SELF ORGANIZING MAP WITH DYNAMIC NEIGHBOURHOOD

PARCOURS RECHERCHE HALF YEAR REPORT

Buoy Emeline

Mines Nancy - Loria

Under the supervision of Bernard Girau

Table of contents

Introduction

For the last few years, the number of studies about artificial neural networks is growing. Indeed, these computing systems inspired by biological neural networks, have many applications in a wide range of disciplines such as medical diagnosis, finance, video games and even more.

By now, the interest remains in how design of multicore reconfigurable circuit may take advantage of bio-inspired principles.

Thus, the aim is to design a circuit in which routing configurations are self-organized.

Two artificial neural networks models are interesting to this problem: self-organizing maps and growing neural gas. Both of these algorithms have characteristics that fit to the issue, but can't be used as it is.

This is why, as part of my Parcours Recherche, I will work on the elaboration of an intermediary model between self-organizing maps and growing neural gas in which the neighbourhood relationship between the prototypes will be able to both suit the inputs and respect hardware restrictions. Thus, the problem is how to create a Self-organizing map of which topology depends of hardware restrictions.

1 Contextual issues and description of the subject matter

1.1 The context and description of the project

1.1.1 Current context

The research of best calculation systems has reach the limits of what quantum physics has discovered. This is why, recent researches are based on computing efficiency which have the advantage to be elaborated with non-conventional approaches such as the brain inspired computing.

Indeed the brain has a very complex architecture with a high density of connexions between neurons. Thus, this structure has already inspired neuromorphic circuit board. However, in addition to the complex architecture, brain has the dynamic ability to modify its connexions over time: cerebral plasticity. It can be a modification because of the lack of stimulations, for instance, blind people are not stimulated by light, and there are no signals coming from their eyes. Thus, the brain will modify the area that normally is allocated to eyes signals processing to the process of other senses such as smell, hear, taste or feel. It can also be after a trauma and some parts of the brain are lost. Thus, other parts are affected in order to never be really unable to do the thing that the affected part used to do.

Nevertheless, this capacity has not been harnessed to the hardware architecture yet. This brain feature is interesting to the elaboration of a model that can suit the inputs and fit hardware restrictions.

Two existing models are interesting to the elaboration of the model: self-organizing map (SOM) and growing neural gas. The aim is to elaborate an intermediate model which combines the learning of the SOM and the reconfiguration of the map at all time.

1.1.2 Description

The SATURN project (Self-Adaptive Technologies for Upgraded Reconfigurable Neural computing) works towards the design of an intelligent embedded robot that will be able to self-organize its tasks so that its architecture will fit to the robot behaviour [1]. The Kohonen's map is used in order to do this allocation of information. Each neuron of the Self-organizing map represent a core of the circuit. The inputs are all the tasks that the system has to process. The self-organization algorithm allocates the task between the neurons using the principle of vector quantification. Finally, the map topology represents the local routing between the cores. The Kohonen algorithm trains the neurons and the communication between the latter has the shape of a grid.

1.1.3 SOMA Project

The SOMA project (Self-organized Machine Architecture) [2] has started in order to develop an architecture that could use the principles of the SATURN project towards applying a SOM in hardware configuration. However, in this work, the configuration is not free. This is why, in the SOMA project, the aim is to adapt the allocation of data to a Network on Chip - which is a communication system on an integrated circuit – of which configuration is free. Thus, the principle of Growing Neural Gases to create the topology in function of the inputs, is very interesting to this problem. However, both of Self-organizing maps and Growing Neural Gas approaches have properties interesting for the project, such as self-organization and free topology, but SOM have a fixed topology and the topology of Growing Neural Gas doesn't take in count hardware restrictions. This is why an intermediate model is needed.

1.2 SOM

1.2.1 The principle of a Self-organizing map

Self-organizing maps (SOM) are a type of artificial neural network trained using unsupervised learning. The latter being a learning model where the inputs are processed as random variable and similar data are gathered together. It has been developed by the statistician Kohonen during the 1980s. This map is usually in 2 dimension and is used to study the repartition of data in a multi-dimensional space. A SOM is inspired by biology reproducing the behaviour of brain to stimuli. Neurons are organized in the brain in order to interpret every stimulus possible. Likewise, the SOM will unfold to best represent inputs by every neuron learning how to best represent a group of similar inputs. Actually, the principle of a SOM is to divide the space in areas where each one will be represented by a vector, in the algorithm it is a neuron [3].

1.2.2 Architecture

In the Kohonen map [3], neurons are organized in a grid. Each neuron communicates with the neurons next to it. The inputs are given to all the inputs in order to find the Best Matching Unit (or winner), which is the neuron of which weight is the closest of the prototype (one of the inputs vector). The architecture is described in Figure 1.

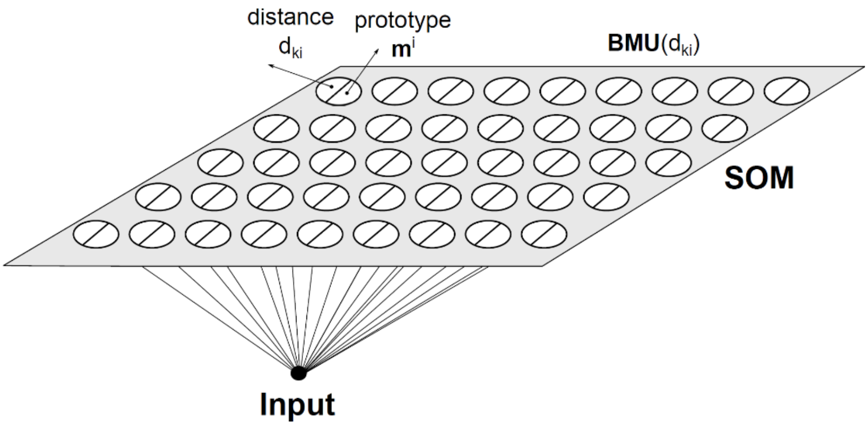

Figure 1 Illustration of the architecture of the Self-Organizing Map

1.2.3 The algorithm itself [4]

First of all, the map is created with each neuron weight randomly set, the algorithm can be optimized by giving to the weights values included in the space of the inputs.

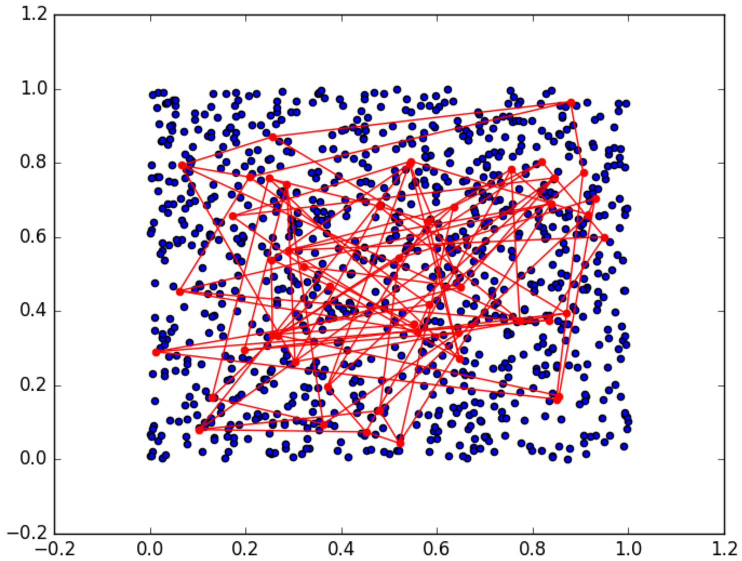

Figure 2 Example of setting weights of a SOM (in red) using random data (in blue) included in [0,1]

Then, the training begins by randomly picking a vector v in the inputs. A winner (BMU) w_g is determined amongst the neurons which is the neuron with the weight nearest to the weight of the vector v. The distance usually used is the L^2 norm.

The training of the winner is done through the following equation:

$$w_g(t+1) = w_g(t) + \varepsilon \cdot \Theta(\sigma, v, w_g(t)) \cdot \left(v - w_g(t)\right)$$

with ε and σ time functions

All the neighbours of the winner vector are trained with an decreasing radius, it means that the farest neurons from the winner are barely modified, through the following equation:

$$w_i(t+1) = w_i(t) + \varepsilon \cdot \Theta(\sigma, i, g) \cdot \ (v - w_i(t))$$

with the same ε and σ, g being the coordinates of the winner neuron

Furthermore, ε and σ are decreasing time function in order to have a convergence of neurons towards a group of data they represent best and to stabilize its weight.

1.2.4 Dynamic

The training is repeted as many iterations as required so that all the inputs are processed. Eventually, the SOM might represent well the inputs.

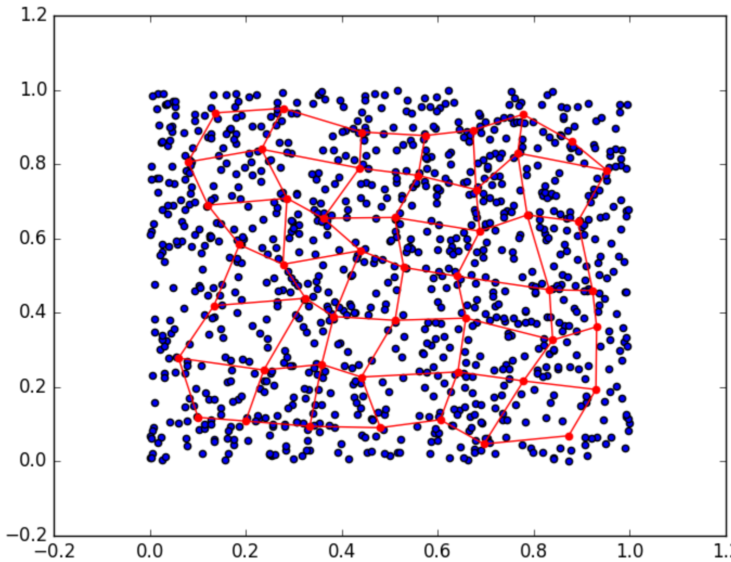

Figure 3 Example of what the SOM might represent after 10000 iterations with the set of the Figure 1

1.3 Growing neural gas (GNG)

1.3.1 The principle of Growing neural gas

Growing neural gas are an algorithm developed after Self-organizing maps where the neurons are trained to represent best data without having a topology defined at first, but from the inputs [5].

1.3.2 The algorithm itself

The growing neural gas is an iterative algorithm. First, such as for the SOM, a number of vector for the GNG is created of which neuron weight is randomly set.

Then, a data vector x is randomly chosen in a data distribution. All of the vectors are adapted according to the following equation:

$$w(t+1) = w(t) + \epsilon \cdot e^{-k/\lambda} \cdot (x - w(t))$$

with ε and λ constant parameters

Then, growing neural gas could fit data and fulfil hardware restrictions at the same time.

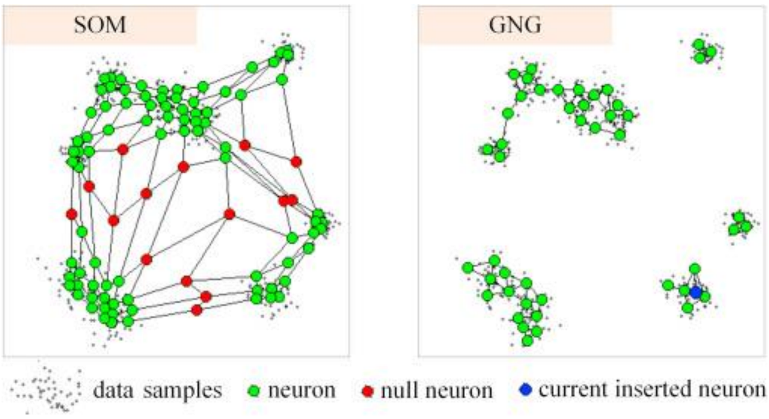

Figure 4 Different results between SOM and Growing Neural Gas. Pattern Recognition Volume 64, April 2017

As it can be seen in the Figure 4, Self-organizing maps are not able to represent data that are made up of many distinct clusters of data. Some neurons are between the clusters because the map topology is static. On the contrary, Growing neural gas with no static topology represent best this kind of data.

2 SOM with dynamic topologies

2.1 Self-organized Machine Architecture

Extract of the proposition of the SOMA project for the ANR [2] describing the 5 layers of this architecture:

"In the SOMA project, we explore hardware self-organization as a change in the processing paradigm itself that would enhance environment awareness in embedded systems and connected objects. The hardware architecture of the SOMA will then imitate this hierarchy structure in an application specific way from raw sensors data to dynamic self-organized

learning structures. The goal of the project is to design this multi-layer architecture. As depicted in figure (6)*, it will be composed of 5 layers:*

- *the first layer is the sensor layer, interfacing the generic architecture to specific types of sensors,*
- *the second layer is the pre-processing layer, extracting higher semantic data (features) from raw data.*
- *the third one is the neural layer. This layer performs the learning of the environment from the pre-processed sensor data. It is the main responsible of the system adaptation.*
- *the fourth layer is the communication layer interconnecting the underlying neurons. The communication topology is self-organized and is mainly studied in the following challenges,*
- *the last layer is the programmable layer, integrating a multi-core pro-cessing map and supporting the application specific software. This layer addresses the problem of the integration of neural and conventional com-puting. It can read back from the neurons the internal representation of the environment learned into the neural layer."*

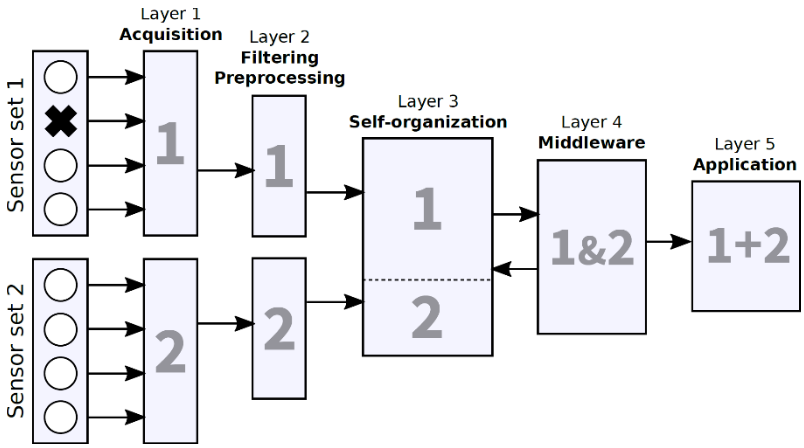

Figure 5 A SOMA architecture with self-organization in layer 3 and between layers 3 and 4

In the SATURN project, the current work is on the third layer of the architecture whereas in a Self-Organized Machine Architecture, the third layer takes in count the fourth layer. The aim is to see if self-organizing properties might have consequences on hardware configuration.

2.2 A possible underlying Network on Chip: FPNA

FPGAs (Field-programmable Gate Array) are integrated circuit designed to be configured by the user after manufacturing. This interesting property of FPGAs has been combined with neural computation called Field Programmable Neural Array [6]. The aim was to easily map neural structures on hardware.

In FPNAs, each neuron is not connected to the others and have a specific location in a kind of grid. Between each neuron there are two operators (represented as arrows going back and forth between each neuron) that can be connected to a neuron or another operator.

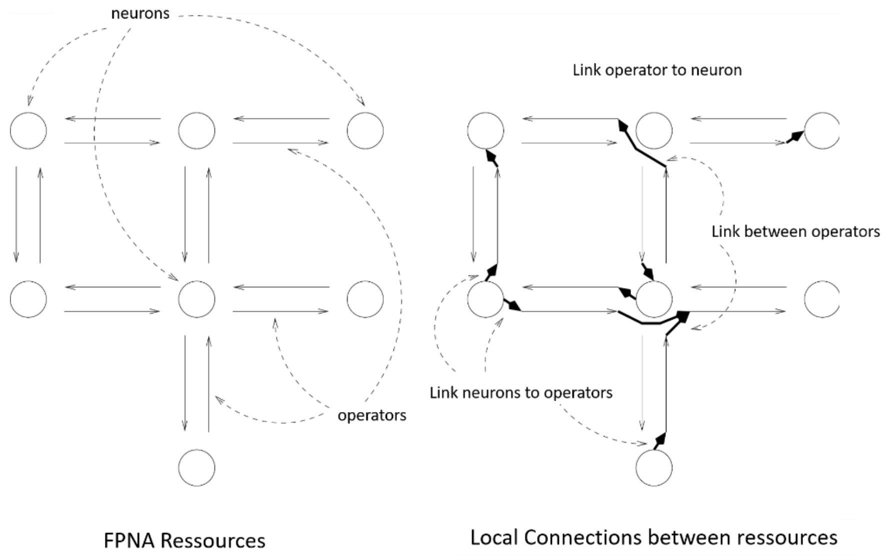

Figure 6 Simplified representation of a FPNA.

2.3 The software development

2.3.1 Self-organizing map algorithm development

The project started with the creation of a Self-organizing map program following the Kohonen algorithm described above. Numbers of SOM classes already exist, but the choice to create a new class has been made in order to modify it when the project will be gone on and a FPNA type routing will be implemented.

Two classes have been created: Neurone and SOM. The Neurone class represents the neurons of a SOM and its object is initialize with a weight, its (i,j) couple of position in the map, its reduce position (i/number_of_row,j/number_of_column) which make easier calculations. The weight is set randomly between the min and the max of the data.

The SOM class creates the map. The latter is a numpy array of which components are neurons. The number of neurons in a row and in a column is given by the user as well as data. This class has many methods in order to train the map as in the Kohonen algorithm.

The values of ε, σ and Θ parameters have been chosen in order to get the best results. Many different functions are used, but only tests can show what sample of parameters fits for the study.

The training equation of the winner is reminded below:

$$w_g(t+1) = w_g(t) + \varepsilon \cdot \Theta(\sigma, v, w_g(t)) \cdot \left(v - w_g(t)\right)$$

The training equation of the winner's neighbours have the same parameters as the winner's.Thus, the equation is :

$$\varepsilon = \varepsilon_0 + (\varepsilon_{max} - \varepsilon_0) \cdot \frac{nbiter - i}{nbiter}$$
$$\sigma = \sigma_0 + (\sigma_{max} - \sigma_0) \cdot \frac{nbiter - i}{nbiter}$$

with $nbiter$ the number of iterations and i the value of the current iteration

Then, Θ is a parameter representing the neighbourhood function decreasing when the distance between the winner and the picked vector or the winner and its neighbours increases.

$$\Theta(\sigma, x, y) = \frac{e^{-\frac{\|x-y\|}{2\sigma^2}}}{\sigma}$$

Thus many tests have been run in order to verify if the chosen parameters were good. The first test was to deploy the map on random data chosen in $[0, 1]^2$. The results were predictable and the map should look like a grid.

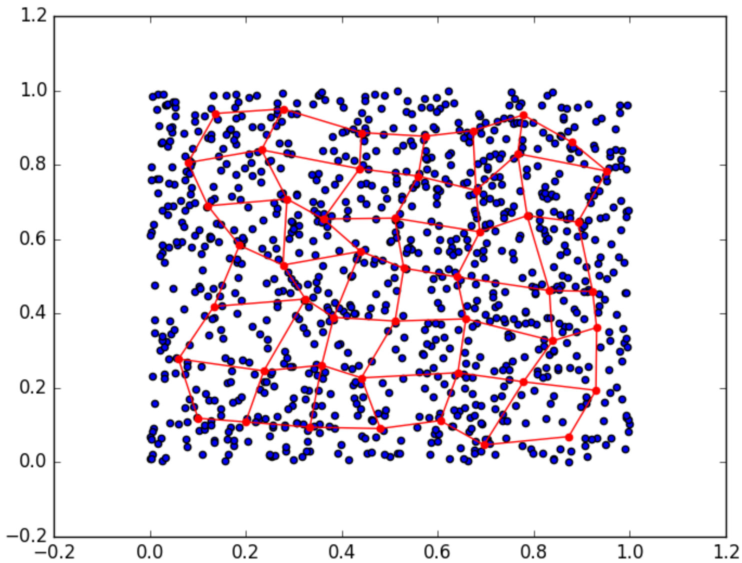

Figure 7 Map deployment on a random dataset after 10000 iterations

Next, a test has been run on data in the shape of weights. The test shows how the map interacts with data that have round shape and are not evenly distributed is the space.

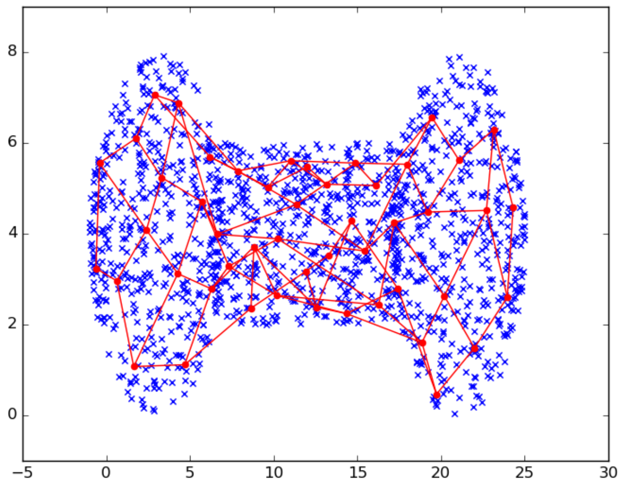

Figure 8 Results of map deployment in data distributed in a weight shape after 30000 iterations

What is significant with these results is that the map does successfully fill in spaces and represent all data. However, the training doesn't successfully unfold the map.

The algorithm itself is not very difficult, but choosing the right parameters after the visualization of all the experience takes much time.

2.3.2 Image compression application

One of the applications of Self-organizing maps is image compression. The principle is to represent the picture with a list of sub-image which colours represent best the picture.

The picture is cut into sub-image of 20x20 pixels and to train the map. Each neuron is a 20x20 pixels sub-image. There are 25 neurons in the SOM. Thus, the compressed image will be represented by a palette of 25 sub-images. Then, the compressed

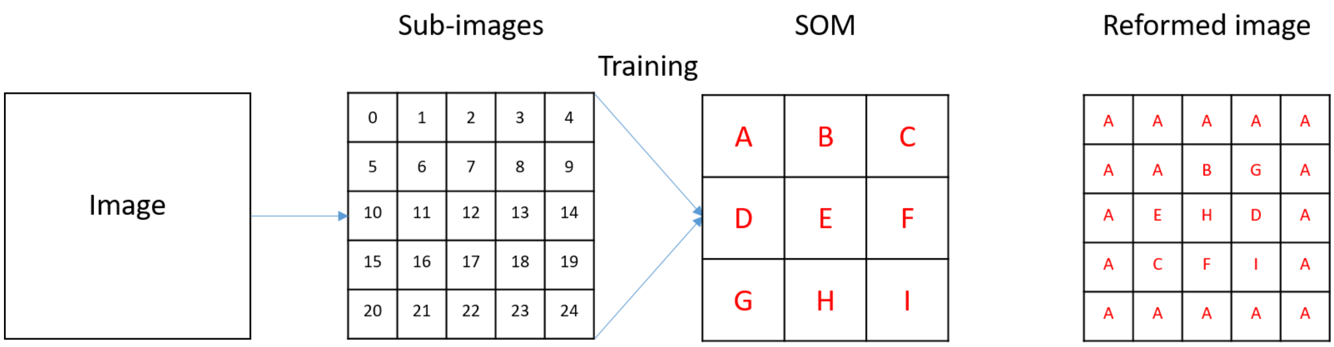

Figure 9 Simplified schema of the image compression process (with only 25 sub-images and 9 neurons)

image only contains the map, a list of the value of the winning neuron for each sub-image, the sizes of the complete image and the sizes of the sub-image.

This application is developed in order to be used in the modified Self-organizing map. Indeed, the way the intermediate model will process is something unknown. This is why, visualizing the results with an image is a good way to have an idea on what is going on.

2.4 Contribution 2: FPNA-SOM definition. A Self-organizing map taking in count hardware restrictions

The SOM and Neurone classes are modified in order to add the operators and links between neurons.

First, each neuron will have a matrix of local connections which will show all the connexions between operators and neurons around of the concerned neuron.

$$\begin{array}{cc} & \text{out} \\ \text{in} & \begin{array}{c|ccccc|} & N & E & W & S & n \\ N & 0 & 1 & 0 & 0 & 1 \\ E & 0 & 0 & 0 & 0 & 0 \\ W & 0 & 0 & 0 & 0 & 1 \\ S & 0 & 0 & 0 & 0 & 1 \\ n & 1 & 0 & 0 & 1 & 0 \end{array} \end{array}$$

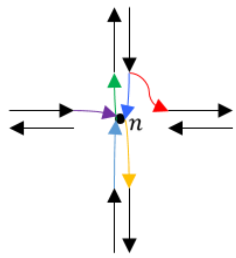

Figure 10 Example a matrix of local connexion

N, E, W, S represents respectively the north, east, west and south operators and the last column and row represent the connections of the neuron. These could be randomly generated or given by the user. The goal is to learn them.

This model raises many questions towards the neighbourhood. Indeed, two neighbour's neurons in the classic SOM might no longer be neighbours. They even can no longer communicate together. Some configuration can also create some cycles. This is why the calculation of distances are reviewed in order to match the model. Many rules are considered in order to define the distance. First, the distance between two connected neurons is considered to be one, even if many operators are between those neurons. Two neurons that are never connected are considered to have an infinite distance. Finally, if there are many path between two connected neurons, it is the shorter distance which is kept.

Conclusion

During those four first month of work, the first goal was to learn about a subject that I know little of and which is not the major of my studies. I could learn through this project that there were many different types of machine learning.

Besides increasing my theoretical knowledge, this project taught me how to synthetize my bibliography reading and to gain much knowledge by myself. Thanks to developing the Kohonen algorithm, I could really understand how it works, and I understood it better than by just reading books and articles.
This project allows me to gain a new skill. Indeed, during all my scholarship, I never ran into an issue that didn't have a solution yet. Thus, I can learn to think differently and to always question what is done and what will be done.

All of this work wouldn't have been done without my tutor Dr. Bernard Girau, whose person I could ask all my questions and who helped me to focus when I didn't know what the main tasks were.

Eventually, this experience is so far very instructive and I am looking forward to do the rest of this project.

Bibliography

[1] A. SATURN, "The SATURN project - self-adaptive technologies for upgraded reconfigurable neural computing, french research agency (anr)," 2011-2014. [Online]. Available: http://projet-saturn.ensea.fr.

[2] B. Miramond, B. Girau, N. Rougier and A. Upegui, *Proposition for ANR PRCI 2017 - SOMA Project*.

[3] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biological Cybernetics,* vol. 43, pp. 59-69, 1982.

[4] N. P. Rougier, "LaBri," [Online]. Available: https://www.labri.fr/perso/nrougier/downloads/Kohonen.pdf.

[5] B. Fritzke, "A growing neural gas network lears topologies," *Advances in Neural Information Processing Systems 7,* p. 625–632, 1995.

[6] J. C. Rajapakse and A. R. Omondi, Eds., FPGA Implementations of Neural Networks, Springer, 2006.