REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique

**Electronics Department**

Laboratoire des Dispositifs de Communication et de Conversion Photovoltaïque

Master thesis on electronics

**Theme**

# Artificial Neural Networks Hardware Implementation

Presented by:

**SAADI Khalid**

Publicly presented in Juin19th 2017

Jury members:

| | | | |
|---|---|---|---|
| SADOUN Rabah | MCA | ENP | President |
| GUELLAL Ammar | PHD | ENP | Mentor |
| LARBES Cherif | Professor | ENP | Mentor |
| HADDADI Mourad | Professor | ENP | Examiner |

ENP 2017

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique

**Electronics Department**

Laboratoire des Dispositifs de Communication et de Conversion Photovoltaïque

Master thesis on electronics

**Theme**

# Artificial Neural Networks Hardware Implementation


Presented by:

**SAADI Khalid**


Publicly presented in Juin19th 2017

Jury members:

| | | | |
|---|---|---|---|
| SADOUN Rabah | MCA | ENP | President |
| GUELLAL Ammar | PHD | ENP | Mentor |
| LARBES Cherif | Professor | ENP | Mentor |
| HADDADI Mourad | Professor | ENP | Examiner |

ENP 2017

# ACKNOWLEDGEMENTS

*I would like to thank My mentor, Amar Guellal for guiding and supporting me over the graduate period. He has set an example of excellence as a researcher, advisor and instructor.*

*I would like also to thank my professor Cherif LARBES for all of his guidance through this process; his discussion, ideas, and feedback have been absolutely invaluable.*

*I would like to thank my fellow graduate students who contributed to this research. I am very grateful to all of you.*

*Finally, we would especially like to thank my family for the love, support, and constant encouragement I got over the years. I undoubtedly could not have done this without them.*

**ملخص:** في السنوات الأخيرة، أظهرت RNA قدرات حسابية عالية وهي تستخدم بشكل متزايد في العديد من المجالات نظرا لمتانة ومرونة هندستها. للاستفادة الكاملة منها عمل الباحثون لإيجاد طريقة أفضل لتنفيذ هذه النظم في مجال البرمجيات أو الأجهزة. أظهر تنفيذها صعوبات كبيرة. وهكذا، تم تقديم دراسة لتحديد أفضل تنفيذ. كل تكنولوجيا الأجهزة المتوفرة لديها مزايا وعيوب. وكانت هناك العديد من الطرق لتصنيف أجهزة التنفيذ في هذه الأطروحة، يتم سرد بعض من أساليب التصنيف وأنواع الأجهزة المستخدمة وفق تصنيف HEEMSKERK

**كلمات مفاتيح :** الشبكات العصبية الاصطناعية, أجهزة التنفيذ, التصنيف.

**Résumé :** Au cours des dernières années, les RNA ont montré des capacités de calcul massives. Ils sont de plus en plus utilisés dans de nombreux domaines en raison de leur robustesse et de leur plasticité dans l'architecture. Pour profiter pleinement des RNA, les chercheurs ont travaillé pour trouver un meilleur moyen de mettre en œuvre ces réseaux en logiciel ou en matériel. L'implémentation des RNA a montré des difficultés. Ainsi, une étude visant à sélectionner la meilleure implémentation a été introduite. Chaque technologie matérielle disponible présente ses propres avantages et inconvénients. Il y a eu de nombreuses approches pour classer le matériel neuronal. Dans cette thèse, il est listé certaines des approches de classification utilisées, puis les types de matériel neuronal utilisés selon l'approche de classification HEEMSKERK. Pour conclure, des exemples sur le matériel neuronal ont été donnés.

**Mot clés : Réseaux de neurones artificiels (RNA), Implémentation hardware , Classification**

**Abstract** In the last decade the ANNs have shown massive computing capabilities. They are being used more and more in many fields because of their robustness and plasticity of architecture. To take full advantage of the ANNs, researchers have been working hard to find a better way to implement these networks in software or hardware. The ANN implementation has shown some difficulties. Thus a study to select the best implementation has been introduced. Each available hardware technology has its own advantages, and drawbacks. There have been many approaches to classify the neural hardware. In this thesis, it is listed some of the classification approaches used, and then the types of neural hardware used according to HEEMSKERK classification approach. To conclude some examples on neural hardware was given.

**Key words: Artificial Neural Network (ANN), hardware implementation, Classification.**

# CONTNENTS

# FIGURE LIST

# INTRODUCTION

It is believed the huge ability of computing of the brain is due the massive parallelism of its processing units called the neurons. The artificial networks were inspired directly from the brain so as to make machines with comparable computing capabilities. But the brain is still a very complex system that scientists have not fully understand yet, even with today's highly developed technologies. Thus modeling it stays a farfetched dream to reach for now. Instead researchers tried to model its most important elementary unit called the neuron. Many works have been done this way, and the results were impressive; the actual models of ANNs made them capable of learning from experience, and have flexibility that allows them to adapt their structure for a particular application. These features are what made them very successful. Now neural networks are being used everywhere, in patter recognition, voice recognition, data mining, machine learning, intelligence control ... etc.

A certain fact is that, in the next decade artificial neural networks (ANNs) based systems, would still have an important role, in many applications. Thus an as based applications designer would still have to choose the best technology to implement them, whether to do that in software or which type of hardware, because this choice still plays a determining factor in the performance of the application.

To choose the most convenient hardware technology for ANNs; a designer must know the available hardware technologies and the performance of each technology with ANNs. There were many approaches to classify the hardware of ANNs.

In this works we have tried to give an overview on the hardware used for ANNs implementations, and the different ANN hardware classifications approaches that were proposed through years

We devoted the first chapter for introducing the neural networks. Then a survey on the hardware for neural networks was presented in the second chapter. Then we concluded by a General conclusion

# CHAPTER 1

# CHAPTER 1. ARTIFICIAL NEURAL NETWORKS (ANNS)

## 1.1 Introduction

Today's conventional digital computers are getting extremely fast; they can perform a lot of instructions and highly complex operations in just few clock cycles, it is way quicker than the human in this. However faster is not enough in solving problems, there are many tasks in which the computer loses against the human brain, the latter is a highly *complex, non linear, and parallel computer* (information processing-system). It works in a totally different way, it has the capability to adapt its structural constituents called neurons, so as to perform certain computations (e.g., patter recognition, perception, and motor control) many times faster than the fastest digital computer in existence today. For instance, given two pictures, a preschool child can easily tell the difference between a cat and a dog. Yet, this same simple task is extremely difficult for today's computers.

The artificial neural network is a machine designed to model the way our brain performs a particular task in solving a given problem, it can be defined as following:

> *A neural network is a massively parallel distributed processor made up of simple processing units whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the inter-unit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns.*[**1**]

This chapter begins with a small historical overview on neural networks, and their development through years, then comes a brief part in which we exposed the biological neuron's anatomy, the origin of the actual model of artificial neurons. After that the mathematical model of a single neuron is presented as well as some topologies of ANNs. To arrive to the most important part, that is the learning characteristic of ANNs, and its different methods. Finally, we presented a non exhaustive list of actual areas of applications for neural networks.

## 1.2 Biological inspiration

### 1. Structure

To create a machine capable of "human-like thought", researchers have used the best available model available "the human brain". However, this one is far too complex to be modeled. Rather, they studied the individual cells that make it up. At the most basic level the brain is composed of neuron cells. They are the basic building blocks of the human brain; there are about 100 milliards of them in it. Artificial neural networks are an attempt to simulate theses cells' behavior.

A stereotypical neuron cell is show in Figure 1-1. It consists of:

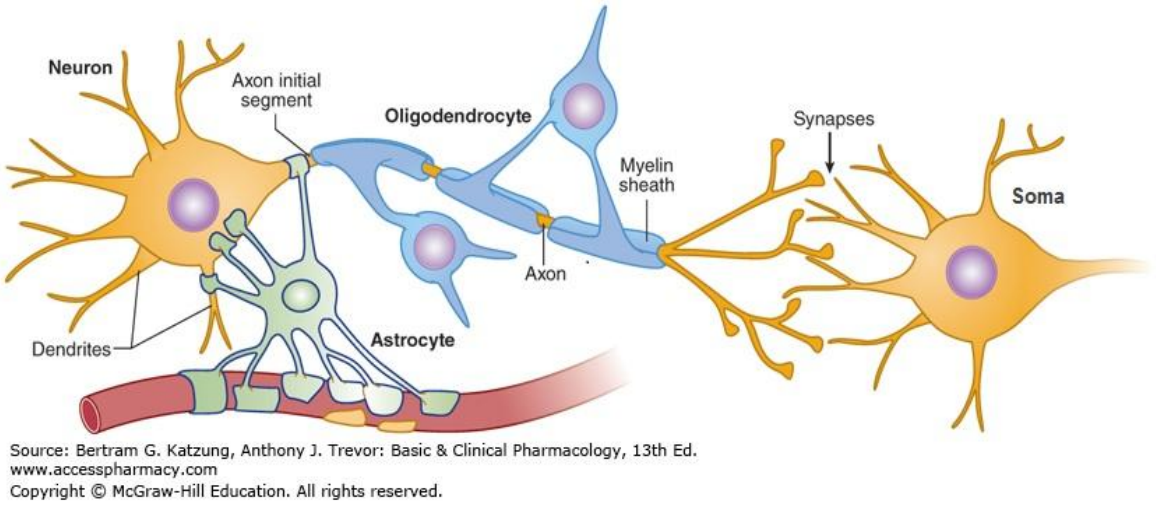



Figure 1-1 A Neuron Cell Anatomy **[**2**]**

**Cell body** or "*soma"* contains the usual sub-cellular components to be found in most cells throughout the body (nucleus, mitochondria, Golgi body, etc.) but these are not shown in the diagram. Instead this diagram was made to focus on what differentiates neurons from other cells allowing the neuron to function as a signal processing device. This ability stems largely from the properties of the neuron's surface covering or membrane, which supports a wide variety of electrochemical processes. Morphologically the main difference lies in the set of fibers that emanate from the cell body. One of these fibers is called the axon.

**The axon** is responsible for transmitting signals to other neurons and may therefore be considered the neuron output. All other fibers are called dendrites.

**The dendrites** carry signals from other neurons to the cell body, thereby acting as neural inputs. Each neuron has only one axon but can have many dendrites. The latter often appear to have a highly branched structure and so we talk of dendritic arbors. The axon may, however, branch into a set of collaterals allowing contact to be made with many other neurons. With respect to a particular neuron, other neurons that supply input are said to be afferent, while the given neuron's axonal output, regarded as a projection to other cells, is referred to as an

efferent. Afferent axons are said to innervate a particular neuron and make contact with dendrites at the junctions called *synapses* see Figure 1-2

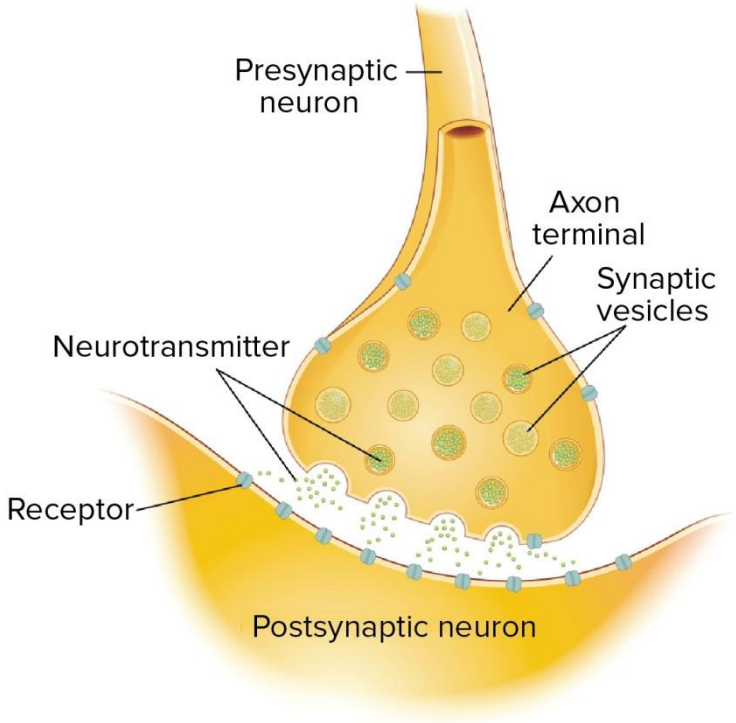



Figure 1-2: The Synapse

Here, the extremity of the axon, or axon terminal, comes into close proximity with a small part of the dendritic surface—the postsynaptic membrane. There is a gap, the synoptic cleft, between the presynaptic axon terminal membrane and its postsynaptic counterpart, which is of the order of 20 nanometers ($2\times10^{-8}$m) wide. Only a few synapses are shown in Figure 1-1, but in reality they are located over all dendrites and also, possibly, the cell body.

Finally the two other cells "Astrocyte" and "Oligodendrocytes" are the Glial cells (Figure 1-1). Their main role is to assure protection for the neuron cells.

## 1.3 Artificial neural networks

An Artificial neural network is an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neuron. The processing ability of the network is stored in the inter unit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns [**3**]. They posses several fundamental characteristics:

- They are composed of two or more layers. Typically, these include an input layer, whose processing units encode the initial representation of the situation, one or more hidden layers, The units combine the information from the input units, and an output layer, Whose units produce the system's response to the situation.

- Simple artificial neurons are connected to other neurons in different layers (and sometimes within the same layer). The weight of connections changes when the system acquires more experience (training), these weights are crucial for determining the treatment performed.
- As in the brain, a given processing unit activates when the stimulus level received from all other units to which it is connected exceeds a certain threshold. The level of stimulus received from each unit is determined, on the one hand, by the degree of activation of that unit and, on the other hand, by the weight of the connection between the sending and the receiving unit.
- The activity of most processing units occurs in parallel (simultaneously).
- Knowledge is represented by the weight of connections within all units of the system.
- Learning occurs when the system that receives inputs, elaborates a response, observes the difference between the response provided and the correct response and adjusts the weight of the connections between the processing units to produce a better response. Adjustments include strengthening some connections and weakening others.
- The generalization of knowledge of the system is based on the similarity of new situations to those already encountered by the system.

## 2. Mathematical model of artificial neuron:

A neuron is an information-processing unit that is fundamental to the operation of a neural network. The Table 1-1 resumes the analogy between a real and an artificial neuron. The diagram of Figure 1-3 shows the model of a neuron, which forms the basis for designing an artificial neural network. Three basic elements of the neuronal modal can be identified:

- ***A set of synapses*** or connecting links, each of which is characterized by weight or strength of its own. Specifically a signal $X_k$ at the input of synapse k connected to neuron $\boldsymbol{j}$ is multiplied by the synaptic weight $W_{jk}$ **.** The first subscript refers to the neuron in question and the second subscript refers to the input end synapse to which the weight refers. Unlike real neurons, the synaptic weights can have negative values.
- ***An adder*** for summing the input signals, weighted by the respective synapses of the neuron; till here the operations described constitute a linear combiner.
- ***An activation function*** for limiting the amplitude of the output of a neuron. The activation function is also referred to as a squashing function in that it squashes (limits) the permissible amplitude range of the output signal to some finite value. Typically, the normalized amplitude range of the output of a neuron is written as the closed interval [0; 1] or alternatively [-1; 1].

The model also in the Figure 1-3 includes an externally applied bias, denoted $\theta_j$ It has the effect of increasing or lowering the net input of the activation function, depending on whether it is positive or negative, respectively.

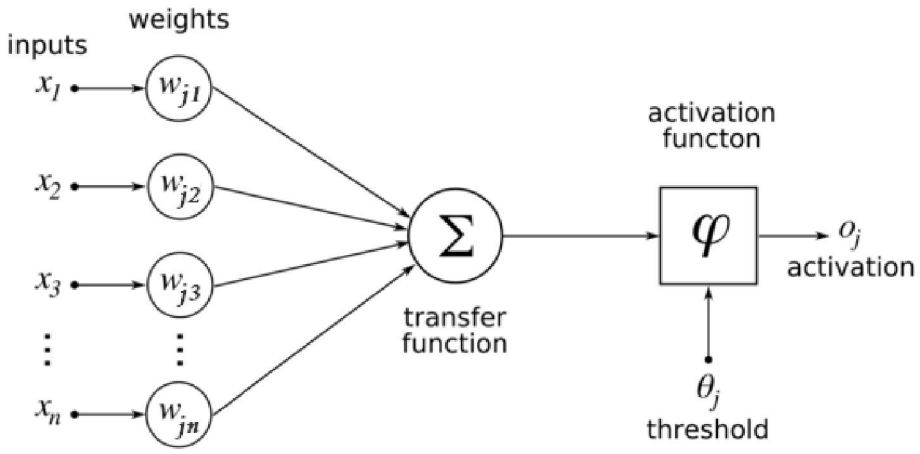



Figure 1-3 : Non linear model of a neuron

In mathematical terms, we may describe a neuron *j* by writing the following pair of equations:

$$\boldsymbol{S_j = \sum_{k=1}^{k=n} W_{jk} \times X_k} \tag{1-1}$$

$$o_j = \varphi(S_j + \theta_j) \tag{1-2}$$

where $X_1$, $X_2$,.... $X_n$ are the inputs, and $\varphi$ is the activation function .

Table 1-1 : Analogy between real and artificial neuron

| **Real neuron** | **Artificial neuron** |
|---|---|
| Cell body (Soma) | Activation Functions |
| Axons | Output signals |
| Synapses | Synaptic weights |
| Dendrites | Input signals |

#### 1.3.1.1 Types of Activation Functions:

Here we identified three basic Activation functions:

- **Threshold Function:** also is referred to as a Heaviside function, it is described the following (Figure 1-4):

$$\varphi(s) = \begin{cases} 1 & if\ s \geq 0 \\ 0 & if\ s < 0 \end{cases} \tag{1-3}$$

This model of neurons based on this Activation Function is referred to in the literature as the McCulloch-Pitts model, in recognition of the pioneering work done by McCulloch and Pitts (1943).

- **Piecewise-Linear Function:** For the one described in Figure 1-4 we have

$$\varphi(s) = \begin{cases} 1, & ,s \geq \frac{1}{2} \\ s + \frac{1}{2}, & \frac{1}{2} > s > -\frac{1}{2} \\ 0, & s \leq -\frac{1}{2} \end{cases} \quad (1\text{-}4)$$

where the amplification factor inside the linear region of operation is assumed to be unity. This form of an activation function may be viewed as an approximation to a non-linear amplifier.

This function can have two special forms:

- A linear combiner, if the linear region of operation is maintained without running into saturation (Figure 1-4).
- A threshold function, if the amplification factor of the linear region is made infinitely large.

• **Sigmoid Function:** It is by far the most common form of activation function used in the construction of artificial neural networks. It is defined as a strictly increasing function that exhibits a graceful balance between linear and nonlinear behavior. An example of the sigmoid function is the logistic function (Figure 1-4), defined by

$$\varphi(s) = \frac{1}{1 + \exp(-as)} \quad 1\text{-}5)$$

where *a* is the slope parameter of the sigmoid function. By varying the parameter *a*, sigmoid functions of different slopes can be obtained. In fact, if the slope parameter approaches infinity, the sigmoid function becomes simply a threshold function. In contrast with the threshold function the sigmoid function assumes a continuous range of values from 0 to 1 and is differentiable (Differentiability is an important feature of neural network theory).

The activation functions defined in equations (1-3**)**, (1-4), and (1-5) range from 0 to 1. Other activation functions are antisymmetric, and range from -1 to 1 (see Figure 1-4)

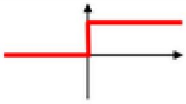
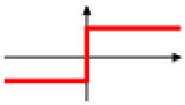
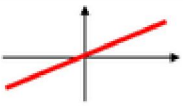
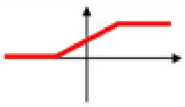
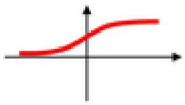
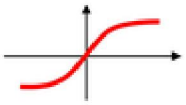
| Activation function | Equation | Example | 1D Graph |
|---|---|---|---|
| Unit step (Heaviside) | $\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$ | Perceptron variant |  |
| Sign (Signum) | $\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$ | Perceptron variant |  |
| Linear | $\phi(z) = z$ | Adaline, linear regression |  |
| Piece-wise linear | $\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$ | Support vector machine |  |
| Logistic (sigmoid) | $\phi(z) = \frac{1}{1 + e^{-z}}$ | Logistic regression, Multi-layer NN |  |
| Hyperbolic tangent | $\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$ | Multi-layer NN |  |

Figure 1-4: Types of Activation Functions

Another important model of artificial neurons is the stochastic neuron, described as follows:

$$\varphi(s) = \begin{cases} 1 & with\ probabilty\ P(s) \\ 0 & with\ probabilty\ 1 - P(s) \end{cases}$$

Where the probability is chosen to be:

$$P(s) = \frac{1}{1 + \exp\left(-\frac{s}{T}\right)} \qquad 1\text{-}6)$$

This model has the same activation function of the McCulloch-Pitts model with a probabilistic interpretation. That is to say that the neuron is permitted to stay in only one of two states 0 or 1. The decision for the neuron to fire (i.e. to change the state from 0 to 1) is probabilistic.

### 3. Architectures of neural networks

From an architectural view, neural networks can sorted into two big categories:

- **Feed-forward** networks, where the data flow from input to output units is strictly feed-forward. The data processing can extend over multiple layers of units, but no feedback connections are present, that is, connections extending from outputs of units to inputs of units in the same layer or previous layers. In this category, we can distinguish single-layer networks (e.g. perceptron) and multilayer networks with an input layer, an output layer and one or more hidden layers (Figure 1-5).

• **Recurrent networks** that do contain feedback connections. In this category, we can distinguish competitive networks, the Kohonen network, the Hopfield networks (Figure 1-6) and the ART models "Theory of Artificial Resonance".

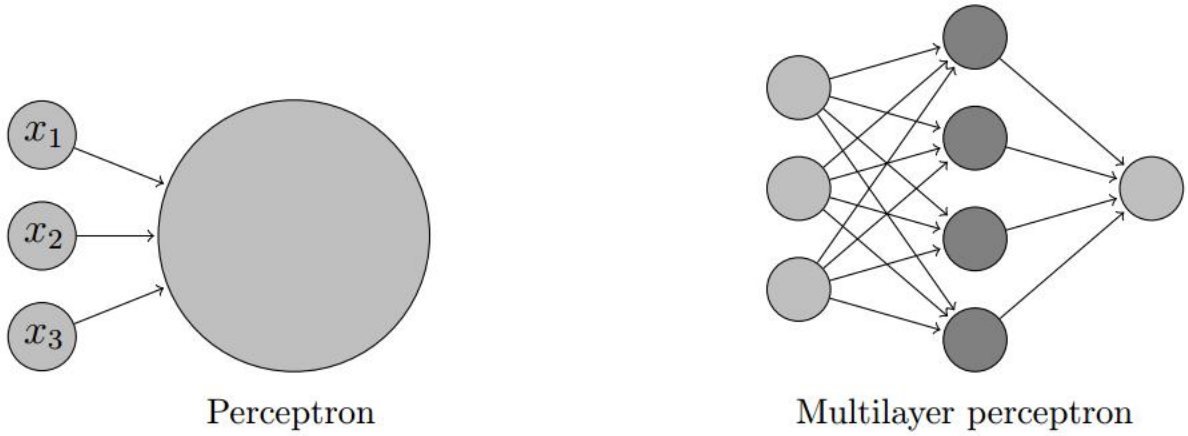



Figure 1-5: Feed-Forward neural networks **[**4**]**

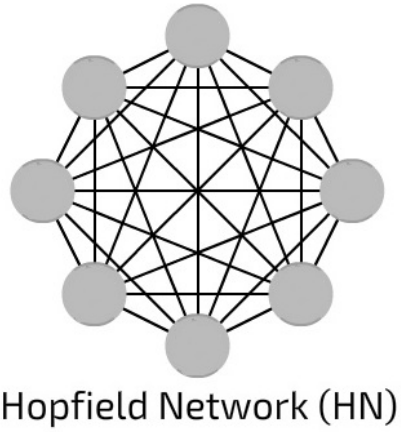



Figure 1-6: Recurrent Network

## 4. Training Neural Networks [5], [6]

In a neural network, individual neurons are interconnected through their synapses. These connections allow the neurons to signal each other as information is processed. Not all connections are equal. Each connection is assigned a connection weight. If a weight is zero then there is not a connection. These weights are what determine the output of the neural network; therefore, it can be said that these weights form the memory of the neural network. Thus training the networks means to configure it (its weights) such that the application of a set of inputs produces the desired set of outputs.

In general training algorithms begin by assigning random values to the weights. Then, the validity of the neural network is examined. Next, the weights are adjusted based on how well the neural network performed and the validity of the results. This process is repeated until the

validation error is within an acceptable limit. There are many ways of training. One way is to set the weights explicitly, using a priori knowledge. Another way is to 'train' the neural network by feeding it teaching patterns and letting it change its weights according to some learning rule.

### 1.3.1.2 Types of trainings:

There are mainly two categories of training (see Figure 1-7):

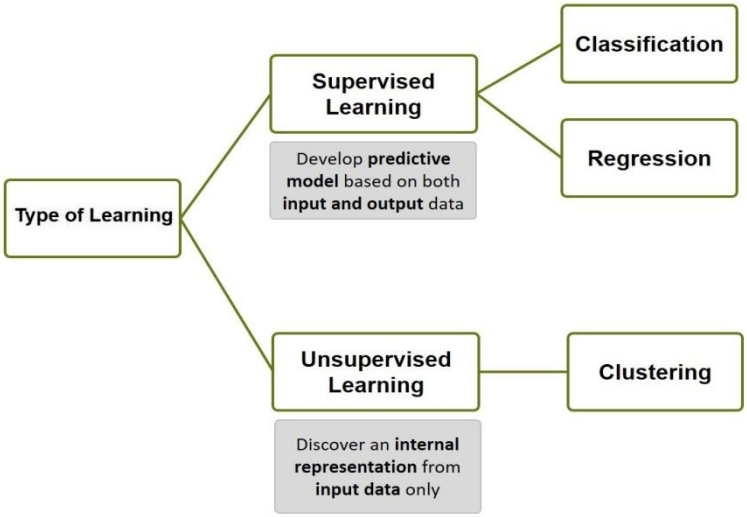



Figure 1-7: Types of training

• **Supervised training** is when the network is trained by providing it a set of inputs along with the anticipated outputs from each of these samples. Supervised training is the most common form of neural network training. As supervised training proceeds, the neural network is taken through a number of iterations, or epochs, until the output of the neural network matches the anticipated output, with a reasonably small rate of error. Each epoch is one pass through the training samples (see Figure 1-8).

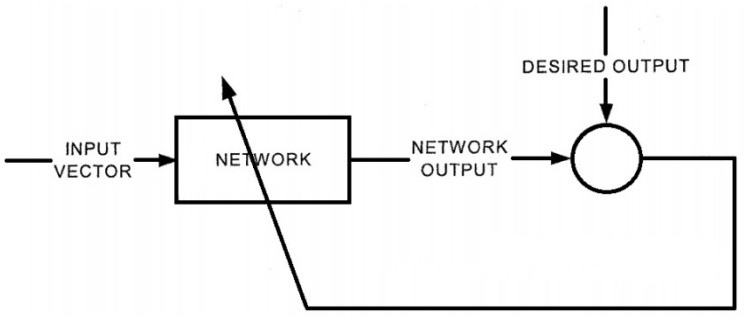



Figure 1-8 : Supervised learning scheme [7]

• **Unsupervised training** is similar to supervised training, except that no anticipated outputs are provided. Unsupervised training usually occurs when the neural network is being used to classify inputs into several groups. The training involves many epochs, just as in supervised

training. As the training progresses, the classification groups are "discovered" by the neural network (see Figure 1-9).

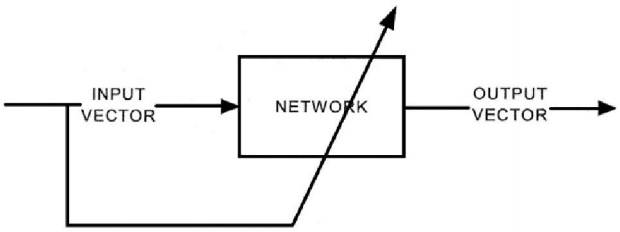



Figure 1-9: Unsupervised learning scheme **[**7**]**

There are several hybrid methods that combine aspects of both supervised and unsupervised training. One such method is called *reinforcement training*. In this method, a neural network is provided with sample data that does not contain anticipated outputs, as is done with unsupervised training. However, for each output, the neural network is told whether the output was right or wrong given the input (see Figure 1-10).

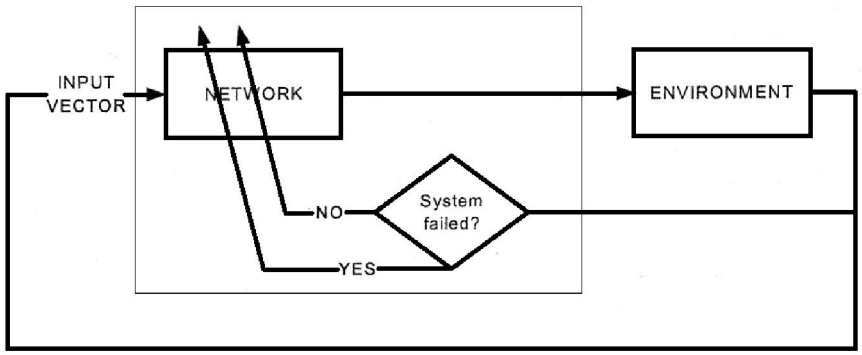



Figure 1-10: Reinforcement learning scheme **[**7**]**

It is very important to understand how to properly train a neural network. Once the neural network is trained, it must be validated to see if it is ready for use.

## 1.4 The properties of neural networks

The main interest in neural networks is justified in the following properties:

- **Learning capacity:**

Learning ability refers to the ability of the neural network to learn to solve problems from examples in a similar way to humans or animals

- **The generalization capacity:**

The ability to generalize translates into the ability of a system to learn and retrieve from a set of examples rules that solve a given problem not learned.

- **The parallelism:**

This notion is at the basis of the architecture of neural networks considered as a set of elementary entities that work simultaneously. Parallelism allows higher computational speed but requires thinking and posing problems differently.

## 1.5 Areas of application of neural networks

Being at the intersection of different domains (computer science, electronics, cognitive science, neurobiology and even philosophy), the study of neural networks is a promising avenue of Artificial Intelligence, which has applications in many areas:

- **Defense:** Weapons management, target tracking, radars: processing, compression, noise suppression, signal / image identification, etc.
- **Industry:** quality control, process control, fault diagnosis, correlations between data provided by different sensors, handwritten signature or writing analysis, speech synthesis, automated vehicle guidance system, etc.
- **Entertainment:** Animation, special effects.
- **Finance:** Forecasting and modeling of the market (currencies ...), forecasting of economic indicators, selection of investments, credit allocation, forecasting of prices, etc.
- Telecommunications and data processing: signal analysis, noise cancellation, recognition of shapes (noises, images and lyrics), data compression, etc.
- **Medical:** analysis of EEG signals, ECG, prostheses, cancer analysis, etc.
- **Environment:** risk assessment, chemical analysis, forecasting and weather modeling, resource management, etc.

## 1.6 Conclusion

To conclude with, the artificial neural networks characteristics inspired from the human brain (parallelism, nonlinearity, and learning) allowed them to perform effectively in many tasks, where a conventional digital computer may have had a hard time. They are being used more and more in many fields because of their robustness and plasticity of architecture. However despite the fact that research in neural networks is an open field, the question is whether it will last long like that, knowing that neural networks have some big challenges like:

1- The actual model is too simple compared to the complexity of the human brain, which means that it stills far from being able to behave like a real brain.
2- There must be a technology that allows the implementation of complex neural networks models.

This leads us to other questions, like: what is the actual technology used in implementing actual neural networks models? And what are the techniques used in these implementations?

# CHAPTER 2

# CHAPTER 2. ARTIFICIAL NEURAL NETWORKS HARDWARE

## 2.1 Introduction

A neural network is a sum of connected elementary processors interconnected in a parallel distribution, forming a given topology. These processing units are implementations of the elementary unit that is the neuron.

In the last two decades there have being a huge diversity in hardware for ANNs. One electrical model that fits for the majority of these types of hardware is the one presented in

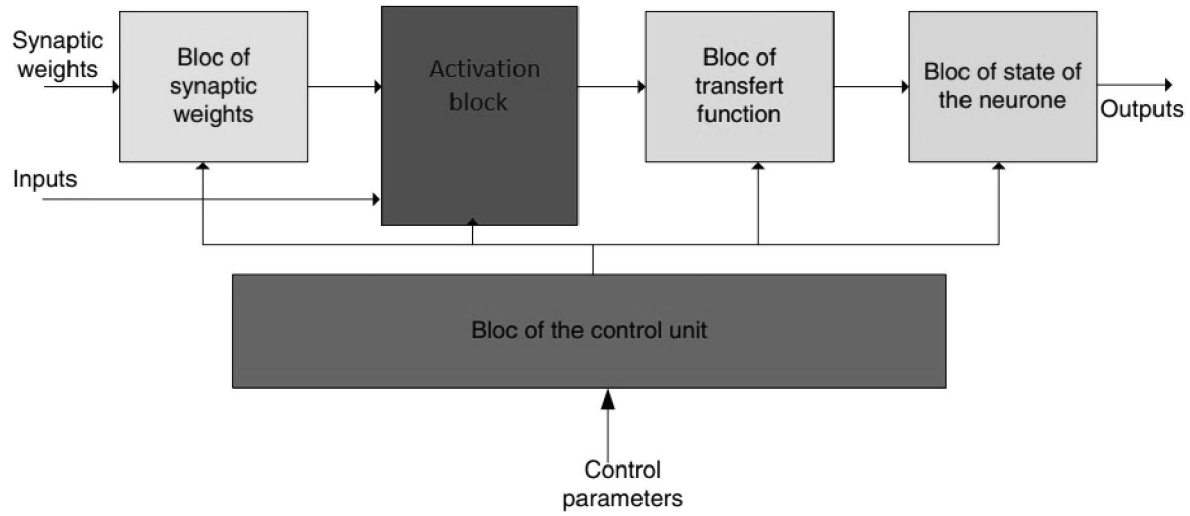



Figure 2-1 : Basic architecture of artificial neuron

The activation block which performs the multiplications and summation of the elementary products is a block that is always in the neuron-chip (in any type of implementation), however the other blocks may be on the chip or off the chip depending on the type of implementation, some of these functions could be performed by a host computer. The data flows by a control block that controls all.

The weights are provided from the synaptic weight block, they get multiplied with the inputs in the activation block, and those products summed together and then fed to the transfer function (activation function) to compute the output. The synaptic weights can be loaded either statically before the activation computation, or they get dynamically updated in the learning phase while activation steps are being performed.

The performance of a neural hardware is quantified by:

- Number of products and accumulation operation in the time unit (MCPS; millions connection updates per second)
- The rate of weights updates (MCUPS: millions connection updates per second)

## 2.2 Approaches of classification of neural hardware:[8] , [9]

The ANNs hardware ranges from a simple neurochips to full-fledged computers called neurocomuters. Many approaches have been taken to classify such a variety of neural hardware, such as system architecture, degree of parallelism, inter-processors communications. In this part we present a non exhaustive list of some famous approaches of classifications in a chronological order. The Figure **2-2** shows order of many neural hardware classifications through the years, from 1972 to 2010:

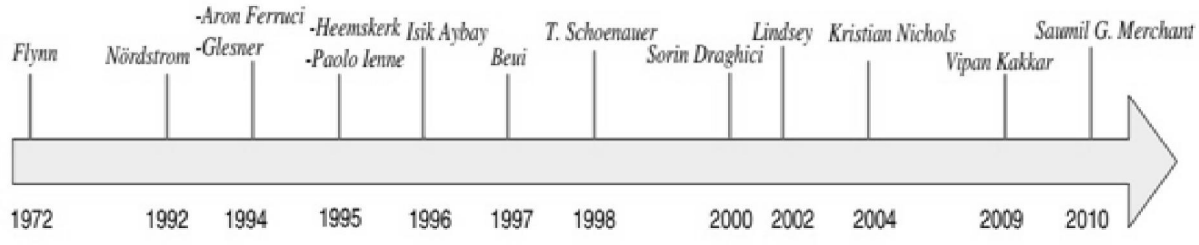



Figure 2-2 : Approaches of Neural hardware classification

### 1. NÖRDSTROM classification approach:

The authors of the classification reference in izeboudgen, have performed a study on the use of parallel machines to implement ANNs .Based on architectural classes defined by reference Flynn (SISD, SIMD, MISD and MIMD), they have shown that the most appropriate architecture for implementing ANNs is the SIMD-type architecture

The criteria used in this classification was the number of processors in parallel) and the complexity of the processors, the defined four degrees of parallelism in relation to the number of parallel processors as it is shown in the Figure **2-3**

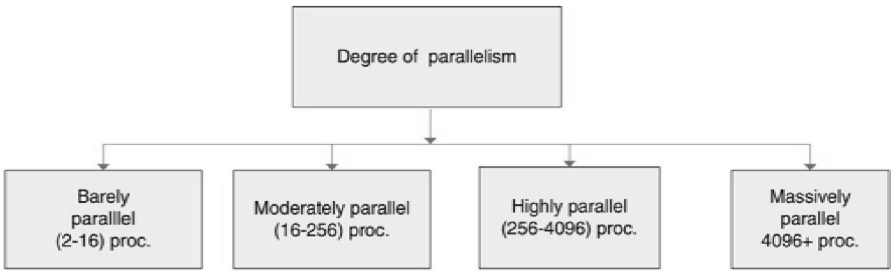



Figure 2-3 : NÖRDSTROM classification

Although this classification quantified the parallelism of an implementation, it didn't mention other important properties like, whether the implementation is in analog or digital.

Another more detailed classification is:

## 2. AARON FERRUCI classification approach

In this approach neurocomuters were classified base on a list of criteria such as:

- The Data representation (Analog values, digital or stochastic bit streams).
- Interconnect strategy (SIMD or MIMD)
- Arithmetic precision
- The technology used (VLSI,DSP,FPGAs,...etc)
- Mapping of algorithm, that means the degree of parallelism, three types were used (neurons parallelism, synapses parallelism and layers parallelism) (Figure 1-4)

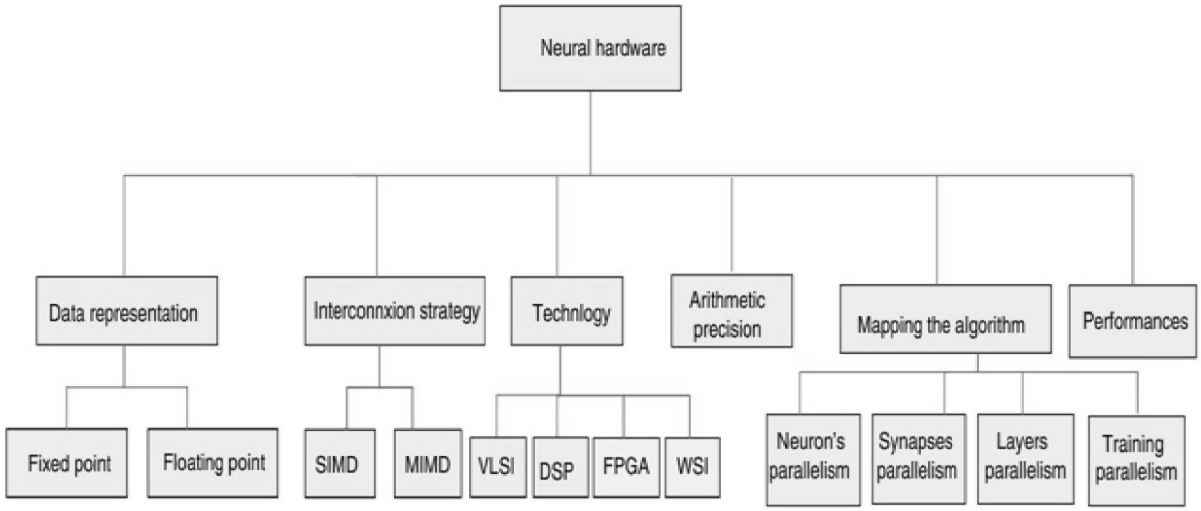



Figure 2-4 : AARON FEUUCI classification

According to this classification is focused on implementations for the back-propagation Algorithm

## 3. PAOLO IENNE classification approach:

It consists of a classification of neural hardware according to two criteria: the flexibility and performances

- Serial systems
- Parallel systems using general purpose processing elements
- Parallel systems using single- model custom processing elements
- Parallel systems using programmable custom processing elements

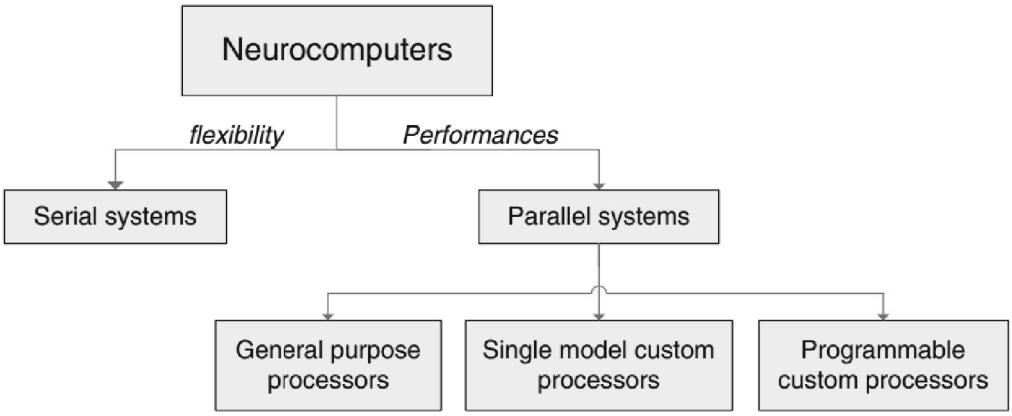



Figure 2-5 : PAOLO IENNE classification approach:

Another important classification approach is

## 4. HEEMSKERK classification approach

Its classification criteria is whether the neurocomputers are built from standards chips or Neurochips

- Standards chips: sequential accelerator boards or multiprocessors.
- Neurochips: where analog, digital and hybrid implementations are classified.

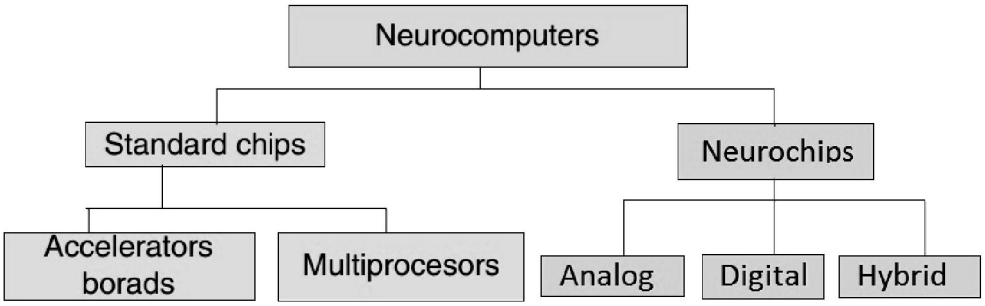



Figure 2-6 : HEEMSKERK classification approach

This classification approach covered a large number of ANNs implementations (Analog, digital and hybrid). However, it was too general because it did not emphasis on the design techniques for each class.

## 5. KRISTIAN NICHOLS classification approach

This classification is based on other criteria:

- Learning/training algorithm that identifies in one hand, if whether the learning type is “on chip” or “off chip” and on the other hand, if the circuit can implement one single or several types of algorithms.

- Signal representation which means implementations using fixed-point representation or those currently using a representation of pulse "Spike Train."
- Multiplier Reduction scheme used for the hardware implementation of neural networks. For this the author reported five types of multipliers; the bit-serial multiplier, the pipeline multiplier, Elimination of the multiplier is obtained by using a typical signal representation that allows the substitution of the multiplier circuit by simple logic functions. The time multiplexed approach.
- The concept of virtual neurons is identical to the concept of virtual memory in the case of computers. For this, an FPGA prototyping board is first chosen as the base platform, then all the parameters of the neural network (synaptic weights, input / output activation function, etc.) are stored in external memory and the rest of the architecture (mainly the multiplier) is implemented into the FPGA. The advantage of this approach is that the maximum number of neurons that can be supported depends only on the size of the external memory available. However the time lost to access the memory is a serious problem which must be taken into consideration.

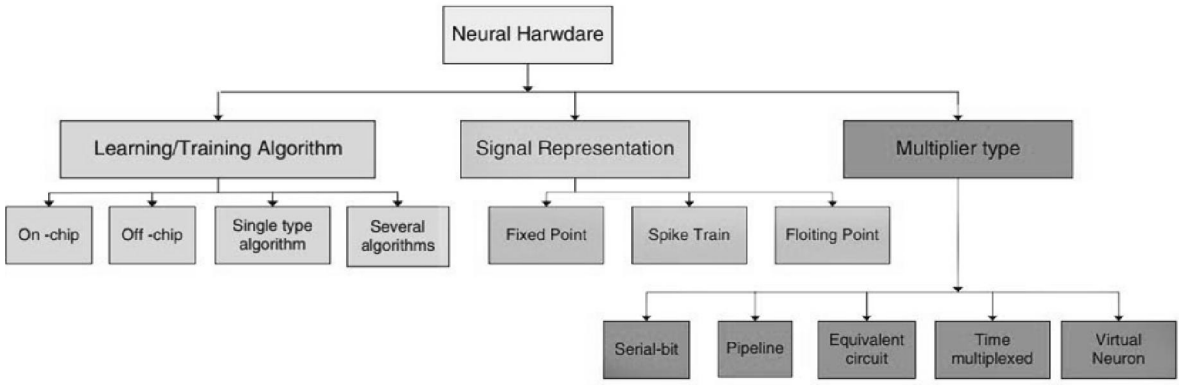



Figure 2-7 : KRISTIAN NICHOLS classification approach

Nichols introduced new criteria related to the learning algorithm as the "on-chip learning" and "off-chip learning". He also used the "the multiplier reduction scheme" as a criterion for classification. Nevertheless, his study focused on FPGA implementations only.

## 6. SAUMIL G. MERCHANT classification approach

This classification is based on the following criteria:

- Digital implementation,
- Analog implementation
- Hybrid implementation

The digital hardware implementations have been grouped into FPGA and ASIC implementations and classified according to design issues such as data representation, design flexibility, on-chip/off-chip learning, and transfer function implementation.

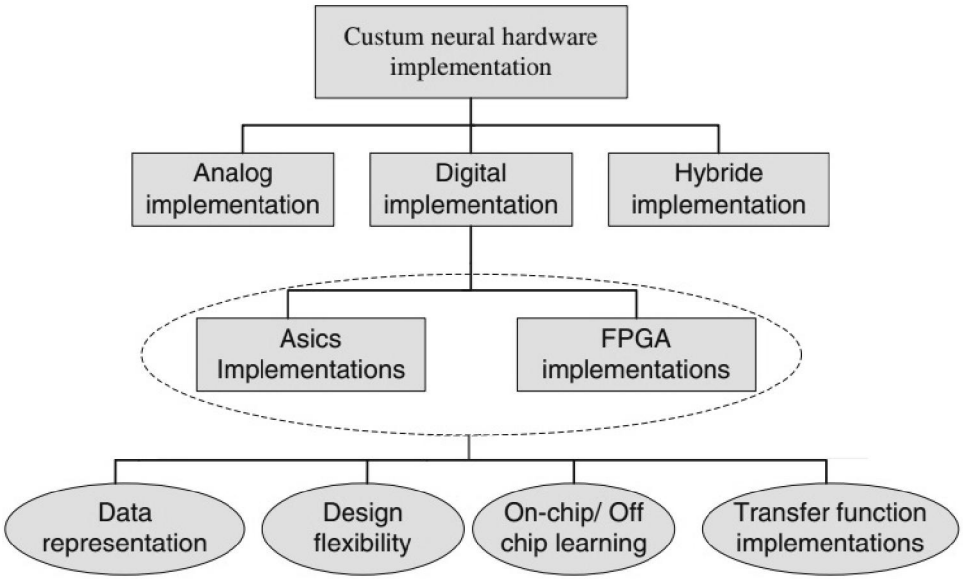



Figure 2-8 : SAUMIL G. MERCHANT classification approach

Both analog, digital and hybrid implementation were considered. Design features related to ASICs and FPGA implementation of ANN such as data representation, flexibility, etc. were considered. However, design issues related to analog and hybrid were not considered. Also, new design techniques and approaches, such as systems on chip, were not considered in this classification.

## 2.3 Types of hardware according to HEEMSKERK classification approach: [10]

In the next part we are going follow the HEEMSKERK classification approach in order to set examples of hardware implementations:

### 1. Accelerator Boards

Used accelerator board are based on neural chips, but many just use fast digital signal processors (DSP), since they have enhanced units for doing fast signal processing like products and some of the products that are needed in ANNs. This type of hardware is the most frequently used for implementing ANNs because of the price, the availability and simplicity of use ( simple to connect to a PC or work station ), they are usually equipped with user-friendly software tool. Despite all these advantages, these accelerators still are specialized for certain tasks, they luck flexibility and do not offer many possibilities for setting new paradigms

## 2. Neurocomputers Built from General Purpose Processors

A general-purpose processor offers enough programmability for the implementation of neural functions. But because of their wide availability and relatively low prices, a number of neurocomputers have been assembled from general-purpose chips. Implementations range from architectures of simple, low-cost elements to architectures with rather sophisticated processors like transputers, or DSPs. Much experience has been gained from these implementations, which can be useful for the design of neurocomputers, i.e., dedicated neurocomputers completely built from special purpose elements like neurochips. For instance, in many cases the sigmoid function forms the most computationally expensive part of the neural calculation. A solution for this can be found in using look-up tables rather than calculating the function .For large numbers of processors the interconnecting strategy has turned out to be another non-trivial problem. Fortunately, much knowledge about the architectures of these massively parallel computers can be directly applied in the design of neural architectures

## 3. Neurochips

Dedicated circuits are devised in special purpose chips for the neural functions. This speeds up the neural iteration time by about 2 orders of magnitude compared to general-purpose processor implementations. Several implementation technologies can be chosen for the design of neurochips. The main distinction lies in choice of a fully digital, fully analog, or hybrid design. Direct implementation in circuits in many cases alters the exact functioning of the original (simulated or analyzed) computational elements. This is mainly due to limited precision. The influence of this limited precision is of great importance to the proper functioning of the original paradigm. In order to build large-scale implementations, many neurochips have to be interconnected. Some chips are therefore supplied with special communication channels. Other neurochips are to be interconnected by specially designed communication elements.

1) **Analog electronics** have some interesting characteristics that can directly be used for neural network implementation. Operational amplifiers (Op amps), for instance, are easily built from single transistors and automatically perform neuron-like functions, such as integration and sigmoid transfer. These otherwise computationally intensive calculations are automatically performed by physical processes such as summing of currents or charges. See Figure **2-9**

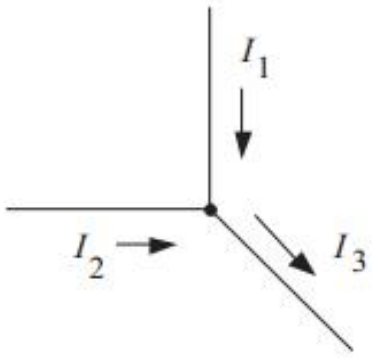



Figure 2-9 : Signals addition in analog

The weighting of the signal can be implemented using variable resistances. Rosenblatt used this approach in his first perceptron designs [185]. If the resistance isRand the currentI, the potential difference Vis given by Ohm's law V =RI. A network of resistances can simulate the necessary network connections and the resistors are the adaptive weights we need for learning see Figure **2-10**

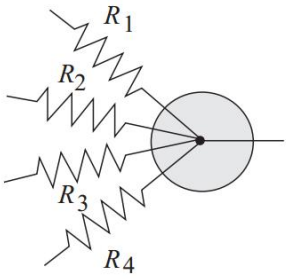



Figure 2-10 : Signals weighting with resistors

Analog electronics are very compact and offer high speed at low energy dissipation. With current state-of-the-art micro electronics, simple neural (non-learning) associative memory chips with more than 1000 neurons and 1000 inputs each can be integrated on a single chip performing about 100 CPS.

Disadvantages of analog technology are the susceptibility to noise and process-parameter variations that limit computational precision and make it harder to understand what exactly is computed. Chips built according to the same design will never function in exactly the same way

Apart from the difficulties involved in designing analog circuits, the problem of representing adaptable weights is limiting the applicability of analog circuits. Weights can for instance be represented by resistors, but these are not adaptable after the production of the chips. Chips with fixed weights can only be used in the recall phase. Implementation techniques that do allow for adaptable weights are: capacitors, floating gate transistors, charge coupled devices (CCDs).The main problems with these techniques arise from process-parameter variations

across the chip, limited storage times (volatility), and lack of compatibility with standard VLSI processing technology. The weight sets for these train-able chips are obtained by training on a remote system (PC or workstation) and are then downloaded onto the chip.

In order to get the benefits of fast analog implementation and the adaptability properties of neural networks, one has to implement learning mechanisms on the chip. Only then can the adaptive real-time aspects of neural networks be fully exploited. However, the implementation of most learning rules into analog VLSI turns out to be very hard. Many research groups are investigating learning methods that better suit implementation in analog circuits. Most proposed methods use the so-called weight perturbation technique that only requires a feed forward phase. These methods have proved to be quite successful .Although analog chips will never reach the flexibility attainable with digital chips, their speed and compactness make them very attractive for neural network research, especially when they adopt the adaptive properties of the original neural network paradigms. A final promising advantage is that they more directly interface with the real, analog world, whereas digital implementations will always require fast analog-to-digital converters to read in world information and digital-to-analog converters to put their data back into the world.

2) **Digital Neural ASICs or FPGAs** are the most powerful and mature neurochips. Digital techniques offer high computational precision, high reliability, and high programmability. Furthermore, powerful design tools are available for digital full- and semi-custom design. Disadvantages are the relatively large circuit size compared to analog implementations. Synaptic weights can be stored on or off chip. This choice is determined by the trade-off between speed and size.

3) **Hybrid Neurochips**

Both digital and analog techniques offer unique advantages, but they also have drawbacks with regard to their suitability for neural network implementations. The main disadvantages of digital techniques are the relative slowness of computation and the large amount of silicon and power that is required for multiplication circuits. Shortcomings of analog techniques are, for instance, the sensitivity to noise and susceptibility to interference and process variations. The right mixture of analog and digital techniques for the implementation of these processes will be very advantageous. In order to gain advantages of both techniques, and avoid the major drawbacks, several research groups have implemented hybrid systems.

## 2.4 Example of Neural hardware

### 1. A Standard chip (parallel machine), The IBM GF11:

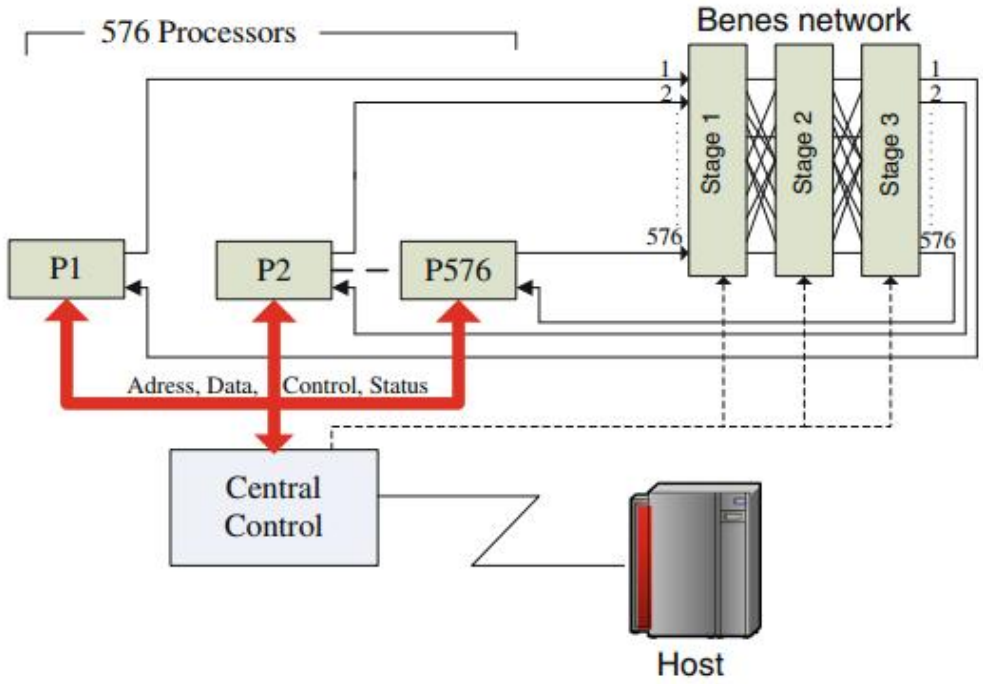



Figure 2-11 : The IBM GF11

IBM GF11 (Beetem et al. 1987) is an experimental SIMD machine with 566 processors interconnected through a BENES network and achieving a peak performance of 11 gigaflops (Figure **2-11**). Each processor is capable of 20 million floating or fixed point operations per second. The processors contain static memories of 16 Kbytes of static memory and 512 Kbytes of dynamic memory. The GF11 has been evaluated on the NETtalk, tool, using a network of 203-60-26 dimension. With 356 operational processors, the GF11 realizes 901MCUPS.The authors estimate that with 566 operational processors, 1231 MCUPS can be reached.

## 2. Digital ASIC circuit, The CNAPS system

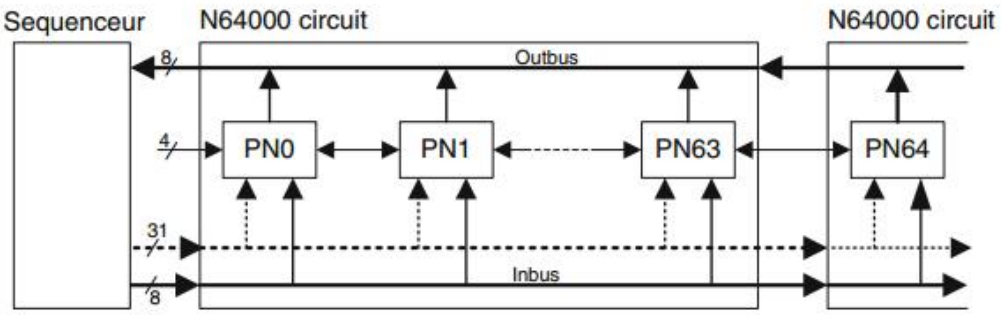



Figure 2-12 : CNAPS multiprocesseur

One of the most well known commercially available neurocomputers is the CNAPS (Connected Network of Adaptive Processors) from Adaptive Solutions. The basic building block of the CNAPS system is the neurochips N6400. As shown in Figure **2-12**, the N6400 itself consists of 64 processing elements (referred to as processing nodes PN) that are connected by a broadcast bus in a SIMD (Single Instruction Multiple Data) mode. Two 8-bit buses allow the broadcasting of input and output data to all PNs.

## 2.5 Conclusion

In this chapter we have seen that have being many hardware implementations for ANNs, this made the choice of the best technology a though task, more over there were no consensus on how to classify hardware for ANNs, we listed a non exhaustive list of classification approaches. After that we used HEEMSKERK classification approach in order to describe types of hardware.

# GENERAL CONCLUSION

To conclude with, the artificial neural networks characteristics inspired from the human brain (parallelism, nonlinearity, and learning) allowed them to perform effectively in many tasks, They are being used more and more in many fields because of their robustness and plasticity of architecture. To take full advantage of ANNs implicit properties, there should be a technology to explicit these properties, A bench of hardware have been used in neural networks. Thus the were nearly no consensus on how to classify them, A small survey was done in this works to list some of the classification approaches used, then the types of neural hardware used according to HEEMSKERK classification approach. Then we concluded with some examples of hardware for different types.

# BIBLIOGRAPHY

[1] Simon Haykin, *Neural Networks - A Comprehensive Fondation, Second Edition*, Prentice Hall ed., 1998.

[2] Bertram G. Katzung, *Basic and Clinical Pharmacology*.: McGraw-Hill Education, 2015.

[3] Kevin Gurney, *An introduction to NUERAL NETWORKS*.: CRC Press, 1997.

[4] Wiley Corning, "Topology of Neural Networks," Thesis 2016.

[5] Jeff Heaton, *Introduction to Neural Networks with C#*, WordsRU.com, Ed.: Heaton Research, Inc, 2008.

[6] Patrick van der Smagt Ben Krose, *An introduction to Neural Networks*.: The University of Amsterdam, 1996.

[7] Paulo E. M. Almeida, Marcelo Godoy Simões Magali R. G. Meireles, "A Comprehensive Review for Industrial Applicability," *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, vol. 50, no. 3, JUNE 2003.

[8] IZEBOUDJEN Nouma, "Plateforme pour l ' Implémentation des Réseaux de Neurones sur FPGA : Application à l ' Algorithme de la Rétro Propagation du Gradient ( RPG )," Ecole nationale Polytechnique, Algiers, Algeria, Thèse de Doctorat 2014.

[9] C. Larbes, A. Farah N. Izeboudjen, "A new classification approach for neural networks hardware: From standards chips to embedded systems on chip," *Artificial Intelligence Review*, vol. 41, no. 4, pp. 491-534, 2014.

[10] Yihua Liao, "Neural networks in hardware: A survey," *Department of Computer Science, University of California*, 2001.