$	(1131)	(2242)	(3313)	(4424)
	$(abaa)$	(1311)	(2422)	(3133)	(4244)
	$(abbb)$	(1333)	(2444)	(3111)	(4222)
	$(aaab)$	(1113)	(2224)	(3331)	(4442)
	$(aabb)$	(1133)	(2244)	(3311)	(4422)
	$(abba)$	(1331)	(2442)	(3113)	(4224)
	$(abab)$	(1212)	(2121)	(3131)	(4141)
		(1313)	(2323)	(3232)	(4242)
		(1414)	(2424)	(3434)	(4343)
S_3	$(abac)$	(1214)	(2123)	(3234)	(4143)
		(1412)	(2321)	(3432)	(4341)
	$(abcb)$	(1232)	(2141)	(3212)	(4121)
		(1434)	(2343)	(3414)	(4323)
S_4	$(abcd)$	(1234)	(2341)	(3214)	(4123)
		(1432)	(2143)	(3412)	(4321)

Table B.11: 64 elements converging to one, two, three and four output values.

	$(abcd)$	T_i	(I, b_0, b_+, b_-, p, s)	restrictions $s > 1$
S_1	(1111)	T_1	$(1, b_0, -b_0, -b_0, p, s)$	$p < 0$, $p > \max\{-1 \pm 2b_0\}$ $p > 0$, $p > \max\{-1 \pm 2b_0\}$
	(3333)	T_2	$(-1, b_0, -b_0, -b_0, p, s)$	$p > 0$, $p > \max\{-1 \pm 2b_0\}$ $p < 0$, $p > \max\{-1 \pm 2b_0\}$
S_2	(1131)	T_3	$(\varepsilon, b_0, 1-b_0, 1-b_0, p, s)$	$p > \max \pm(2b_0 - 1) - \varepsilon$ $p > 0$, $\varepsilon > 0$
	(3313)	T_4	$(\varepsilon, b_0, -1-b_0, -1-b_0, p, s)$	$p > \max\{\pm(1 + 2b_0) + \varepsilon\}$ $p > 0$, $\varepsilon < 0$
	(2242)	T_5	$(0, b_0, 1-b_0, 1-b_0, p, s)$	$p < \min\{1 - 2b_0, \pm(1 + 2b_0)\}$ $p < 0$, $b_0 < 2$
	(1333)	T_6	$(\varepsilon, b_0, 1-b_0, 1-b_0, p, s)$	$p > \max\{\pm(1 + 2b_0) - \varepsilon\}$ $p > 0$, $\varepsilon < 0$
	(3111)	T_7	$(\varepsilon, b_0, -1-b_0, -1-b_0, p, s)$	$p > \max\{\pm(1 + 2b_0) + \varepsilon\}$ $p > 0$, $\varepsilon > 0$
	(2244)	T_8	$(0, b_0, 1-b_0, -1-b_0, p, s)$ $(0, b_0, 1-b_0, -1-b_0, p, s)$	$p > 0$, $p < -1 + 2b_0$ $p < 0$, $p < 1 - 2b_0$, $b_0 > 2$
	(4422)	T_9	$(0, b_0, -1-b_0, 1-b_0, p, s)$ $(0, b_0, -1-b_0, 1-b_0, p, s)$	$p > 0$, $p < -1 - 2b_0$ $p < 0$, $p < \min\{\pm 1 - 2b_0\}$, $b_0 < 2$
	(2442)	T_{10}	$(0, b_0, 1-b_0, -1-b_0, p, s)$	$p > 0$, $p < -1 - 2b_0$
	(4224)	T_{11}	$(0, b_0, -1-b_0, 1-b_0, p, s)$ $(0, b_0, -1-b_0, 1-b_0, p, s)$	$p > 0$, $p < -1 + 2b_0$ $p < 0$, $p < \min\{\pm 1 + 2b_0\}$, $b_0 > 2$
	(3232)	T_{12}	$(-1, b_0, -b_0, -b_0, p, s)$	$p < 0$, $p < \min\{\pm 1 - 2b_0\}$, $b_0 < 0$
	(3434)	T_{13}	$(-1, b_0, -b_0, -b_0, p, s)$	$p < 0$, $p < \min\{\pm 1 + 2b_0\}$, $b_0 > 0$
S_3	(1214)	T_{14}	$(1, b_0, -b_0, -b_0, p, s)$	$p < 0$, $p < -1 + 2b_0$, $b_0 > 0$
	(1412)	T_{15}	$(1, b_0, -b_0, -b_0, p, s)$	$p < 0$, $p < -1 - 2b_0$, $b_0 < 0$
	(3432)	T_{16}	$(-1, b_0, -b_0, -b_0, p, s)$	$p > 0$, $p < -1 - 2b_0$
	(3234)	T_{17}	$(-1, b_0, -b_0, -b_0, p, s)$	$p > 0$, $p < -1 + 2b_0$
	(4121)	T_{18}	$(0, b_0, -1-b_0, 1-b_0, p, s)$	$p < 0$, $p > \max\{-1 \pm 2b_0\}$
	(1232)	T_{19}	$(0, b_0, 1-b_0, 1-b_0, p, s)$	$p < 0$, $p < -1 - 2b_0$, $b_0 < 2$
S_4	(1432)	T_{20}	$(\varepsilon, b_0, 1-b_0, 1-b_0, p, s)$ $(0, b_0, 1-b_0, 1-b_0, p, s)$	$p > 0$, $p < \min\{-1 + 2b_0 \pm \varepsilon\}$ $p < 0$, $p < -1 - 2b_0$, $b_0 < 2$
	(3412)	T_{21}	$(\varepsilon, b_0, -1-b_0, -1-b_0, p, s)$ $(0, b_0, -1-b_0, -1-b_0, p, s)$	$p > 0$, $p < \min\{-(1 + 2b_0) \pm \varepsilon\}$ $p < 0$, $p < 1 + 2b_0$, $b_0 > -2$
	(3214)	T_{22}	$(\varepsilon, b_0, -1-b_0, -1-b_0, p, s)$ $(0, b_0, -1-b_0, -1-b_0, p, s)$	$p > 0$, $p < \min\{1 + 2b_0 \pm \varepsilon\}$ $p < 0$, $p < -1 - 2b_0$, $b_0 < -2$
	(4123)	T_{23}	$(0, b_0, -1-b_0, 1-b_0, p, s)$	$p > 0$, $p > \max\{-1 \pm 2b_0\}$
	(2341)	T_{24}	$(0, b_0, 1-b_0, -1-b_0, p, s)$ $(0, b_0, 1-b_0, 1-b_0, p, s)$	$p > 0$, $p > \max\{\pm(1 + 2b_0) + \varepsilon\}$ $p < 0$, $p > \max\{-1 \pm 2b_0\}$
	(1234)	T_{25}	$(\varepsilon, b_0, 1-b_0, 1-b_0, p, s)$	$p > 0$, $p < \min\{1 - 2b_0 \pm \varepsilon\}$

Table B.12: Input-output associations using a single template with their parameter conditions.

S_1	$(aaaa)$	(2222)	$T_{26} = T_5 \circ T_1$	(4444)	$T_{27} = T_9 \circ T_1$
S_2	$(aaba)$	(4424)	$T_{28} = T_{11} \circ T_3$		
	$(abaa)$	(1311)	$T_{29} = T_3 \circ T_{24}$	(3133)	$T_{30} = T_4 \circ T_{24}$
		(4244)	$T_{31} = T_8 \circ T_{17}$	(2422)	$T_{32} = T_5 \circ T_{24}$
	$(abbb)$	(2444)	$T_{33} = T_{10} \circ T_6$	(4222)	$T_{34} = T_{11} \circ T_3$
	$(aaab)$	(1113)	$T_{35} = T_3 \circ T_{23}$	(3331)	$T_{36} = T_4 \circ T_{23}$
		(4442)	$T_{37} = T_9 \circ T_{14}$	(2224)	$T_{38} = T_5 \circ T_{23}$
	$(aabb)$	(1133)	$T_{39} = T_{23} \circ T_8$	(3311)	$T_{40} = T_{24} \circ T_8$
	$(abba)$	(1331)	$T_{41} = T_{23} \circ T_{10}$	(3113)	$T_{42} = T_{24} \circ T_{10}$
	$(abab)$	(1212)	$T_{43} = T_{14} \circ T_{12}$	(1313)	$T_{44} = T_4 \circ T_{12}$
		(1414)	$T_{45} = T_{15} \circ T_{12}$	(2121)	$T_{46} = T_{18} \circ T_{12}$
		(2323)	$T_{47} = T_{23} \circ T_{13}$	(2424)	$T_{48} = T_{11} \circ T_{13}$
		(3131)	$T_{49} = T_3 \circ T_{12}$	(4141)	$T_{50} = T_{24} \circ T_{13}$
		(4242)	$T_{51} = T_5 \circ T_{12}$	(4343)	$T_{52} = T_{24} \circ T_{12}$
S_3	$(abac)$	(2123)	$T_{53} = T_{24} \circ T_{15}$	(2321)	$T_{54} = T_{24} \circ T_{14}$
		(4143)	$T_{55} = T_{23} \circ T_{14}$	(4341)	$T_{56} = T_{23} \circ T_{15}$
	$(abcb)$	(1434)	$T_{57} = T_{20} \circ T_{19}$	(2141)	$T_{58} = T_{14} \circ T_{24}$
		(2343)	$T_{59} = T_{16} \circ T_{18}$	(3212)	$T_{60} = T_{19} \circ T_{22}$
		(3414)	$T_{61} = T_{23} \circ T_{18}$	(4323)	$T_{62} = T_{16} \circ T_{24}$
S_4	$(abcd)$	(2143)	$T_{63} = T_{24} \circ T_{20}$	(4321)	$T_{64} = T_{20} \circ T_{24}$

Table B.13: Input-output associations using template compositions.

	$T_1 \circ T_j = (1111)$	$T_2 \circ T_j = (2222)$
	$T_j \circ T_1$	$T_j \circ T_2$
T_1	$(1111)(1111) = (1111)$	$(1111)(2222) = (1111)$
T_2	$(3333)(1111) = (3333)$	$(3333)(2222) = (3333)$
T_3	$(1131)(1111) = (1111)$	$(1131)(2222) = (1111)$
T_4	$(3313)(1111) = (3333)$	$(3313)(2222) = (3333)$
T_5	$(2242)(1111) = (2222)$	$(2242)(2222) = (2222)$
T_6	$(1333)(1111) = (1111)$	$(1333)(2222) = (3333)$
T_7	$(3111)(1111) = (3333)$	$(3111)(2222) = (1111)$
T_8	$(2244)(1111) = (2222)$	$(2244)(2222) = (2222)$
T_9	$(4422)(1111) = (4444)$	$(4422)(2222) = (4444)$
T_{10}	$(2442)(1111) = (2222)$	$(2442)(2222) = (4444)$
T_{11}	$(4224)(1111) = (4444)$	$(4224)(2222) = (2222)$
T_{12}	$(3232)(1111) = (3333)$	$(3232)(2222) = (2222)$
T_{13}	$(3434)(1111) = (3333)$	$(3434)(2222) = (4444)$
T_{14}	$(1214)(1111) = (1111)$	$(1214)(2222) = (2222)$
T_{15}	$(1412)(1111) = (1111)$	$(1412)(2222) = (4444)$
T_{16}	$(3432)(1111) = (3333)$	$(3432)(2222) = (4444)$
T_{17}	$(3234)(1111) = (3333)$	$(3234)(2222) = (2222)$
T_{18}	$(4121)(1111) = (4444)$	$(4121)(2222) = (1111)$
T_{19}	$(1232)(1111) = (1111)$	$(1232)(2222) = (2222)$
T_{20}	$(1432)(1111) = (1111)$	$(1432)(2222) = (4444)$
T_{21}	$(3412)(1111) = (3333)$	$(3412)(2222) = (4444)$
T_{22}	$(3214)(1111) = (3333)$	$(3214)(2222) = (2222)$
T_{23}	$(4123)(1111) = (4444)$	$(4123)(2222) = (1111)$
T_{24}	$(2341)(1111) = (2222)$	$(2341)(2222) = (3333)$
T_{25}	$(1234)(1111) = (1111)$	$(1234)(2222) = (2222)$

Table B.14: Composition study for T_1-element and T_2-element.

	$T_3 \circ T_j$	$T_j \circ T_3$
T_3	$(1131)(1131) = (1131)$	$(1131)(1131) = (1131)$
T_4	$(1131)(3313) = (3313)$	$(3313)(1131) = (3313)$
T_5	$(1131)(2242) = (1111)$	$(2242)(1131) = (2242)$
T_6	$(1131)(1333) = (1111)$	$(1333)(1131) = (1131)$
T_7	$(1131)(3111) = (3111)$	$(3111)(1131) = (3313)$
T_8	$(1131)(2244) = (1111)$	$(2244)(1131) = (2242)$
T_9	$(1131)(4422) = (1111)$	$(4422)(1131) = (4424)$
T_{10}	$(1131)(2442) = (1111)$	$(2442)(1131) = (2242)$
T_{11}	$(1131)(4224) = (1111)$	$(4224)(1131) = (4424)$
T_{12}	$(1131)(3232) = (3131)$	$(3232)(1131) = (3333)$
T_{13}	$(1131)(3434) = (3131)$	$(3434)(1131) = (3333)$
T_{14}	$(1131)(1214) = (1111)$	$(1214)(1131) = (1111)$
T_{15}	$(1131)(1412) = (1111)$	$(1412)(1131) = (1111)$
T_{16}	$(1131)(3432) = (3131)$	$(3432)(1131) = (3333)$
T_{17}	$(1131)(3234) = (3131)$	$(3234)(1131) = (3333)$
T_{18}	$(1131)(4121) = (1111)$	$(4121)(1131) = (4424)$
T_{19}	$(1131)(1232) = (1131)$	$(1232)(1131) = (1131)$
T_{20}	$(1131)(1432) = (1131)$	$(1432)(1131) = (1131)$
T_{21}	$(1131)(3412) = (3111)$	$(3412)(1131) = (3313)$
T_{22}	$(1131)(3214) = (3111)$	$(3214)(1131) = (3313)$
T_{23}	$(1131)(4123) = (1113)$	$(4123)(1131) = (4424)$
T_{24}	$(1131)(2341) = (1311)$	$(2341)(1131) = (2242)$
T_{25}	$(1131)(1234) = (1131)$	$(1234)(1131) = (1131)$

Table B.15: Composition study for T_3-element.

	$T_4 \circ T_j$	$T_j \circ T_4$
T_4	$(3313)(3313) = (1131)$	$(3313)(3313) = (1131)$
T_5	$(2242)(3313) = (4424)$	$(3313)(2242) = (3333)$
T_6	$(1333)(3313) = (3313)$	$(3313)(1333) = (3111)$
T_7	$(3111)(3313) = (1131)$	$(3313)(3111) = (1333)$
T_8	$(2244)(3313) = (4424)$	$(3313)(2244) = (3333)$
T_9	$(4422)(3313) = (2242)$	$(3313)(4422) = (3333)$
T_{10}	$(2442)(3313) = (4424)$	$(3313)(2442) = (3333)$
T_{11}	$(4224)(3313) = (2242)$	$(3313)(4224) = (3333)$
T_{12}	$(3232)(3313) = (3333)$	$(3313)(3232) = (1313)$
T_{13}	$(3434)(3313) = (3333)$	$(3313)(3434) = (1313)$
T_{14}	$(1214)(3313) = (1111)$	$(3313)(1214) = (3333)$
T_{15}	$(1412)(3313) = (1111)$	$(3313)(1412) = (3333)$
T_{16}	$(3432)(3313) = (3333)$	$(3313)(3432) = (1313)$
T_{17}	$(3234)(3313) = (3333)$	$(3313)(3234) = (1313)$
T_{18}	$(4121)(3313) = (2242)$	$(3313)(4121) = (3333)$
T_{19}	$(1232)(3313) = (3313)$	$(3313)(1232) = (3313)$
T_{20}	$(1432)(3313) = (3313)$	$(3313)(1432) = (3313)$
T_{21}	$(3412)(3313) = (1131)$	$(3313)(3412) = (1333)$
T_{22}	$(3214)(3313) = (1131)$	$(3313)(3214) = (1333)$
T_{23}	$(4123)(3313) = (2242)$	$(3313)(4123) = (3331)$
T_{24}	$(2341)(3313) = (4424)$	$(3313)(2341) = (3133)$
T_{25}	$(1234)(3313) = (3313)$	$(3313)(1234) = (3313)$

Table B.16: Composition study for T_4-element.

	$T_5 \circ T_j$	$T_j \circ T_5$
T_5	$(2242)(2242) = (2222)$	$(2242)(2242) = (2222)$
T_6	$(1333)(2242) = (3333)$	$(2242)(1333) = (2444)$
T_7	$(3111)(2242) = (1111)$	$(2242)(3111) = (4222)$
T_8	$(2244)(2242) = (2242)$	$(2242)(2244) = (2222)$
T_9	$(4422)(2242) = (4424)$	$(2242)(4422) = (2222)$
T_{10}	$(2442)(2242) = (4424)$	$(2242)(2442) = (2222)$
T_{11}	$(4224)(2242) = (2242)$	$(2242)(4224) = (2222)$
T_{12}	$(3232)(2242) = (2222)$	$(2242)(3232) = (4242)$
T_{13}	$(3434)(2242) = (4444)$	$(2242)(3434) = (4242)$
T_{14}	$(1214)(2242) = (2242)$	$(2242)(1214) = (2222)$
T_{15}	$(1412)(2242) = (4424)$	$(2242)(1412) = (2222)$
T_{16}	$(3432)(2242) = (4424)$	$(2242)(3432) = (4242)$
T_{17}	$(3234)(2242) = (2242)$	$(2242)(3234) = (4242)$
T_{18}	$(4121)(2242) = (1111)$	$(2242)(4121) = (2222)$
T_{19}	$(1232)(2242) = (2222)$	$(2242)(1232) = (2242)$
T_{20}	$(1432)(2242) = (4424)$	$(2242)(1432) = (2242)$
T_{21}	$(3412)(2242) = (4424)$	$(2242)(3412) = (4222)$
T_{22}	$(3214)(2242) = (2242)$	$(2242)(3214) = (4222)$
T_{23}	$(4123)(2242) = (1131)$	$(2242)(4123) = (2224)$
T_{24}	$(2341)(2242) = (3313)$	$(2242)(2341) = (2422)$
T_{25}	$(1234)(2242) = (2242)$	$(2242)(1234) = (2242)$

Table B.17: Composition study for T_5-element.

	$T_6 \circ T_j$	$T_j \circ T_6$
T_6	$(1333)(1333) = (1333)$	$(1333)(1333) = (1333)$
T_7	$(1333)(3111) = (3111)$	$(3111)(1333) = (3111)$
T_8	$(1333)(2244) = (3333)$	$(2244)(1333) = (2444)$
T_9	$(1333)(4422) = (3333)$	$(4422)(1333) = (4222)$
T_{10}	$(1333)(2442) = (3333)$	$(2442)(1333) = (2444)$
T_{11}	$(1333)(4224) = (3333)$	$(4224)(1333) = (4222)$
T_{12}	$(1333)(3232) = (3333)$	$(3232)(1333) = (3333)$
T_{13}	$(1333)(3434) = (3333)$	$(3434)(1333) = (3333)$
T_{14}	$(1333)(1214) = (1313)$	$(1214)(1333) = (1111)$
T_{15}	$(1333)(1412) = (1313)$	$(1412)(1333) = (1111)$
T_{16}	$(1333)(3432) = (3333)$	$(3432)(1333) = (3333)$
T_{17}	$(1333)(3234) = (3333)$	$(3234)(1333) = (3333)$
T_{18}	$(1333)(4121) = (3131)$	$(4121)(1333) = (4222)$
T_{19}	$(1333)(1232) = (1333)$	$(1232)(1333) = (1333)$
T_{20}	$(1333)(1432) = (1333)$	$(1432)(1333) = (1333)$
T_{21}	$(1333)(3412) = (3313)$	$(3412)(1333) = (3111)$
T_{22}	$(1333)(3214) = (3313)$	$(3214)(1333) = (3111)$
T_{23}	$(1333)(4123) = (3133)$	$(4123)(1333) = (4222)$
T_{24}	$(1333)(2341) = (3331)$	$(2341)(1333) = (2444)$
T_{25}	$(1333)(1234) = (1333)$	$(1234)(1333) = (1333)$

Table B.18: Composition study for T_6-element.

	$T_7 \circ T_j$	$T_j \circ T_7$
T_7	(3111)(3111) = (1333)	(3111)(3111) = (1333)
T_8	(3111)(2244) = (3311)	(2244)(3111) = (4222)
T_9	(3111)(4422) = (1111)	(4422)(3111) = (2444)
T_{10}	(3111)(2442) = (1111)	(2442)(3111) = (4222)
T_{11}	(3111)(4224) = (1111)	(4224)(3111) = (2444)
T_{12}	(3111)(3232) = (1111)	(3232)(3111) = (3333)
T_{13}	(3111)(3434) = (1111)	(3434)(3111) = (3333)
T_{14}	(3111)(1214) = (3131)	(1214)(3111) = (1111)
T_{15}	(3111)(1412) = (3131)	(1412)(3111) = (1111)
T_{16}	(3111)(3432) = (1111)	(3432)(3111) = (3333)
T_{17}	(3111)(3234) = (1111)	(3234)(3111) = (3333)
T_{18}	(3111)(4121) = (1313)	(4121)(3111) = (2444)
T_{19}	(3111)(1232) = (3111)	(1232)(3111) = (3111)
T_{20}	(3111)(1432) = (1131)	(1432)(3111) = (3111)
T_{21}	(3111)(3412) = (1131)	(3412)(3111) = (1333)
T_{22}	(3111)(3214) = (1131)	(3214)(3111) = (1333)
T_{23}	(3111)(4123) = (1311)	(4123)(3111) = (2444)
T_{24}	(3111)(2341) = (1113)	(2341)(3111) = (4222)
T_{25}	(3111)(1234) = (3111)	(1234)(3111) = (3111)

Table B.19: Composition study for T_7-element.

	$T_8 \circ T_j$	$T_j \circ T_8$
T_8	$(2244)(2244) = (2244)$	$(2244)(2244) = (2244)$
T_9	$(2244)(4422) = (4422)$	$(4422)(2244) = (4422)$
T_{10}	$(2244)(2442) = (2442)$	$(2442)(2244) = (4422)$
T_{11}	$(2244)(4224) = (4224)$	$(4224)(2244) = (2244)$
T_{12}	$(2244)(3232) = (4242)$	$(3232)(2244) = (2222)$
T_{13}	$(2244)(3434) = (4444)$	$(3434)(2244) = (4444)$
T_{14}	$(2244)(1214) = (2224)$	$(1214)(2244) = (2244)$
T_{15}	$(2244)(1412) = (2422)$	$(1412)(2244) = (4422)$
T_{16}	$(2244)(3432) = (4442)$	$(3432)(2244) = (4422)$
T_{17}	$(2244)(3234) = (4244)$	$(3234)(2244) = (2244)$
T_{18}	$(2244)(4121) = (4222)$	$(4121)(2244) = (1111)$
T_{19}	$(2244)(1232) = (2242)$	$(1232)(2244) = (2222)$
T_{20}	$(2244)(1432) = (2442)$	$(1432)(2244) = (4422)$
T_{21}	$(2244)(3412) = (4422)$	$(3412)(2244) = (4422)$
T_{22}	$(2244)(3214) = (4224)$	$(3214)(2244) = (2244)$
T_{23}	$(2244)(4123) = (4224)$	$(4123)(2244) = (1133)$
T_{24}	$(2244)(2341) = (2442)$	$(2341)(2244) = (3311)$
T_{25}	$(2244)(1234) = (2244)$	$(1234)(2244) = (2244)$

Table B.20: Composition study for T_8-element.

	$T_9 \circ T_j$	$T_j \circ T_9$
T_9	$(4422)(4422) = (2244)$	$(4422)(4422) = (2244)$
T_{10}	$(4422)(2442) = (4224)$	$(2442)(4422) = (2244)$
T_{11}	$(4422)(4224) = (2442)$	$(4224)(4422) = (4422)$
T_{12}	$(4422)(3232) = (2424)$	$(3232)(4422) = (2222)$
T_{13}	$(4422)(3434) = (2222)$	$(3434)(4422) = (4444)$
T_{14}	$(4422)(1214) = (4442)$	$(1214)(4422) = (4422)$
T_{15}	$(4422)(1412) = (4244)$	$(1412)(4422) = (2244)$
T_{16}	$(4422)(3432) = (2224)$	$(3432)(4422) = (2244)$
T_{17}	$(4422)(3234) = (2422)$	$(3234)(4422) = (4422)$
T_{18}	$(4422)(4121) = (2444)$	$(4121)(4422) = (1111)$
T_{19}	$(4422)(1232) = (4424)$	$(1232)(4422) = (2222)$
T_{20}	$(4422)(1432) = (4224)$	$(1432)(4422) = (4422)$
T_{21}	$(4422)(3412) = (2244)$	$(3412)(4422) = (4422)$
T_{22}	$(4422)(3214) = (2442)$	$(3214)(4422) = (4422)$
T_{23}	$(4422)(4123) = (2442)$	$(4123)(4422) = (1133)$
T_{24}	$(4422)(2341) = (4224)$	$(2341)(4422) = (1133)$
T_{25}	$(4422)(1234) = (4422)$	$(1234)(4422) = (4422)$
	$T_{10} \circ T_j$	$T_j \circ T_{10}$
T_{10}	$(2442)(2442) = (2442)$	$(2442)(2442) = (4224)$
T_{11}	$(2442)(4224) = (2442)$	$(4224)(2442) = (2442)$
T_{12}	$(2442)(3232) = (4444)$	$(3232)(2442) = (2222)$
T_{13}	$(2442)(3434) = (4242)$	$(3434)(2442) = (4444)$
T_{14}	$(2442)(1214) = (2422)$	$(1214)(2442) = (2442)$
T_{15}	$(2442)(1412) = (2224)$	$(1412)(2442) = (4224)$
T_{16}	$(2442)(3432) = (4244)$	$(3432)(2442) = (4224)$
T_{17}	$(2442)(3234) = (4442)$	$(3234)(2442) = (2442)$
T_{18}	$(2442)(4121) = (2242)$	$(4121)(2442) = (1111)$
T_{19}	$(2442)(1232) = (2444)$	$(1232)(2442) = (2222)$
T_{20}	$(2442)(1432) = (2244)$	$(1432)(2442) = (4224)$
T_{21}	$(2442)(3412) = (4224)$	$(3412)(2442) = (4224)$
T_{22}	$(2442)(3214) = (4422)$	$(3214)(2442) = (2442)$
T_{23}	$(2442)(4123) = (2244)$	$(4123)(2442) = (1331)$
T_{24}	$(2442)(2341) = (4422)$	$(2341)(2442) = (3113)$
T_{25}	$(2442)(1234) = (2442)$	$(1234)(2442) = (2442)$

Table B.21: Composition study for T_9 and T_{10}-elements.

	$T_{11} \circ T_j$	$T_j \circ T_{11}$
T_{11}	$(4224)(4224) = (4224)$	$(4224)(4224) = (4224)$
T_{12}	$(4224)(3232) = (2222)$	$(3232)(4224) = (2222)$
T_{13}	$(4224)(3434) = (2424)$	$(3434)(4224) = (4444)$
T_{14}	$(4224)(1214) = (4244)$	$(1214)(4224) = (4224)$
T_{15}	$(4224)(1412) = (4442)$	$(1412)(4224) = (2442)$
T_{16}	$(4224)(3432) = (2422)$	$(3432)(4224) = (2442)$
T_{17}	$(4224)(3234) = (2224)$	$(3234)(4224) = (4224)$
T_{18}	$(4224)(4121) = (4424)$	$(4121)(4224) = (1111)$
T_{19}	$(4224)(1232) = (4222)$	$(1232)(4224) = (2222)$
T_{20}	$(4224)(1432) = (4422)$	$(1432)(4224) = (2442)$
T_{21}	$(4224)(3412) = (2442)$	$(3412)(4224) = (2442)$
T_{22}	$(4224)(3214) = (2244)$	$(3214)(4224) = (4224)$
T_{23}	$(4224)(4123) = (4422)$	$(4123)(4224) = (3113)$
T_{24}	$(4224)(2341) = (2244)$	$(2341)(4224) = (1331)$
T_{25}	$(4224)(1234) = (4224)$	$(1234)(4224) = (4224)$
	$T_{12} \circ T_j$	$T_j \circ T_{12}$
T_{12}	$(3232)(3232) = (3232)$	$(3232)(3232) = (3232)$
T_{13}	$(3232)(3434) = (3232)$	$(3434)(3232) = (3434)$
T_{14}	$(3232)(1214) = (3232)$	$(1214)(3232) = (1212)$
T_{15}	$(3232)(1412) = (3232)$	$(1412)(3232) = (1414)$
T_{16}	$(3232)(3432) = (3232)$	$(3432)(3232) = (3434)$
T_{17}	$(3232)(3234) = (3232)$	$(3234)(3232) = (3232)$
T_{18}	$(3232)(4121) = (2323)$	$(4121)(3232) = (2121)$
T_{19}	$(3232)(1232) = (3232)$	$(1232)(3232) = (3232)$
T_{20}	$(3232)(1432) = (3232)$	$(1432)(3232) = (3434)$
T_{21}	$(3232)(3412) = (3232)$	$(3412)(3232) = (1414)$
T_{22}	$(3232)(3214) = (3232)$	$(3214)(3232) = (1212)$
T_{23}	$(3232)(4123) = (2323)$	$(4123)(3232) = (2121)$
T_{24}	$(3232)(2341) = (2323)$	$(2341)(3232) = (4343)$
T_{25}	$(3232)(1234) = (3232)$	$(1234)(3232) = (3232)$

Table B.22: Composition study for T_{11} and T_{12}-elements.

	$T_{13} \circ T_j$	$T_j \circ T_{13}$
T_{13}	$(3434)(3434) = (3434)$	$(3434)(3434) = (3434)$
T_{14}	$(3434)(1214) = (3434)$	$(1214)(3434) = (1414)$
T_{15}	$(3434)(1412) = (3434)$	$(1412)(3434) = (1212)$
T_{16}	$(3434)(3432) = (3434)$	$(3432)(3434) = (3232)$
T_{17}	$(3434)(3234) = (3434)$	$(3234)(3434) = (3434)$
T_{18}	$(3434)(4121) = (4343)$	$(4121)(3434) = (2121)$
T_{19}	$(3434)(1232) = (3434)$	$(1232)(3434) = (3232)$
T_{20}	$(3434)(1432) = (3434)$	$(1432)(3434) = (3232)$
T_{21}	$(3434)(3412) = (3434)$	$(3412)(3434) = (1212)$
T_{22}	$(3434)(3214) = (3434)$	$(3214)(3434) = (1414)$
T_{23}	$(3434)(4123) = (4343)$	$(4123)(3434) = (2323)$
T_{24}	$(3434)(2341) = (4343)$	$(2341)(3434) = (4141)$
T_{25}	$(3434)(1234) = (3434)$	$(1234)(3434) = (3434)$
	$T_{14} \circ T_j$	$T_j \circ T_{14}$
T_{14}	$(1214)(1214) = (1214)$	$(1214)(1214) = (1214)$
T_{15}	$(1214)(1412) = (1412)$	$(1412)(1214) = (1412)$
T_{16}	$(1214)(3432) = (1412)$	$(3432)(1214) = (3432)$
T_{17}	$(1214)(3234) = (1214)$	$(3234)(1214) = (3234)$
T_{18}	$(1214)(4121) = (4121)$	$(4121)(1214) = (4141)$
T_{19}	$(1214)(1232) = (1212)$	$(1232)(1214) = (1212)$
T_{20}	$(1214)(1432) = (1412)$	$(1432)(1214) = (1412)$
T_{21}	$(1214)(3412) = (1412)$	$(3412)(1214) = (3432)$
T_{22}	$(1214)(3214) = (1214)$	$(3214)(1214) = (3234)$
T_{23}	$(1214)(4123) = (4121)$	$(4123)(1214) = (4143)$
T_{24}	$(1214)(2341) = (2141)$	$(2341)(1214) = (2321)$
T_{25}	$(1214)(1234) = (1214)$	$(1234)(1214) = (1214)$

Table B.23: Composition study for T_{13} and T_{14}-element.

	$T_{15} \circ T_j$	$T_j \circ T_{15}$
T_{15}	$(1412)(1412) = (1214)$	$(1412)(1412) = (1214)$
T_{16}	$(1412)(3432) = (1214)$	$(3432)(1412) = (3234)$
T_{17}	$(1412)(3234) = (1412)$	$(3234)(1412) = (3432)$
T_{18}	$(1412)(4121) = (2141)$	$(4121)(1412) = (4141)$
T_{19}	$(1412)(1232) = (1414)$	$(1232)(1412) = (1212)$
T_{20}	$(1412)(1432) = (1214)$	$(1432)(1412) = (1214)$
T_{21}	$(1412)(3412) = (1214)$	$(3412)(1412) = (3234)$
T_{22}	$(1412)(3214) = (1412)$	$(3214)(1412) = (3432)$
T_{23}	$(1412)(4123) = (2141)$	$(4123)(1412) = (4341)$
T_{24}	$(1412)(2341) = (4121)$	$(2341)(1412) = (2123)$
T_{25}	$(1412)(1234) = (1412)$	$(1234)(1412) = (1412)$
	$T_{16} \circ T_j$	$T_j \circ T_{16}$
T_{16}	$(3432)(3432) = (3234)$	$(3432)(3432) = (3234)$
T_{17}	$(3432)(3234) = (3432)$	$(3234)(3432) = (3432)$
T_{18}	$(3432)(4121) = (2343)$	$(4121)(3432) = (2121)$
T_{19}	$(3432)(1232) = (3434)$	$(1232)(3432) = (3232)$
T_{20}	$(3432)(1432) = (3234)$	$(1432)(3432) = (3234)$
T_{21}	$(3432)(3412) = (3234)$	$(3412)(3432) = (1214)$
T_{22}	$(3432)(3214) = (3432)$	$(3214)(3432) = (1412)$
T_{23}	$(3432)(4123) = (2343)$	$(4123)(3432) = (2321)$
T_{24}	$(3432)(2341) = (4323)$	$(2341)(3432) = (4143)$
T_{25}	$(3432)(1234) = (3432)$	$(1234)(3432) = (3432)$

Table B.24: Composition study for T_{15} and T_{16}-elements.

	$T_{17} \circ T_j$	$T_j \circ T_{17}$
T_{17}	$(3234)(3234) = (3234)$	$(3234)(3234) = (3234)$
T_{18}	$(3234)(4121) = (4323)$	$(4121)(3234) = (2121)$
T_{19}	$(3234)(1232) = (3232)$	$(1232)(3234) = (3232)$
T_{20}	$(3234)(1432) = (3432)$	$(1432)(3234) = (3432)$
T_{21}	$(3234)(3412) = (3432)$	$(3412)(3234) = (1412)$
T_{22}	$(3234)(3214) = (3234)$	$(3214)(3234) = (1214)$
T_{23}	$(3234)(4123) = (4323)$	$(4123)(3234) = (2123)$
T_{24}	$(3234)(2341) = (2343)$	$(2341)(3234) = (4341)$
T_{25}	$(3234)(1234) = (3234)$	$(1234)(3234) = (3234)$
	$T_{18} \circ T_j$	$T_j \circ T_{18}$
T_{18}	$(4121)(4121) = (1414)$	$(4121)(4121) = (1414)$
T_{19}	$(4121)(1232) = (4121)$	$(1232)(4121) = (2121)$
T_{20}	$(4121)(1432) = (4121)$	$(1432)(4121) = (2141)$
T_{21}	$(4121)(3412) = (2141)$	$(3412)(4121) = (2343)$
T_{22}	$(4121)(3214) = (2142)$	$(3214)(4121) = (4323)$
T_{23}	$(4121)(4123) = (1412)$	$(4123)(4121) = (3414)$
T_{24}	$(4121)(2341) = (1214)$	$(2341)(4121) = (1232)$
T_{25}	$(4121)(1234) = (4121)$	$(1234)(4121) = (4121)$
	$T_{19} \circ T_j$	$T_j \circ T_{19}$
T_{19}	$(1232)(1232) = (1232)$	$(1232)(1232) = (1232)$
T_{20}	$(1232)(1432) = (1232)$	$(1432)(1232) = (1434)$
T_{21}	$(1232)(3412) = (3212)$	$(3412)(1232) = (3414)$
T_{22}	$(1232)(3214) = (3212)$	$(3214)(1232) = (3212)$
T_{23}	$(1232)(4123) = (2123)$	$(4123)(1232) = (4121)$
T_{24}	$(1232)(2341) = (2321)$	$(2341)(1232) = (2343)$
T_{25}	$(1232)(1234) = (1232)$	$(1234)(1232) = (1232)$

Table B.25: Composition study for T_{17},T_{18} and T_{19}-elements.

	$T_{20} \circ T_j$	$T_j \circ T_{20}$
T_{20}	$(1432)(1432) = (1234)$	$(1432)(1432) = (1234)$
T_{21}	$(1432)(3412) = (3214)$	$(3412)(1432) = (3214)$
T_{22}	$(1432)(3214) = (3412)$	$(3214)(1432) = (3412)$
T_{23}	$(1432)(4123) = (2143)$	$(4123)(1432) = (4321)$
T_{24}	$(1432)(2341) = (4321)$	$(2341)(1432) = (2143)$
T_{25}	$(1432)(1234) = (1432)$	$(1234)(1432) = (1432)$
	$T_{21} \circ T_j$	$T_j \circ T_{21}$
T_{21}	$(3412)(3412) = (1234)$	$(3412)(3412) = (1234)$
T_{22}	$(3412)(3214) = (1432)$	$(3214)(3412) = (1432)$
T_{23}	$(3412)(4123) = (2341)$	$(4123)(3412) = (2341)$
T_{24}	$(3412)(2341) = (4123)$	$(2341)(3412) = (4123)$
T_{25}	$(3412)(1234) = (3412)$	$(1234)(3412) = (3412)$
	$T_{22} \circ T_j$	$T_j \circ T_{22}$
T_{22}	$(3214)(3214) = (1234)$	$(3214)(3214) = (1234)$
T_{23}	$(3214)(4123) = (4321)$	$(4123)(3214) = (2143)$
T_{24}	$(3214)(2341) = (2143)$	$(2341)(3214) = (4321)$
T_{25}	$(3214)(1234) = (3214)$	$(1234)(3214) = (3214)$
	$T_{23} \circ T_j$	$T_j \circ T_{23}$
T_{23}	$(4123)(4123) = (3412)$	$(4123)(4123) = (3412)$
T_{24}	$(4123)(2341) = (1234)$	$(2341)(4123) = (1234)$
T_{25}	$(4123)(1234) = (4123)$	$(1234)(4123) = (4123)$
	$T_{24} \circ T_j$	$T_j \circ T_{24}$
T_{24}	$(2341)(2341) = (3412)$	$(2341)(2341) = (3412)$
T_{25}	$(2341)(1234) = (2341)$	$(1234)(2341) = (2341)$

Table B.26: Composition study for T_{20}, $_{21}$,T_{22}, T_{23} and T_{24}-elements.

Appendix C

Wolfram's 7-4 Universal Turing machine

In this section we will study all the different possible choices between colors and $\mathcal{S}$-points fulfilling state s_5 of the universal Turing machine. This particular element is the only one which can be written as a permutation and so, it is the key point in order to look for every possible color-points association. At s_5, two colors are interchanged while the other ones remain fix. There are eight different options for this choice. The first one has been already studied.

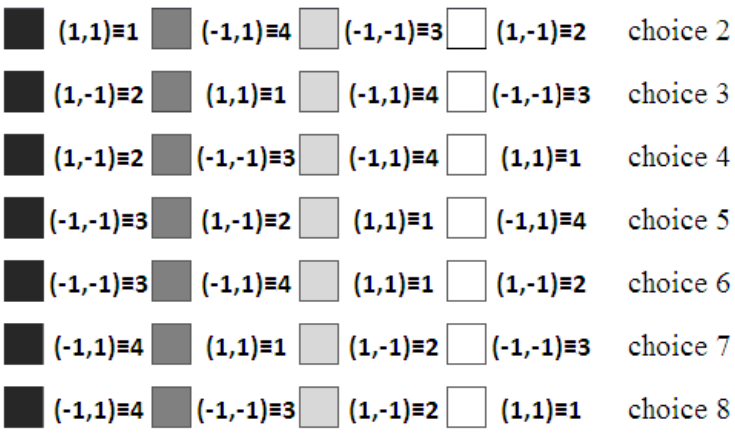


Figure C.1: Different possible choices between colors and states of the universal Turing machine.

Let us now see in each particular case, that a two neuron CNN can not reproduce all states s_i, $i = 1 \dots 7$ at once. For example state s_1 which

converges to three different output values in all the different choices, does not not belong to S_3 where elements are $(abac)$ or $(abcb)$. We can conclude then that a two neuron CNN can not reproduce the header action of a universal Turing machine.

	choice 2	choice 3	choice 4	choice 5	choice 6	choice 7	choice 8
s_1	**(4412)**	**(3112)**	**(3312)**	**(3422)**	**(3442)**	**(3411)**	**(3413)**
s_2	(3212)	(3432)	(1412)	(3414)	(3212)	(3432)	(1412)
s_3	(3414)	(1412)	(3432)	(3212)	(3414)	(1412)	(3432)
s_4	(3114)	(1422)	(2432)	(3213)	(3314)	(1442)	(4432)
s_5	(3214)	(1432)	(1432)	(3214)	(3214)	(1432)	(1432)
s_6	(3434)	(1414)	(3434)	(1212)	(1414)	(1212)	(3232)
s_7	(2432)	(3314)	(3114)	(1442)	(1422)	(3213)	(3211)

Table C.1: Universal Turing machine written as elements $(abcd)$ using every possible colors-points association.

Appendix D

Limit cycles: Antisymmetric Case

All cases $Oi, i = 1, \ldots 6$, $I1, I2$ obtained in the antisymmetric case are studied in this section, but those studied before.

O2=$\{(p_a, s) \in \mathbb{R}^2 | s > 1, p_a > s + 1\}$**.** In this case, named $O2$, equilibrium points are in regions:

	x_1^*	x_2^*	x_3^*	x_4^*	x_5^*	x_6^*	x_7^*	x_8^*
R_i	R_5	R_6	R_7	R_8	R_6	R_7	R_8	R_5

To make $x_1^* \in R_5$ and $x_5^* \in R_6$ parameters must fulfill:

$$x_5^* \in R_6 \Rightarrow p_a > s + 1; p_a > 1 - s \Rightarrow p_a > s + 1,$$

$$x_1^* \in R_5 \Rightarrow s + \frac{p_a}{s-1} > 1; p_a > s - 1 \Rightarrow p_a > s - 1.$$

Curve C_1 is defined as a circle centered on the origin with radius lower than 1. Curve C_2 is constructed using the saddle points principal directions. The symmetry of the problem let us define this curve only in two regions: R_1 and R_6, and extend the results to the rest. In region R_1, C_2 is defined as the repulsive principal direction of the saddle point x_1^*

$$C_{21} = \{(x_0, x_1) = x_1^* + \lambda(s, -p_a), \lambda \in \mathbb{R}\},$$

and in region R_6, it is defined as the line connecting C_{21} with C_{22}.

Example 1. Example for $s = 1.5$ and $p_a = 3 > s + 1$. Equilibrium points are:

$x_1^* = (-6, 19.5)$	$x_2^* = (19.5, 6)$	$x_3^* = (6, -19.5)$	$x_4^* = (-19.5, -6)$
$x_5^* = (1.5, 4.5)$	$x_6^* = (4.5, -1.5)$	$x_7^* = (-1.5, -4.5)$	$x_8^* = (-4.5, 1.5)$

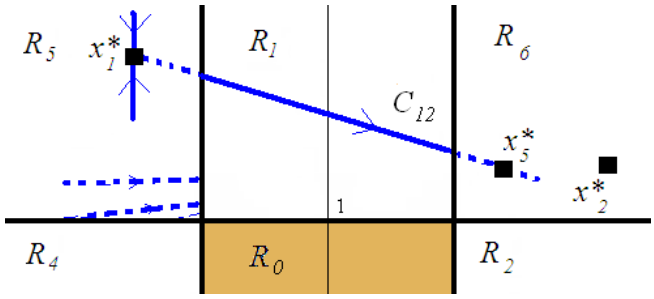


Figure D.1: Limit cycle and C_{21} construction for $s = 1.5$ and $p_a = 3$.

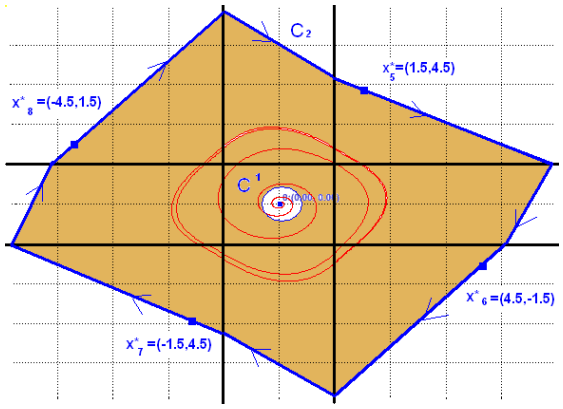


Figure D.2: C_{21} construction for $s = 1.5$ and $p_a = 3 > s + 1$

O3=$\{(p_a, s) \in \mathbb{R}^2 | s > 1, p_a < -s - 1\}$**.** In this case, named $O3$, equilibrium points positions are:

	x_1^*	x_2^*	x_3^*	x_4^*	x_5^*	x_6^*	x_7^*	x_8^*
R_i	R_6	R_7	R_8	R_5	R_8	R_5	R_6	R_7

To make $x_1^* \in R_6$ and $x_5^* \in R_8$ parameters must fulfill:

$$x_5^* \in R_8 \Rightarrow p_a < s - 1; p_a < -s - 1$$

$$x_1^* \in R_6 \Rightarrow -\frac{p_a}{s-1} > 1; s + \frac{p_a^2}{s-1} > 1 \Rightarrow p_a < -s + 1$$

Example 2. Example for $s = 2, p_a = -3.5$ with equilibrium points

$x_0^* = (0, 0)$			
$x_1^* = (3.5, 14.25)$	$x_2^* = (14.25, -3.5)$	$x_3^* = (-3.5, -14.25)$	$x_4^* = (-14.25, 3.5)$
$x_5^* = (-5.5, -1.5)$	$x_6^* = (-1.5, 5.5)$	$x_7^* = (5.5, 1.5)$	$x_8^* = (1.5, -5.5)$

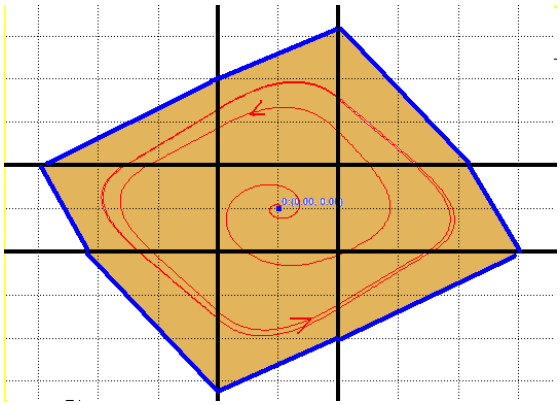


Figure D.3: C_{21} construction for parameters $s = 2$ and $p_a = -3.5$.

C_2 construction is the same as in regions O_1, O_2, using the repulsive principal directions in the saddle regions and connecting them by lines in the output regions.

O4=$\{(p_a, s) \in \mathbb{R}^2 | s > 1, |p_a + s| < 1, s(s-1) + p_a(p_a - 1) > 0\}$**.** In this case, named $O4$, equilibrium points positions are:

	x_1^*	x_2^*	x_3^*	x_4^*	x_5^*	x_6^*	x_7^*	x_8^*
R_i	R_6	R_7	R_8	R_5	R_4	R_1	R_2	R_3

To make $x_1^* \in R_6$ and $x_5^* \in R_4$ parameters must fulfill:

$$x_5^* \in R_4 \Rightarrow p_a - s < -1; p_a + s \in (-1, 1) \Rightarrow |p_a + s| < 1$$

$$x_1^* \in R_6 \Rightarrow -\frac{p_a}{s-1} > 1; s + \frac{p_a^2}{s-1} > 1 \Rightarrow p_a < -s + 1$$

Example 3. Example for $s = 2, p_a = -3.5$ with equilibrium points

$x_0^* = (0, 0)$			
$x_1^* = (2.5, 8.25)$	$x_2^* = (8.25, -2.5)$	$x_3^* = (-2.5, -8.25)$	$x_4^* = (-8.25, 2.5)$
$x_5^* = (-4.5, -0.5)$	$x_6^* = (-0.5, 4.5)$	$x_7^* = (4.5, 0.5)$	$x_8^* = (0.5, -4.5)$

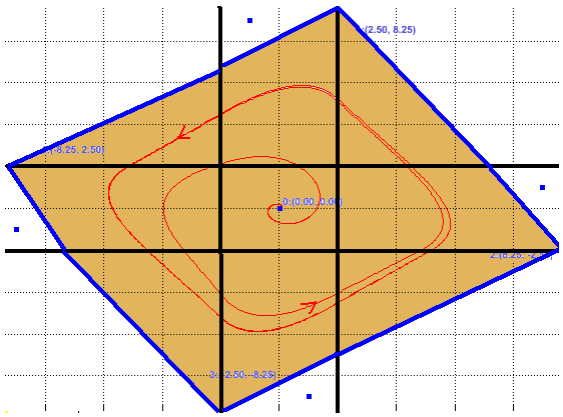


Figure D.4: C_{21} construction for parameters $s = 2$ and $p_a = -2.5$.

C_2 can be construct as we have done in the previous cases. Using saddle points and its principal directions in the inner regions, one can connect these lines with the output ones.

O5∪O6=$\{(p_a, s) \in \mathbb{R}^2 | s > 1, |p_a - s| < 1, s(s-1) + p_a(p_a - 1) < 0\} \bigcup \{(p_a, s) \in \mathbb{R}^2 | s > 1, |p$
Equilibrium points in cases named O5 and O6 are in the same regions as O1 and O4 respectively. In case O5, we can not construct C_2 in the same way as in O1 because the line passing through x_1^* with director vector $(s, -p_a)$ reaches the unit square.

To find C_2 curve limiting region D, we use the saddle points principal directions. Lines in regions $R_1, \ldots, R_4$ are the same as we have used before

$$C_{2j} = x_j^* + \lambda \vec{v}, \lambda \in \mathbb{R}, j = 1, 2, 3, 4$$

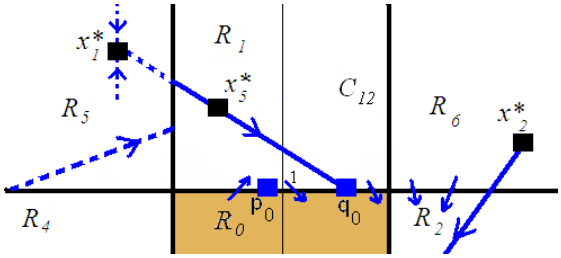


Figure D.5: C_{21} construction .

where $\vec{v}$ is the eigenvector of A-matrix. The problem now is how to find C_{25}. Let us take C_{25} as the line $x_1 = 1$ connecting C_{21} with C_{22}, and let us study the trajectory directions on it.

$$\dot{x_1} = (s-1)x_1 - p_a x_0|_{x_1=1} = (s-1) - p_a x_0 = 0 \Leftrightarrow p_0 = \frac{s-1}{p_a}.$$

Principal direction passing across x_1^* is

$$sx_1 + p_a x_0 = p_a^2 + s^2$$

Let us name q_0 the intersection point between the principal direction and line $x_1 = 1$. Let us name p_0 the point where trajectory directions change from going out from R_0 and going inside.

$$sx_1 + p_a x_0 = p_a^2 + s^2 \cap x_1 = 1 \Rightarrow q_0 = p_a + s\frac{s-1}{p_a}$$

Comparing p_0 and q_0 positions, we obtain

$$p_0 < q_0 \Leftrightarrow \frac{s-1}{p_a} < p_a + s\frac{s-1}{p_a} \Leftrightarrow 0 < p_a + \frac{(s-1)^2}{p_a}$$

Hence, $q_0 > p_0$ for $p_a > 0$, and $\dot{x_1} < 0 \Leftrightarrow x_0 > p_0$. This implies that trajectories moves under $x_1 = 1$ as can be seen in Figure D.5. A similar study can be done in the other regions. We can conclude then that there exist a limit cycle passing across R_0. An example for parameters $s = 1.1, p_a = 0.8$ can be seen in figure D.6.

Using the symmetry of the problem, curve C_2 can be constructed in a same way for case $O6$. From these results, a limit cycle passing across R_0, R_1, R_2, R_3, R_4 will exist.

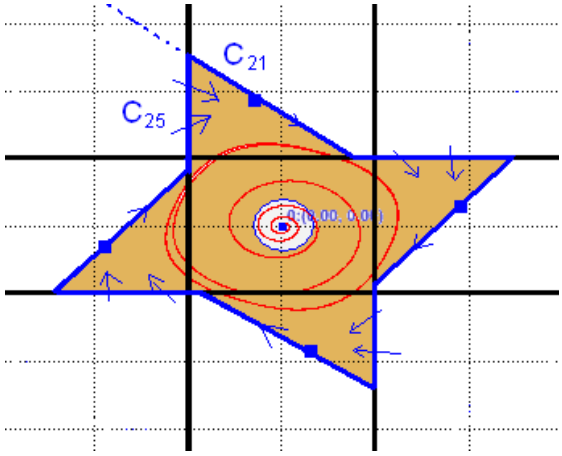


Figure D.6: Limit cycle across R_0 in case $OI1$.

I1=$\{(p_a, s) \in \mathbb{R}^2 | |p_a| < s - 1\}$**.** In this case, each equilibrium point is inside their corresponding region $x_i^* \in R_i$.

In the boundary line $x_1 = 1$ of region $R_1 = \{(x_0, x_1) \in \mathbb{R}^2; |x_0| \leq 1, x_1 > 1\}$,

$$\dot{x_1} = (s-1) - p_a x_0 \geq (s-1) + p_a > 0,$$

because $|p_a| < s - 1, x_0 \in [-1, 1]$. So $x_1(t)$ leave region R_1 and never returns to region R_0. Saddle point x_1^* is inside region R_1, so trajectories will go to regions R_5 or R_6 where equilibrium points x_5^* or x_6^* are stable nodes. Therefore, the system converge to one of these points.

In the boundary line $x_0 = 1$ of region $R_2 = \{(x_0, x_1) \in \mathbb{R}^2; |x_1| \leq 1, x_0 > 1\}$,

$$\dot{x_0} = (s-1) + p_a x_1 \geq (s-1) - p_a > 0$$

because $|p_a| < s - 1, x_1 \in [-1, 1]$. Hence, $x_0(t)$ leave region R_2 and never returns to region R_0. As $x_2^* \in R_2$ is a saddle point, trajectories will go to regions R_6 or R_7 where equilibrium points x_6^* or x_7^* are stable nodes.

Using one more time the symmetry of the problem, we can conclude that a similar dynamic behavior occurs in saddle regions R_3, R_4. The system will always converge to one of the stable equilibrium points of the output regions R_5, R_6, R_7, R_8 and so (y_0, y_1) converge to $(\pm 1, \pm 1)$.

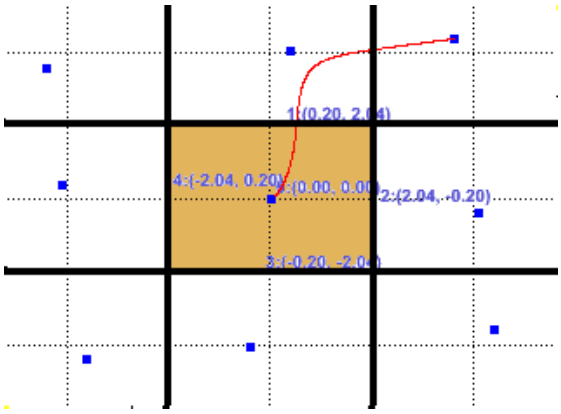


Figure D.7: Trajectory for $s = 2$ and $p_a = -0.2$.

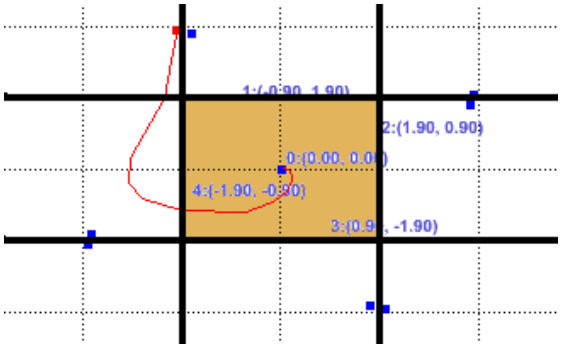


Figure D.8: Trajectory for $s = 1.5$ and $p_a = 0.45$.

Appendix E

Limit cycles: General case

Now we are going to address the general case for parameters $p_+ > 0$ and $p_- < 0$, except the cases studied before. Then, we will continue the study of the CNN dynamic behavior for parameters $p_+ < 0$ and $p_- > 0$.

O2=$\{(s, p_+, p_-) \in \mathbb{R}^3 || p_- - s| < 1, |p_+ - s| < 1\}$**.** All equilibrium points are out from their corresponding region, so there are no fixed-points for the system. Then a limit cycle will exist.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
		p_-						p_+			R_5	R_6	R_1	R_2

In order to prove it, we find curve C_2 using the repulsive principal directions of saddle points. We obtain four different cases for C_2 construction.

Parameter range map for this case depends on parameter s. If we intersect curves L_1 and L_2, we find

$$L_1 \cap L_2 \Rightarrow \begin{cases} s(s-1) + p_-(1-p_+) = 0 \\ s(s-1) - p_+(1+p_-) = 0 \end{cases} \Rightarrow p_-^2 + p_- + s(s-1) = 0$$

There exist a solution for this equation if parameter s fulfill

$$-4s^2 + 4s + 1 \geq 0 \Rightarrow \frac{1-\sqrt{2}}{2} \leq s \leq \frac{1+\sqrt{2}}{2}$$

Studying curves L_1 and L_2 we find that curve L_1 pass across $(s-1, s-1)$. It has a vertical asymptote in $p_+ = 0$, and an horizontal one in $p_- = -1$. Line L_2 pass across $(1-s, 1-s)$ and has a vertical asymptote in $p_+ = 1$ and an horizontal one in $p_- = 0$. We have basically three cases for these curves position depending on $s > 2$, $\frac{1+\sqrt{2}}{2} < s < 2$, and $1 < s < \frac{1+\sqrt{2}}{2}$.

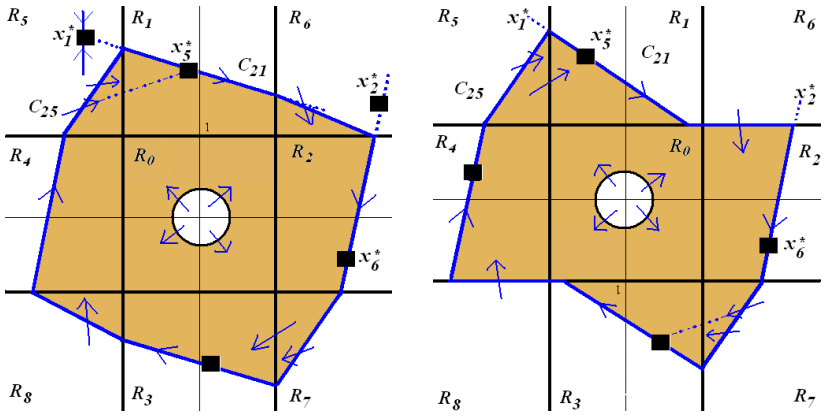


Figure E.1: O2. Case 1 : $L_1 > 0, L_2 > 0$, and Case 2 : $L_1 > 0, L_2 < 0$.

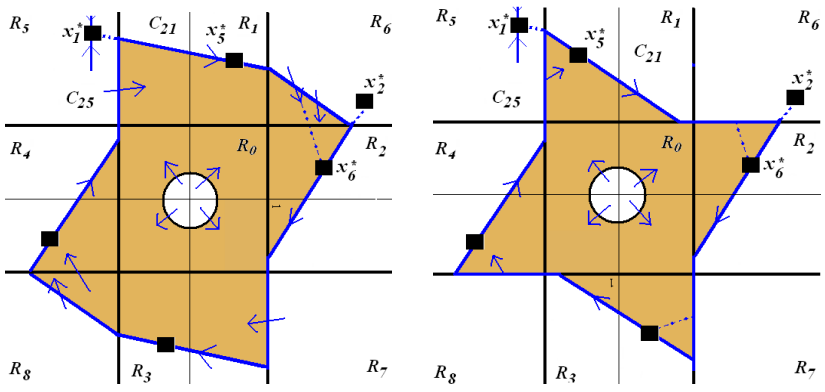


Figure E.2: O2. Case 3: $L_1 < 0, L_2 > 0$ and Case 4: $L_1 > 0, L_2 < 0$.

- For $s > 2$, we have only Case 1,
- For $\frac{1+\sqrt{2}}{2} < s < 2$, we can have Case 1, Case 2 and Case 3,

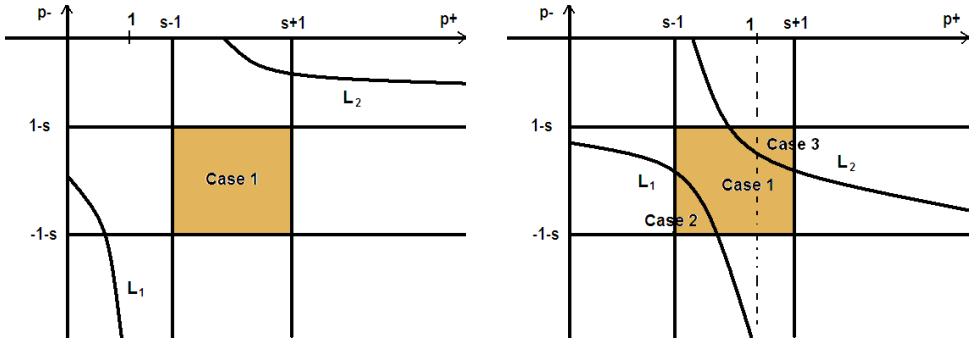


Figure E.3: Parameter range for $s > 2$ and for $\frac{1+\sqrt{2}}{2} < s < 2$.

- For $1 < s < \frac{1+\sqrt{2}}{2}$, we can have Cases 1,2,3 and Case 4.

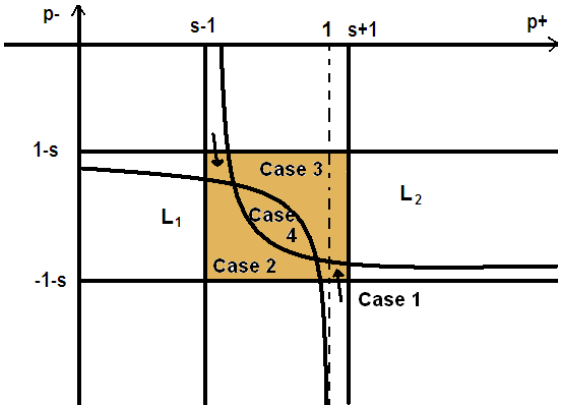


Figure E.4: Parameter range for $1 < s < \frac{1+\sqrt{2}}{2}$.

where

$$\begin{cases} \text{Case 1:} & L_1 > 0, L_2 > 0 \\ \text{Case 2:} & L_1 > 0, L_2 < 0 \\ \text{Case 3:} & L_1 < 0, L_2 > 0 \\ \text{Case 4:} & L_1 < 0, L_2 < 0 \end{cases}.$$

O3=$\{(s, p_+, p_-) \in \mathbb{R}^3 | p_- < 1 - s, p_+ > s + 1\}$**.** All equilibrium points are out from their corresponding region, so there are no fixed-points for the system. Then a limit cycle will exist. In order to prove it, we find curve C_2 using the repulsive principal directions of the saddle points.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
p_-										p_+	R_5	R_6	R_6	R_7

In this case $L_1 > 0$ and $L_2 > 0$ so there is only one option for C_2 construction.

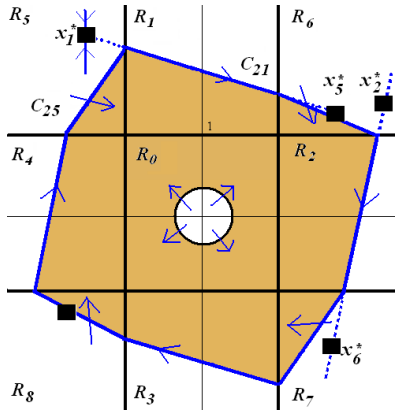


Figure E.5: O3. Case 1 : $L_1 > 0, L_2 > 0$.

Parameter range in this case does not depend on s. We can only have Case 1.

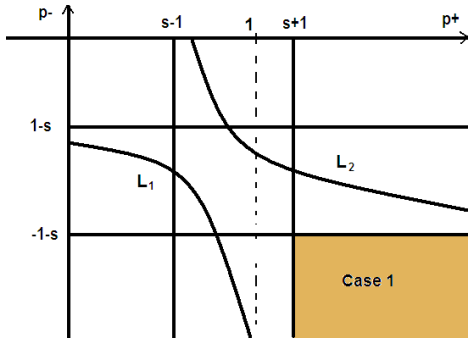


Figure E.6: O3. Parameter range for $1 < s < 2$.

O4=$\{(s, p_+, p_-) \in \mathbb{R}^3 || p_- - s| < 1, p_+ > s + 1\}$**.** All equilibrium points are out from their corresponding region, so we obtain sufficient conditions for a limit cycle to exist. To construct C_2, we must distinguish again the relative position of the repulsive principal directions of the saddle points, which depends on L_1 and L_2 sign.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
		p_-								p_+	R_5	R_6	R_6	R_2

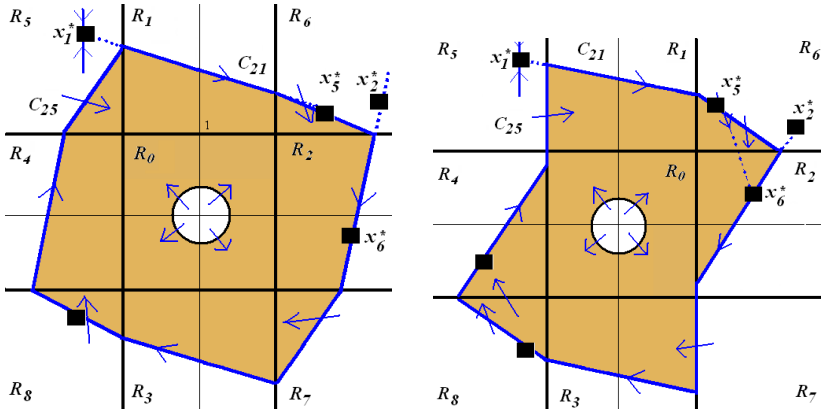


Figure E.7: O4. Case 1: $L_1 > 0, L_2 > 0$ and Case 3: $L_1 > 0, L_2 < 0$.

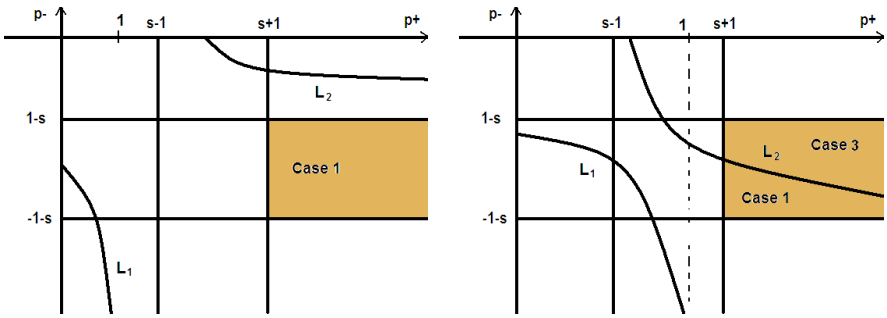


Figure E.8: O4. Parameter range for $s > 2$ and for $1 < s < 2$.

OI2=$\{(s, p_+, p_-) \in \mathbb{R}^3 | 1 - s < p_- < 0, p_+ > s + 1\}$**.** Not all the equilibrium points are out from their corresponding region, so we don't have sufficient conditions for a limit cycle to exist. There are four points inside their region, two saddle points, and two stable points which become fixed-points for the CNN system.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
				p_-						p_+	R_5	R_2	R_6	R_6

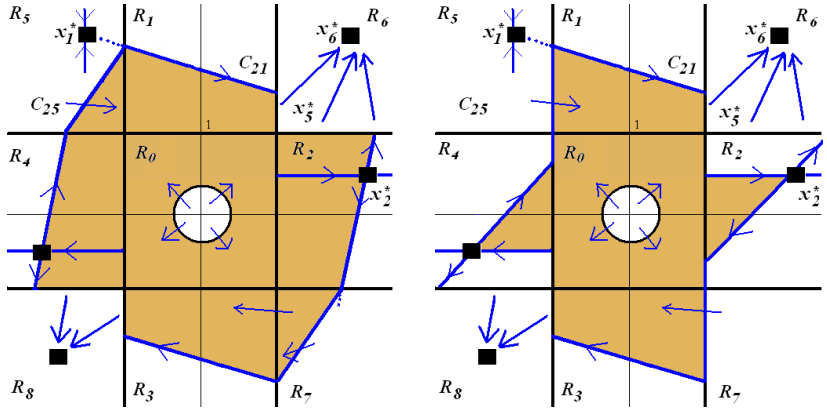


Figure E.9: OI2. Case 1: $L_1 > 0, L_2 > 0$ and Case 3: $L_1 > 0, L_2 < 0$.

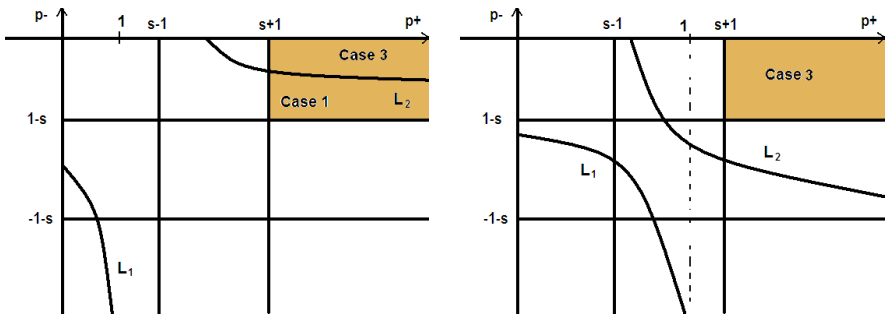


Figure E.10: OI2. Parameter range for $s > 2$ and for $1 < s < 2$.

OI3=$\{(s, p_+, p_-) \in \mathbb{R}^3 | p_- < -1 - s, 0 < p_+ < s - 1\}$**.** Four equilibrium points are inside their corresponding region so we can not assure the existence of a limit cycle.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
p_-						p_+					R_1	R_6	R_5	R_7

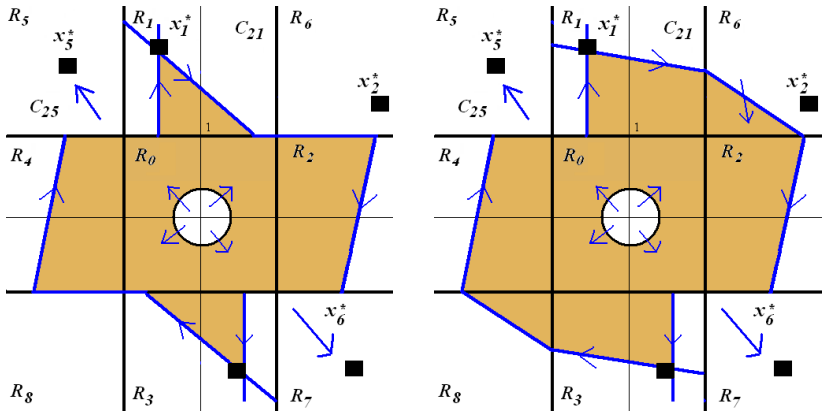


Figure E.11: OI3. Case 1: $L_1 > 0, L_2 > 0$ and Case 3: $L_1 < 0, L_2 > 0$.

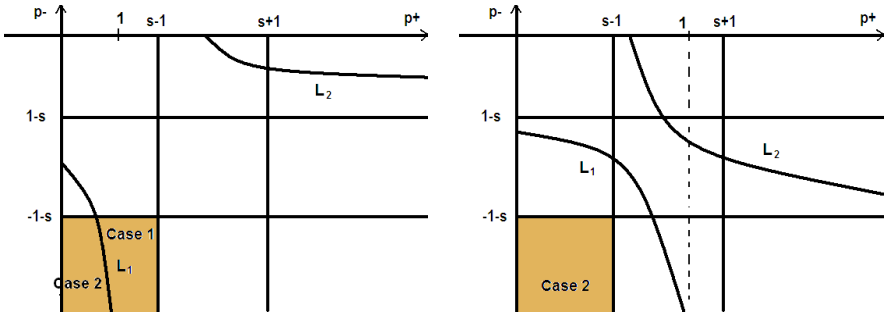


Figure E.12: OI3. Parameter range for $s > 2$ and for $1 < s < 2$.

OI4=$\{(s, p_+, p_-) \in \mathbb{R}^3 || p_- - s| < 1, 0 < p_+ < s - 1\}$**.** Four equilibrium points are inside their corresponding region so we can not assure the existence of a limit cycle.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
		p_-				p_+					R_1	R_6	R_5	R_2

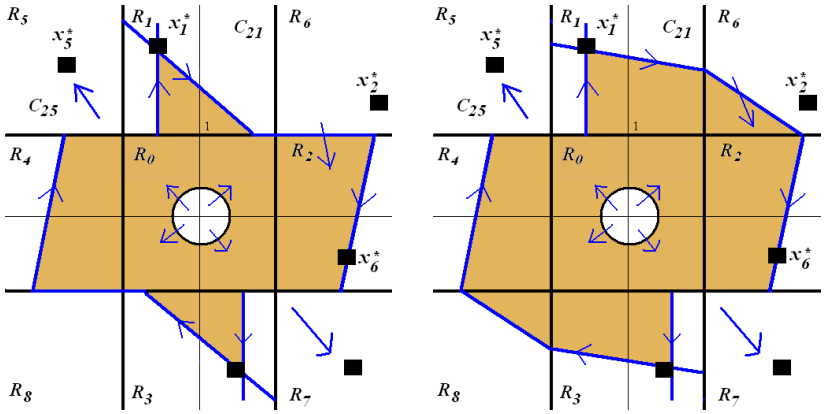


Figure E.13: OI4. Case 1: $L_1 > 0, L_2 > 0$ and Case 3: $L_1 < 0, L_2 > 0$.

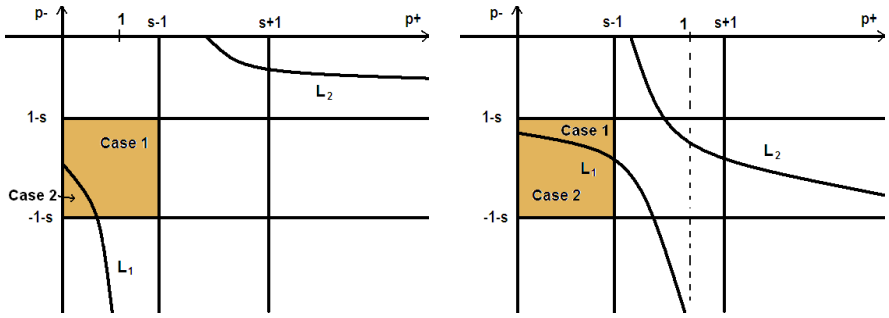


Figure E.14: OI4. Parameter range for $s > 2$ and for $1 < s < 2$.

I1∪I2=$\{(s, p_+, p_-) \in \mathbb{R}^3 | 0 < |p_+| < s-1, 0 < |p_-| < s-1, p_+p_- < 0\}$**.** All equilibrium points are inside their corresponding region, so we can not assure the existence of a limit cycle.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
				p_-		p_+					R_1	R_2	R_5	R_6

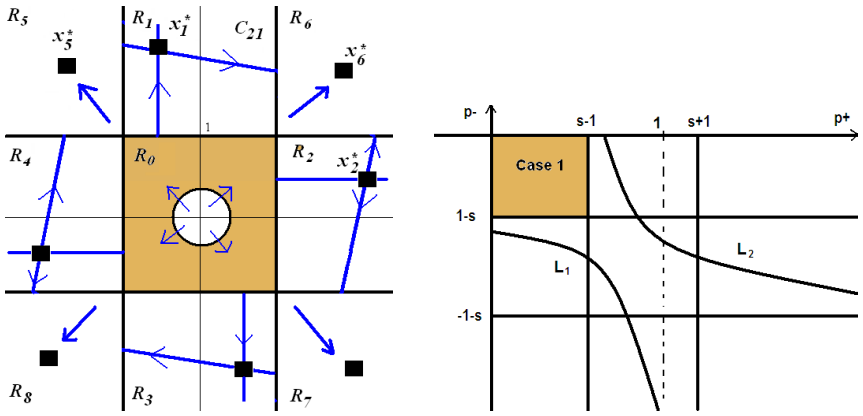


Figure E.15: I1,I2. Case 1: $L_1 > 0, L_2 > 0$ and parameter range which does not depend on s.

In fact in this case, the system will converge to one of the fixed-points because on the boundary of the unit square, trajectories pass across leaving the inner region and do not enter again.

- In $x_0 = 1$, $\dot{x_0} = -x_0 + sy_0 + p_+y_1|_{x_0=1} = s - 1 + p_+x_1 > 0$ for $|x_1| < 1$.
- In $x_1 = 1$, $\dot{x_1} = -x_1 + sy_1 + p_-y_0|_{x_1=1} = s - 1 + p_-x_> 0$ for $|x_0| < 1$.
- In $x_0 = -1$, $\dot{x_0} = -x_0 + sy_0 + p_+y_1|_{x_0=-1} = 1 - s + p_+x_1 < 0$ for $|x_1| < 1$.
- In $x_1 = -1$, $\dot{x_1} = -x_1 + sy_1 + p_-y_0|_{x_1=-1} = 1 - s + p_-x_1 < 0$ for $|x_0| < 1$.

Hence, a trajectory starting inside the unit square will leave it going to a saddle region or going to a stable one. In both cases, trajectories converge to a stable node inside its corresponding stable region.

Now we begin the study for $p_+ < 0$ and $p_- > 0$. In these cases, trajectories have opposite directions as those of the cases studied before.

In region R_1 repulsive principal direction is

$$l_1 : p_- x_0 = s x_1 - s^2 + p_+ p_-$$

Let us intersect it with line $x_1 = 1$ in order to study conditions to reach the unit square or not.

$$l_1 \bigcap x1 = 1 \Rightarrow \overline{x_0} = \frac{s(1-s) + p_+ p_-}{p_-}$$

$$\begin{cases} \overline{x_0} < -1 \Rightarrow L_3 : s(s-1) - p_-(1+p_+) > 0 \\ \overline{x_0} > -1 \Rightarrow L_3 : s(s-1) - p_-(1+p_+) < 0 \end{cases}$$

Curve L_3 pass across $(s-1, s-1)$. It has a vertical asymptote at $p_+ = -1$, and an horizontal one at $p_- = 0$.

In region R_2 repulsive principal direction is

$$l_2 : p_+ x_1 = s x_0 - s^2 + p_+ p_-$$

Let us intersect it with line $x_1 = 1$, in order to study the necessary conditions to make it reach the unit square or not.

$$l_2 \bigcap x1 = 1 \Rightarrow \overline{x_0} = \frac{s^2 + p_+ - p_+ p_-}{s}$$

$$\begin{cases} \overline{x_0} > 1 \Rightarrow L_4 : s(s-1) + p_+(1-p_-) > 0 \\ \overline{x_0} < 1 \Rightarrow L_4 : s(s-1) + p_+(1-p_-) < 0 \end{cases}$$

Curve L_4 pass across $(1-s, 1-s)$. It has a vertical asymptote at $p_+ = 0$,and an horizontal one at $p_- = 1$. These curves does not pass across the square with vertices the origin and $(1-s, s-1)$. Now we begin to study all the cases for this parameter range.

OI5=$\{(s, p_+, p_-) \in \mathbb{R}^3 | 1 - s < p_+ < 0, |p_- - s| < 1\}$**.** There are four equilibrium points inside their corresponding region, so we can not find sufficient conditions for the existence of a limit cycle. The system can converge to one of the stable nodes which are inside their region or to a limit cycle.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
				p_+				p_-			R_1	R_7	R_4	R_6

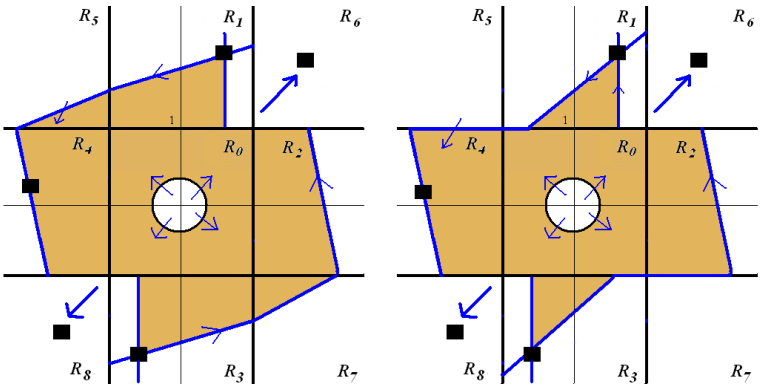


Figure E.16: OI5. Case 1: $L_3 > 0, L_4 > 0$ and Case 2: $L_3 < 0, L_4 > 0$

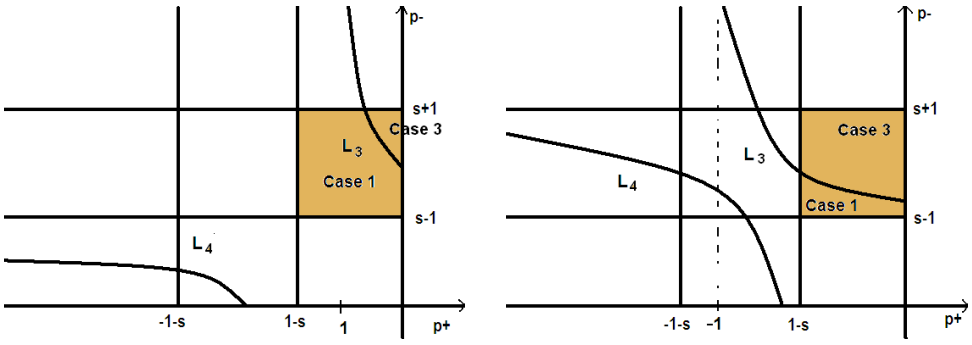


Figure E.17: OI5. Parameter range for $s > 2$ and $1 < s < 2$.

OI6=$\{(s, p_+, p_-) \in \mathbb{R}^3 | 1 - s < p_+ < 0, p_- < s + 1\}$. There are four equilibrium points inside their corresponding region, so we can not find sufficient conditions for the existence of a limit cycle. The system can converge to one of the stable nodes which are inside their region or to a limit cycle.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
				p_+						p_-	R_1	R_7	R_8	R_6

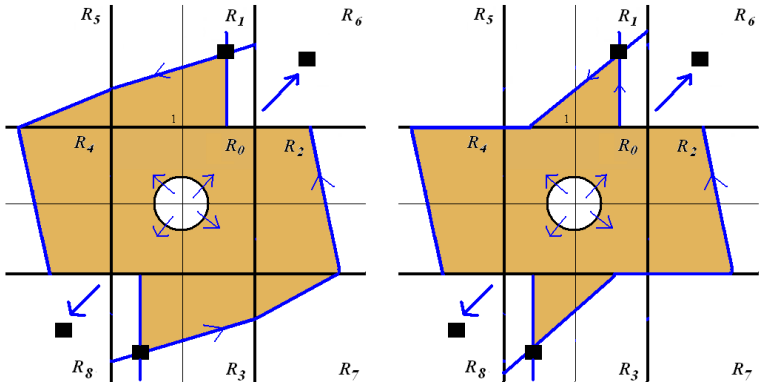


Figure E.18: OI6. Case 1: $L_3 > 0, L_4 > 0$ and Case 2: $L_3 < 0, L_4 > 0$

Parameters range map for this two different cases are:

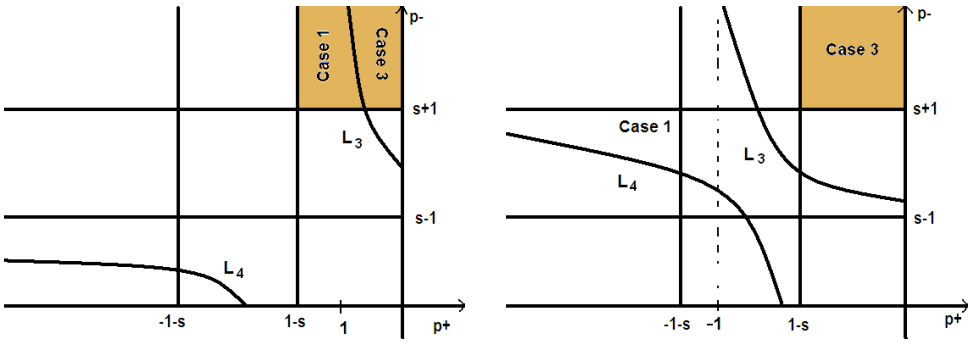


Figure E.19: OI6. Parameter range for $s > 2$ and $1 < s < 2$.

OI7=$\{(s, p_+, p_-) \in \mathbb{R}^3 || p_+ + s| < 1, 0 < p_- < s - 1\}$**.** There are four equilibrium points inside their corresponding region, so we can not find sufficient conditions for the existence of a limit cycle. The system can converge to one of the stable nodes which are inside their region or to a limit cycle.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
		p_+				p_-					R_6	R_2	R_5	R_1

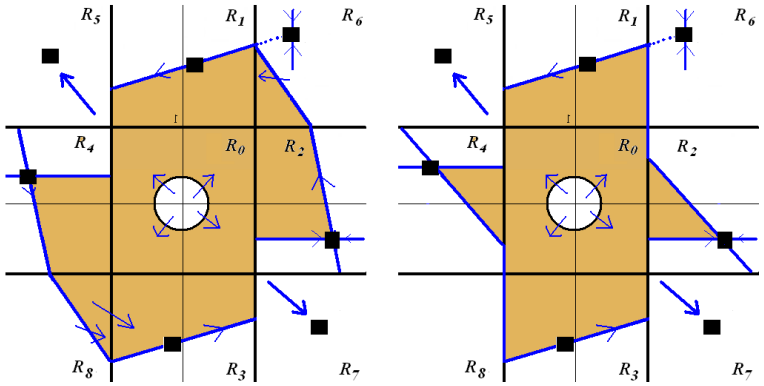


Figure E.20: OI7. Case 1: $L_3 > 0, L_4 > 0$ and Case 2: $L_3 > 0, L_4 < 0$.

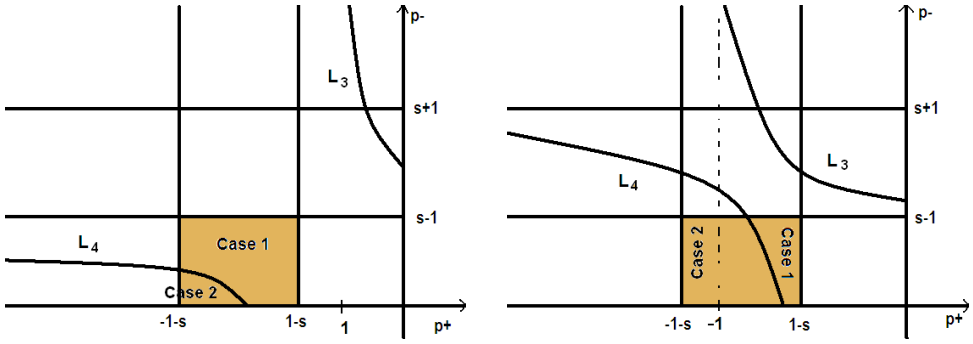


Figure E.21: OI7. Parameter range for $s > 2$ and $1 < s < 2$.

O5=$\{(s, p_+, p_-) \in \mathbb{R}^3 || p_+ + s| < 1, |p_- - s| < 1\}$**.** All equilibrium points are out form their corresponding region, so we have sufficient conditions for the existence of a limit cycle.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
		p_+						p_-			R_6	R_7	R_4	R_1

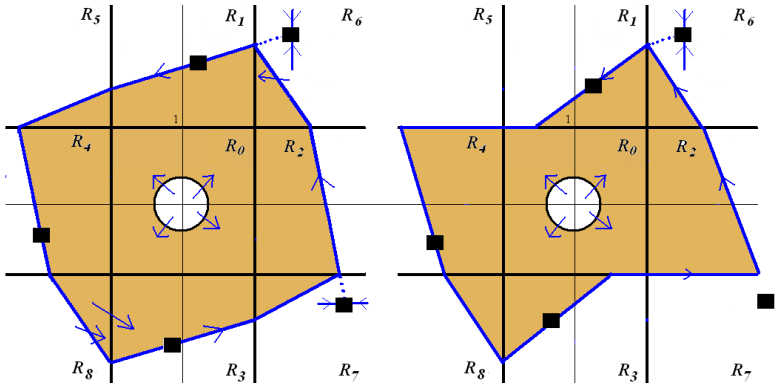


Figure E.22: O5. Case 1: $L_3 > 0, L_4 > 0$ and Case 2: $L_3 < 0, L_4 > 0$

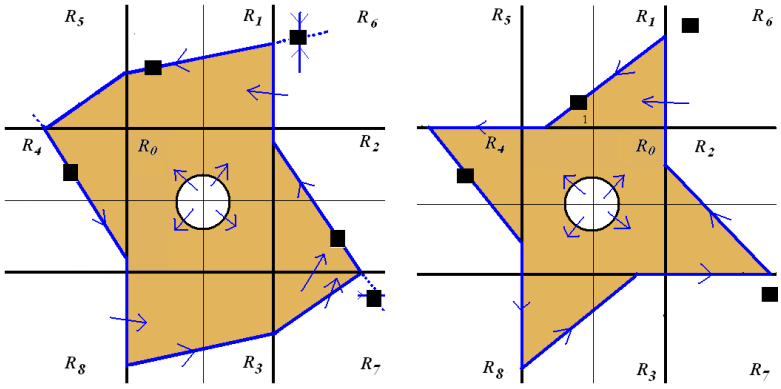


Figure E.23: O5. Case 1: $L_3 > 0, L_4 < 0$ and Case 2: $L_3 < 0, L_4 < 0$

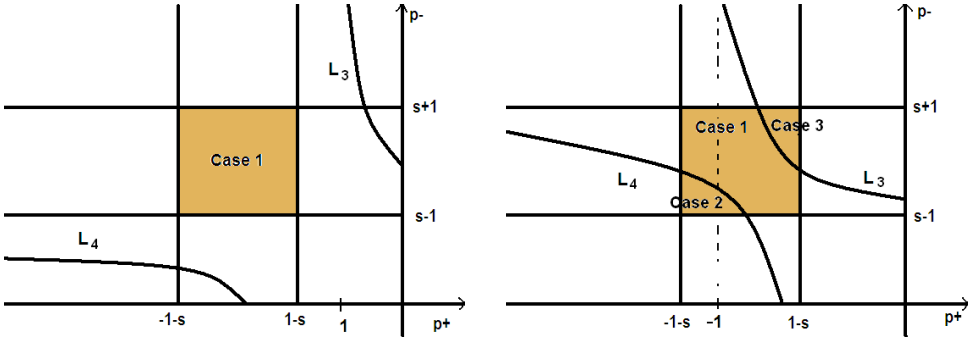


Figure E.24: O5. Parameter range for $s > 2$ and $\frac{1+\sqrt{2}}{2} < s < 2$.

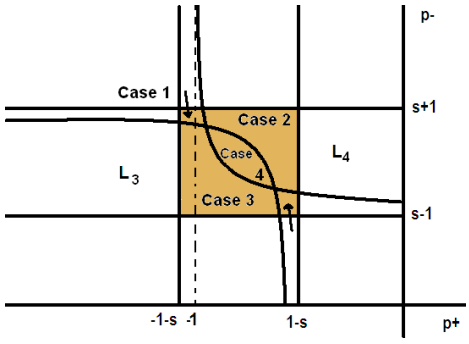


Figure E.25: O5. Parameter range for $1 < s < \frac{1+\sqrt{2}}{2}$.

O6=$\{(s, p_+, p_-) \in \mathbb{R}^3 || p_+ + s| < 1, p_- > s + 1\}$**.** All equilibrium points are out form their corresponding region, so we have sufficient conditions for the existence of a limit cycle.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
		p_+								p_-	R_6	R_7	R_8	R_1

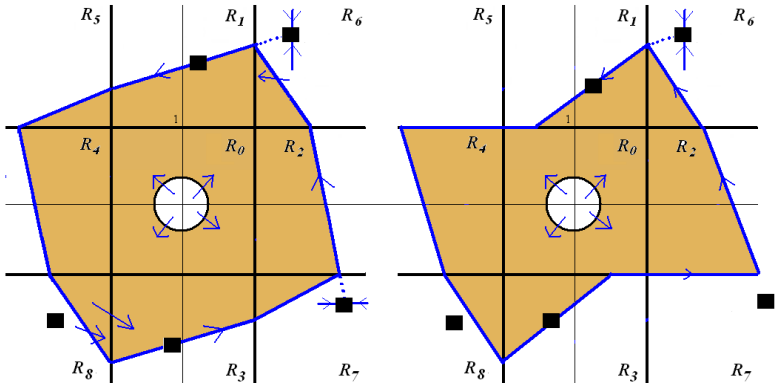


Figure E.26: O6. Case 1: $L_3 > 0, L_4 > 0$ and Case 2: $L_3 > 0, L_4 < 0$.

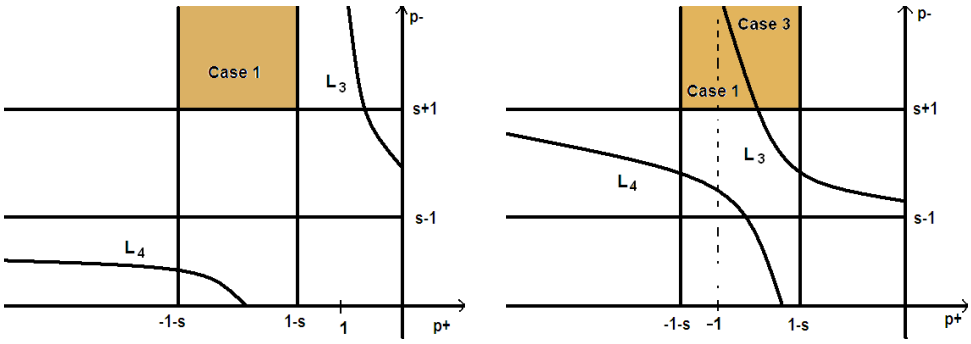


Figure E.27: O6. Parameter range for $s > 2$ and $1 < s < 2$.

OI8=$\{(s, p_+, p_-) \in \mathbb{R}^3 | p_+ < -1 - s, 0 < p_- > s - 1\}$**.** Four equilibrium points are inside their corresponding region, so we can not find sufficient conditions in order to demonstrate the existence of a limit cycle. The system can converge to one of the stable nodes or to a limit cycle.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
p_+						p_-					R_6	R_2	R_5	R_5

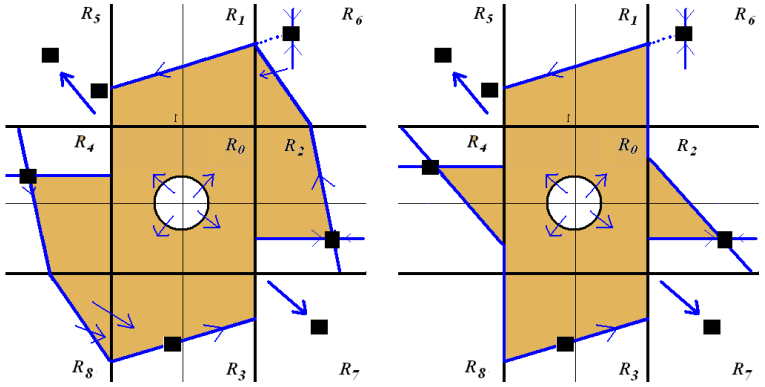


Figure E.28: OI8. Case 1: $L_3 > 0, L_4 > 0$ and Case 2: $L_3 > 0, L_4 < 0$

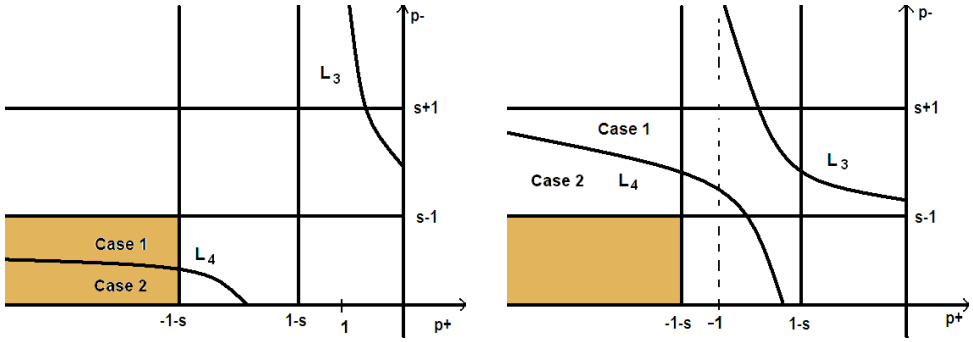


Figure E.29: OI8. Parameter range for $s > 2$ and $1 < s < 2$.

O7=$\{(s, p_+, p_-) \in \mathbb{R}^3 | p_+ < -1 - s, |p_- - s| < 1\}$**.** All equilibrium points are out form their corresponding region, so we have sufficient conditions for the existence of a limit cycle.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
p_+								p_-			R_6	R_7	R_4	R_5

Studying the repulsive principal directions of the saddle points and the equilibrium points positions, we find two possibilities for C_2 depending on the parameters of the cloning template.

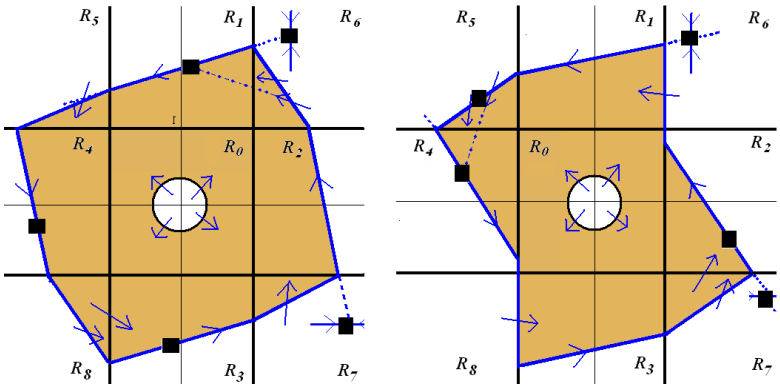


Figure E.30: O7. Case 1: $L_3 > 0, L_4 > 0$ and Case 2: $L_3 < 0, L_4 > 0$

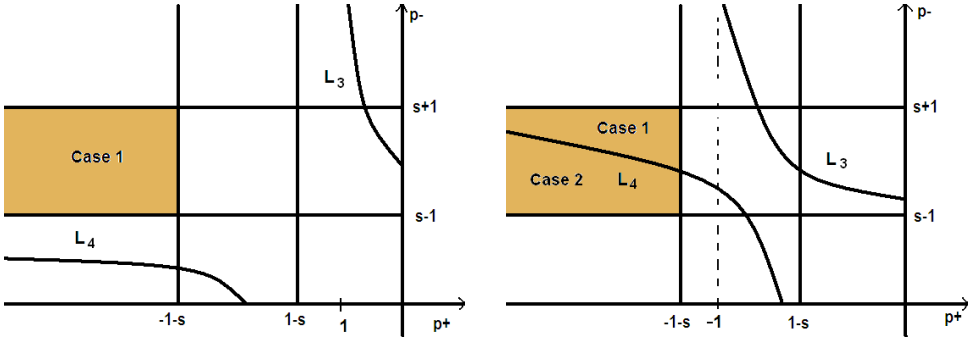


Figure E.31: O7. Parameter range for $s > 2$ and $1 < s < 2$.

O8=$\{(s, p_+, p_-) \in \mathbb{R}^3 | p_+ < -1 - s, p_- > s + 1\}$**.** All equilibrium points are out from their corresponding region, so we have sufficient conditions for the existence of a limit cycle. Parameter range in this case does not depend on s.

	$-1-s$		$1-s$		0		$s-1$		$s+1$		x_1^*	x_2^*	x_5^*	x_6^*
p_+										p_+	R_6	R_7	R_8	R_5

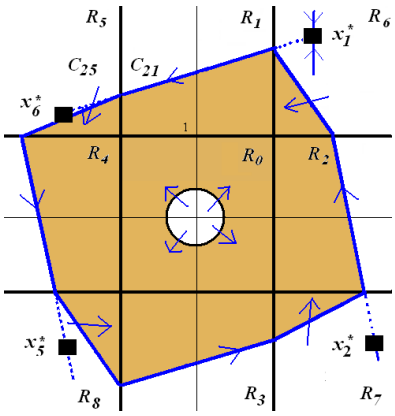


Figure E.32: O8. Case 1 : $L_3 > 0, L_4 > 0$.

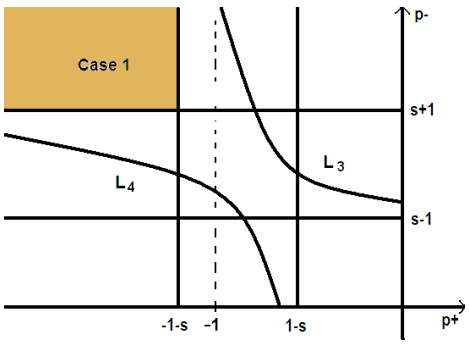


Figure E.33: O8. Parameter range for $1 < s < 2$.

From this study we conclude that in the general case, there are also three different topological distributions according to parameter s value. These different distributions depends on the existence of limit cycles, limit cycles coexisting with two fixed-points, and the converge to one of the four stable nodes. Let us note that limit cycles has certain symmetry with respect those found in case $p_+ > 0$ and $p_- < 0$ but reversing the arrows. From the results founded in this Appendix , we print a limit cycle map in Figures E.34, E.35 and E.36.

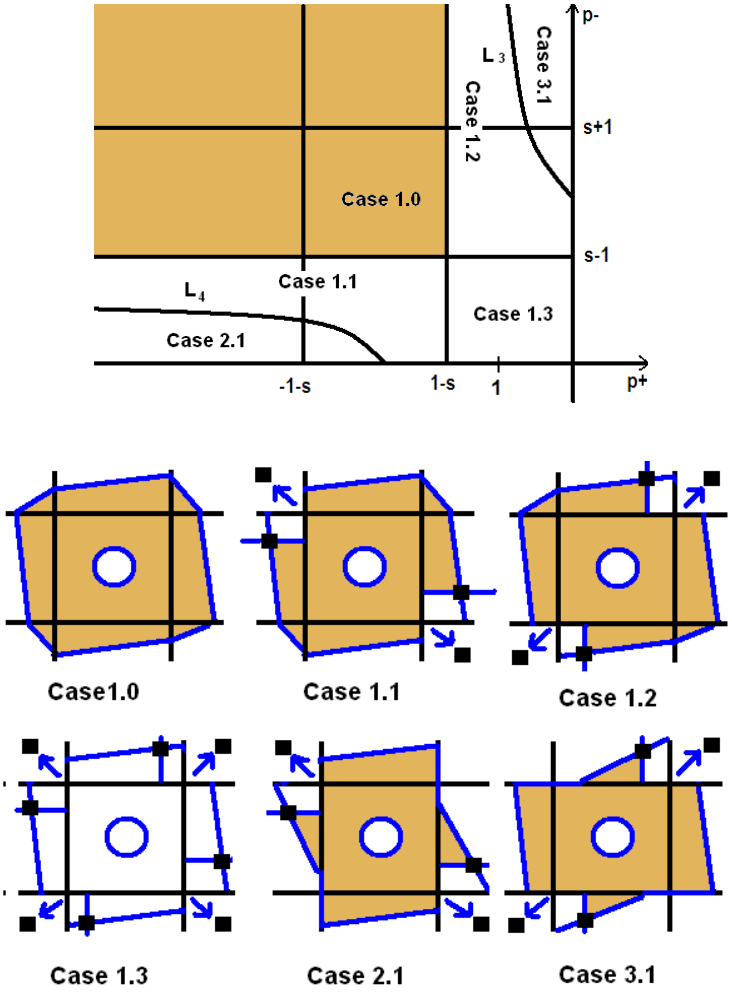


Figure E.34: C_2 construction and regions for $s > 2$.

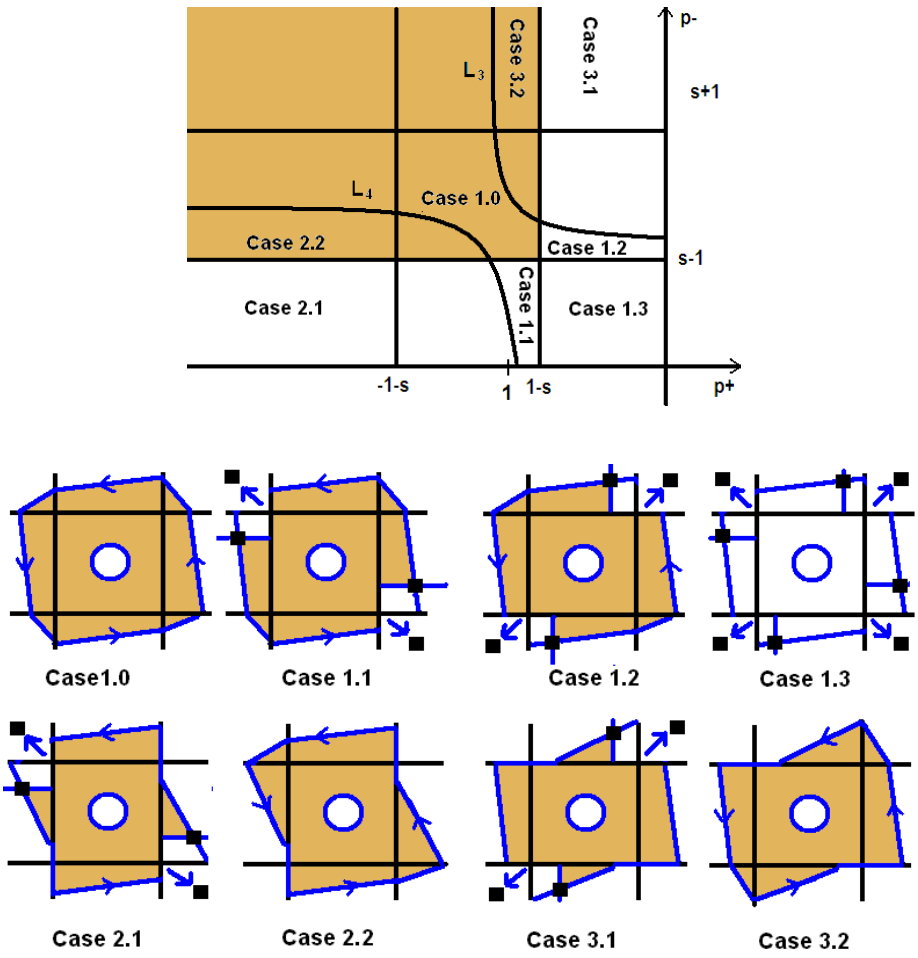


Figure E.35: C_2 construction and regions for $\frac{1+\sqrt{2}}{2} < s < 2$.

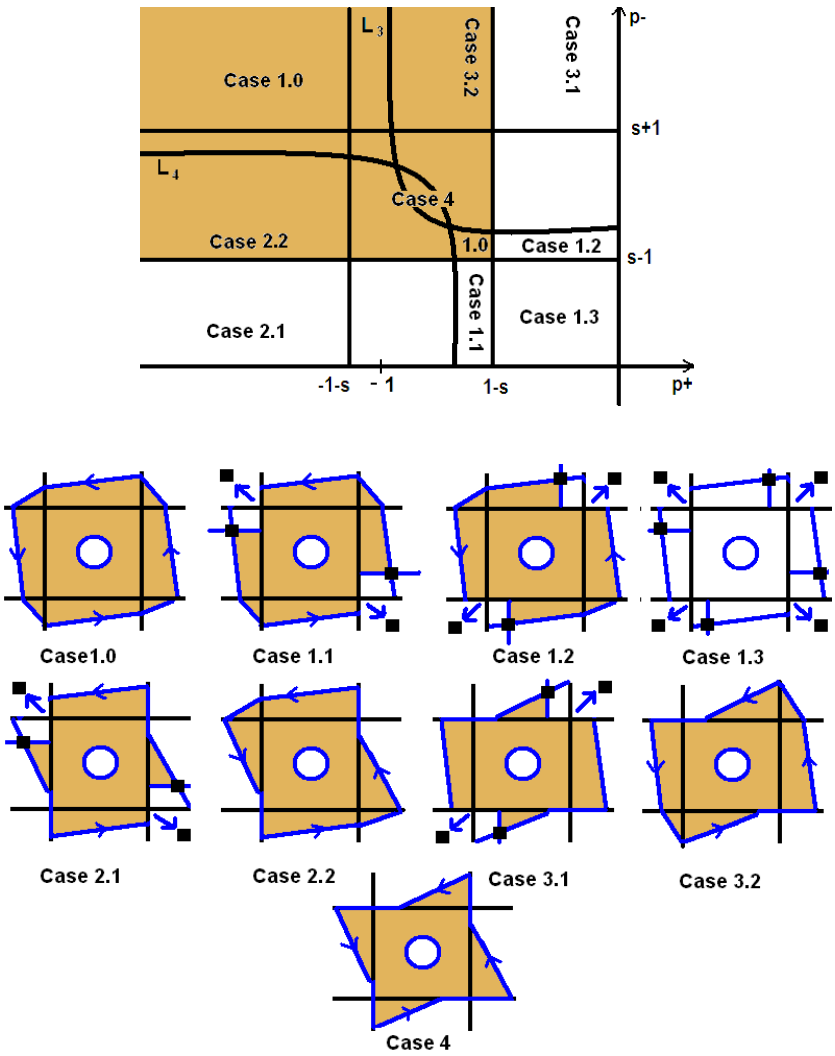


Figure E.36: C_2 construction and regions for $1 < s < \frac{1+\sqrt{2}}{2}$.

Bibliography

[1] Arik, S., Tavsanoglu,V. (1994) A Weaker Condition for the Stability of Nonsymmetric CNNs, IEEE International Workshop on Cellular Neural Networks and their Applications, pp. 15-20.

[2] Arik, S., Tavsanoglu,V., Ozcan,N. (2003) New Criteria for the Existence of Stable Equilibrium Points in Nonsymmetric Cellular Neural Networks, Journal of Electrical and Electronics Engineering, vol. 5, pp. V-753-V-756.

[3] Balsi, M. (1993) Recurrent Back-Propagation for Cellular Neural Networks, Proceedings of ECCTD93- Circuit Theory and Design, Elsevier, pp. 677-682.

[4] Cimalaggi, V., Balsi, M. (1994) Cellular Neural Networks: A Review, E. Caianiello (ed.): "Neural Nets WIRN Vietri-93", World Scientific, Singapore, pp. 55-84.

[5] Chua, L.O., Yang, L. (1988) Cellular Neural Networks: Theory, IEEE Trans. Circ. Syst., vol. 35, pp. 1257-1272.

[6] Chua, L.O., Yang, L. (1988) Cellular Neural Networks: Applications, IEEE Trans. Circuits and Systems, CAS-35, pp. 1273-1290.

[7] Chua, L.O., Roska, T. (1993) The CNN Universal Machine: an Analogic Array Computer, IEEE Trans. Circuits and Systems, vol. 40, pp. 163-173.

[8] Chua, L.O., Roska, T., Venetianer, P. (1993) The CNN is Universal as the Turing Machine, IEEE Trans. Circuits Syst. I, vol. 40, no. 4, pp. 289-291.

[9] Chua, L.O., Thiran, P. (1991) An Analytic Method for Designing Simple Cellular Neural Networks, IEEE Trans. on Circuits and Systems, vol. 38, no. 11, pp. 1332-1341.

[10] Chua, L.O., Wu, C.W. (1992) The Universe of Stable CNN Templates, International Journal of Circuit Theory and Applications, vol.20, pp. 497-517.

[11] Chua, L.O., Roska, T., Venetianer, P. (1990) Stability of a Class of Nonreciprocal Cellular Neural Networks, IEEE Trans. Circuits Syst. I, vol. 37, no. 12, pp. 1520-1527.

[12] Chua, L.O. (1998) CNN: a Paradigm for Complexity, World Scientific.

[13] Chua L.O., Takahashi,N. (1998) On the Complete Stability of Nonsymmetric Cellular Neural Networks, IEEE Trans. Circuits Syst. I, vol. 45, no. 7, pp 754-758.

[14] Chua L.O., Takahashi,N. (1997) A new Sufficient Condition for Nonsymmetric CNNs to Have a Stable Equilibrium Point, IEEE Trans. Circuits Syst. I, vol. 44, no. 11, pp. 1092-1094.

[15] Chua, L.O. (1971) Memristor-The Missing Circuit Element, IEEE Trans. on Circuits and Systems II, Vol. 18, no. 5, pp. 507-519.

[16] Cook, S. (1971) The Complexity of Theorem Proving Procedures, Proceedings of the Third Annual ACM Symposium on Theory of Computing. pp. 151–158.

[17] Deshpande,V., Dasgupta, C. (1991) A Neural Network for Storing Individual Patterns in Limit Cycles , J. Phys. A Math. Gen., vol. 24, pp. 5105-5119.

[18] Doan, M. D., Halgamuge, S., Glesner M., Braunsforth (1996) Application of Fuzzy, GA and Hybrid Methods to CNN Template Learning, Proceedings of CNNA-96, pp. 327-332.

[19] Dmitriev, A.S, Panas, A.I, Starkov, S.O (1991) Storing and recognizing information based on stable cycles of one-dimensional maps, Physics Letters A, vol. 155, pp 494-499.

[20] Dogaru, R., Chua,L.O. (1999) Universal CNN Cells, International Journal of Bifurcation and Chaos, vol. 9, no. 1, pp. 1-48.

[21] Dogaru, R. (2008) Systematic Design for Emergence in Cellular Nonlinear Networks, vol. 95, Springer-Verlag.

[22] Gilli, M. (1994) Stability of Cellular Neural Networks and Delayed Cellular Neural Networks with Nonpositive Templates and Nonmonotonic Output Functions, IEEE Trans. Circuits Syst. I, vol. 41, no. 8, pp. 518-528.

[23] Gilli, M. (1999) Design of Stable Cellular Neural Network Templates, IEEE Trans. Circuits Syst. vol. 35, no. 12, pp. 986 -987.

[24] Gilli, M., Corinto, F. (2003) Comparison between the Dynamic Behavior of Chua-Yand and Full-range Cellular Neural Networks, International Journal of Circuit Theory and Applications, vol. 31, pp. 423-441.

[25] Gilli, M., Civalleri, P. (1993) On the Dynamic Behaviour of Two-Cell Cellular Neural Networks, Journal of Circuit Theory and Applications, vol. 21, pp. 451-471.

[26] Gilli,M. , Civalleri,P. (1999) On Stability of Cellular Neural Networks, Journal of VLSI signal Processing, vol. 23, pp. 429-435.

[27] Gilli,M., Corinto,F., Checco, P. (2004) Periodic Oscillations and Bifurcations in Cellular Nonlinear Networks, IEEE Trans. Circuits Syst. I, vol. 51, no. 5, pp 948-962.

[28] Gómez-Ramírez, E., Mazzanti, F. (2002) Cellular Neural Networks Learning using Genetic Algorithms, Reconocimiento de Patrones: avances y perspectivas, Díaz de León J., Yáñez, C. Eds. IPN, México.

[29] Gardner, M. (1970) The Fantastic Combinations of John Conway's New Solitaire Game 'Life', Sc. Am. 222:4, pp. 120-123.

[30] Guyon, I., Stork, D. (2000) Linear Discriminant and Support Vector Classifiers, In Smola et al Eds. Advances in Large Margin Classifiers. MIT Press, pp. 127-169.

[31] Hebb, D.O. (1949) The Organization of Behavior. Wiley, New York.

[32] Hopfield, J. J. (1982) Neural Networks and Physical systems with Emergent Collective Computational Abilities. Proc. Natl. Acad. Sci. USA, 79, Biophysics.

[33] Jankowski, S., Wanczuk, R. (1992) Nonlinear CNN Cloning Templates for Image Thicking, Proceedings of CNNA-92, pp. 197-201.

[34] Jankowski, S., Wanczuk, R. (1994) CNN models of Complex Pattern Formation in Excitable Media, IEEE Int. Workshop on Cellular Neural Networks and their Applications,, Rome, Italy, pp. 333-338.

[35] Jankowski, S., Tworek, J., Lozowski, A. (1996) Chaotic CNN for Image Segmentation, Proceedings of IEEE Int. Workshop on Cellular Neural Networks and Applications, Sevilla, Spain, pp. 219-223.

[36] Joy, M., Tavsanoglu, V. (1993) A New Parameter Range for the Stability of Opposite-Sign Cellular Neural Networks, IEEE Trans. Circuits Syst. I, vol 40, no. 3, pp. 204-207.

[37] Khalil, Hassan K. (1996) Nonlinear Systems, 2nd edition, Prentice Hall.

[38] Li, X., Huang, L. (2004) Exponential Stability and Global Stability of Cellular Neural Networks, Applied Mathematics and Computation, vol. 147, pp. 843–853.

[39] Lippmann, R. P. (1987) An introduction to Computing with Neural Nets. IEEE ASSP Magazine.

[40] Lehtonen, E., Laiho, M. (2010) CNN Using Memristors for Neighborhood Connections, IEEE Cellular Nanoscale Networks and their Applications, pp. 1-4.

[41] Lehtonen, E., Laiho, M. (2010) Cellular Nanoscale Network Cell with Memristors for Local Implication Logic and Synapses, Circuits and Systems (ISCAS), Proceedings of IEEE Inte. Symposium on, pp. 2051-2054.

[42] Laiho, M.m Paasio, A., Halonene, Kari A.I. (2008) Template Design for Cellular Nonlinear Networks with 1-Bit Weights, IEEE Transactions on Circuits and Systems, vol. 55, no. 3, pp. 904-913.

[43] Mimura1,K., Kawamura, M., Okada, M. (2004) The Path-integral Analysis of an Associative Memory Model Storing an Infinite Number of Finite Limit Cycles. Journal of Physics A: Mathematical and General, vol. 37, pp. 6437–6454.

[44] Minsky, M., Papert, S. (1969) Perceptrons: an Introduction to Computational Geometry. MIT Press, Cambridge, MA.

[45] McCulloch,W. S., Pitts,W. (1943) A Logical Calculus of the Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biology, vol. 52, no. 1-2, pp. 99-115.

[46] Nossek, J.A. (1994) Design and Learning with Cellular Neural Networks, Cellular Neural Networks and their Applications, 1994. CNNA-94., pp. 137-146.

[47] Nemes,L., Chua,L.O, Roska,T. (1998) Implementation of Arbitrary Boolean Functions on the CNN Universal Machine, International Journal of Circuit theory and Applications, vol. 26, no. 6, pp. 593-610.

[48] Roska, T., Chua, L.O. (1993) The CNN Universal Machine: An Analogic Array Computer, IEEE Transactions on Circuits and Systems II, vol. 40, no. 3, pp. 163-173.

[49] Roska, T., Kék, L., Nemes, L., Zaràndy, À., Brendel M. (2000) CSL-CNN Software Library. Report of the Analogical and Neural Computing Laboratory, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, Hungary.

[50] Rosenblatt,F. (1958) The Perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65.

[51] Slavova, A. (2003) Cellular Neural Networks: Dynamics and Modelling, Kluwer Academic Publishers.

[52] Savaci, F.A., Vandewalle, J., (1992) On the Stability Analysis of Cellular Neural Networks, Proc. IEEE Second Int. Workshop on Cellular Neural Networks and Their Applications, pp. 240-245.

[53] Setti, G., Thiran, P., Hasler, M. (1998) An Approach to Information Propagation in 1-D Cellular Neural Networks- Part I: Local Diffusion, IEEE Trans. Circuits Systems I, vol. 45, n. 8, pp. 777-789.

[54] Setti, G., Thiran, P., Hasler, M. (1998) An Approach to Information Propagation in 1-D Cellular Neural Networks- Part II: Global Propagation, IEEE Trans. Circuits Systems I, vol. 45, n. 8, pp. 790-811.

[55] Snider, G. (2007) Self-organized Computation with Unreliable Memristive Nanodevices, Nanotechnology, vol. 18, pp. 1-13.

[56] Strukov, D., Snider, G., Stewart, D., Williams, S., Stanley, R. , (2008) The Missing ,Memristor Found, Nature, vol.453, pp. 80-83.

[57] Strogatz, S.H. (1994) Nonlinear Dynamics and Chaos, Perseus Books, Massachusetts.

[58] Taraglio S., Zanela A. (1996) Cellular Neural Networks: a Genetic Algorithm for Parameters Optimization in Artificial Vision Applications., Cellular Neural Networks and their Applications. Proceedings of CNNA-96, pp. 315-320.

[59] Takahashi,N. , Yamakawa,T., Nishi,T. (2005) Realization of Limit Cycles by Neural Networks with Piecewise Linear Activation Function, Proceedings of ECCTD, Cork, Ireland, vol. 3, pp. III/7 - III10. .

[60] Takahashi,N. , Nishi,T. (2001) On the Global Stability of Two-Cell Cellular Neural Networks with Opposite-Sign Connections, Proceedings of the 15th European Conference on Circuit Theory and Design (ECCTD2001), Espoo, Findland, vol.3, pp.93-96.

[61] Takahashi,N. , Nishi,T. (2002) Necessary and Sufficient Condition for Two-Cell CNNs with Space-Invariant Connections to be Globally Stable, International Symposium on Nonlinear Theory and its Applications (NOLTA2002), Xian, pp.611-614.

[62] Takahashi,N. , Nishi,T. (2006) Necessary and Sufficient Condition for a Class of Planar Dynamical Systems Related to CNNs to be Completely Stable, IEEE Trans. Circuits Syst. I, vol. 53, no. 8, pp. 727-733.

[63] Thiran,P., Crounse,K. R., Chua, L.O, Hasler, M. (1995) Pattern formation properties of Autonomous Cellular Neural Networks, IEEE Trans-

actions on Circuits and Systems, Part I, vol. CAS-42 (I), vol. CAS 42, pp. 757-774.

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

[65] Vinyoles, M., Vilasís, X. (2010) Generating Clock signals using Cellular Neural Networks with Memristor Connections, submitted to ISCAS2011, Rio de Janeiro, Brazil.

[66] Vinyoles, M., Vilasís, X. (2010) Can you achieve any function with a 2-neuron CNN?, NOLTA International Symposium on Nonlinear Theory and its Applications, Krakow, Poland.

[67] Vinyoles, M., Vilasís, X. (2010) Classifying with a two neuron CNN, submitted to International Journal of Bifurcation and Chaos.

[68] Vinyoles, M., Vilasís, X. (2010) Two neuron CNNs: Search for limit cycles, International Journal of Bifurcation and Chaos, vol.20, no. 4 pp. 1137-1173.

[69] Vinyoles, M., Vilasís, X. (2009) Comparison between Chua-Yang and hyperbolic CNNs, IEEE Trans. Circuits Syst, Poster session, Antalya, Turkey, pp. 551-554.

[70] Vilasís, X., Vinyoles, M. (2008) Sufficient conditions for limit cycles in autonomous antisymmetric two neuron CNNs, International workshop on Cellular Neural Networks, Santiago de Compostela, Spain, pp. 236-241.

[71] Vilasís, X., Vinyoles, M. (2005) On cellular neural network learning, Proc. of European Conference on Circuit Theory and Design (ECCTD'05), Cork, Ireland, vol. 1, pp. I/153 - I/156.

[72] Wolfram, S.,(1986) Theory and Applications of Cellular Automata, World Scientific.

[73] Wolfram, S., (1984) Universality and Complexity in Cellular Automata, Physica D, 10.

[74] Zaràndy, À. (1999) The Art of CNN Template Design. International Journal. of Circuit Theory and Applications, vol. 27, pp. 5-23.

[75] Zou,F., Nossek, J.A. (1991) Stability of Cellular Neural Networks with Opposite-Sign Templates, IEEE Trans. Circuits Syst. I, vol. 38, no. 6, pp. 675-677.

[76] Zou,F., Nossek, J.A. (1991) A Chaotic Attractor with Cellular Neural Networks, IEEE Trans. Circuits Syst. I. , vol. 38, no. 7, pp. 811-812.

[77] Zou,F., Nossek, J.A, (1993) Bifurcation and Chaos in Cellular Neural Networks, IEEE Trans. Circuits Syst. I. , vol. 40, no. 3, pp. 166-173.

Universitat Ramon Llull

Aquesta Tesi Doctoral ha estat defensada el dia ____ d _____ de ____

al Centre _____

de la Universitat Ramon Llull

davant el Tribunal format pels Doctors sotasignants, havent obtingut la qualificació:

President/a

_____

Vocal

_____

Vocal

_____

Vocal

_____

Secretari/ària

_____

Doctorand/a

C. Claravall, 1-3
08022 Barcelona
Tel. 936 022 200
Fax 936 022 249
E-mail: urlsc@sec.url.es
www.url.es