HW/SW CLASSIFICATION OF IMAGES ON FPGA

SAMI EL MOUKHLIS [1], ***ABDESSAMAD ELRHARRAS*** [1] **and** ***ABDELLATIF HAMDOUN*** [1]

[1] **Department of Electronics and treatment of information,**
UNIVERSITE HASSAN II MOHAMMEDIA,
Casablanca, Morroco

Abstract
We present a Hardware/software (HW/SW) implementation of an artificial neural network aims at detecting whether or not that the targeted X ray images of breast contain cancerous cell. This system has been trained in order to approximate functions or to achieve classification from a limited number of data.
We implemented our application on FPGA by using the soft processor NIOS II.

Keywords: *FPGA, NIOS II, Artificial Neural Network.*

1. Introduction

In the last decades, Breast cancer has been considered as the most causes of diseases in Morocco. A study [1] has shown that half of all affected women might die of this.
X ray images are used in order to provide to the doctors information about the internal structure of the breast. It's the most common used diagnosis to increase the chance for early detection.
On the other hand, it has been proved that neural network can provide solutions to many problems in the areas of signal processing, pattern recognition and prediction…
The use of this method allows us to reproduce many characteristics of the human brain not present in Von Neumann model, such as: adaptively, generalization, low energy consumption and learning ability… [2].
In this work, the neral network that predicts whether or not the breast cancer exists, is adapted to run on a NIOS II processor.
This paper is not dealing with the processing of the x-ray image of the breast.
This paper is organized as follows: Section 2 introduces the artificial neuron network theory; section 3 presents the dataset used in the training of the ANN; section 4 shows the HW/SW implementation of the ANN on FPGA

2. Structure of an Artificial Neuron Network

The study of neural network has been inspired by the observation of the human nervous systems. It consists of approximately 10¨11 neurons (Figure 1) each with an average of 10¨3 – 10¨4 connections.
The parallel computing is the secret of how the brain can solve and deal with problems [2].

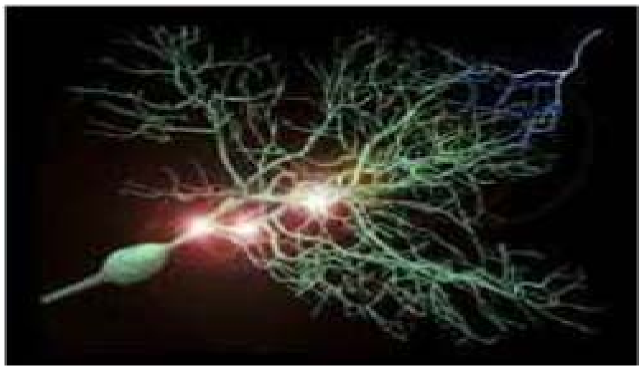


Fig. 1 Human neuron

The artificial neuron model (Figure 2) is proposed by McCulloch and Pitts [3], it attempts to reproduce the characteristic of transmission of signal in biological neurons through synapses.
This model called also perceptron has the input vector x it computes a weighted sum of its components [4].
vk= w1x1+w2x2+……….+wIxI.
The output of the neuron is the result of its activation function: y= f(u).
Where f is: a) a Heaviside function, b) sigmoid function….
The neurons in the MLP (Figure 3) network are organized in the form of layers.
The principal characteristic of the MLP network is that contains a hidden layer. [5]

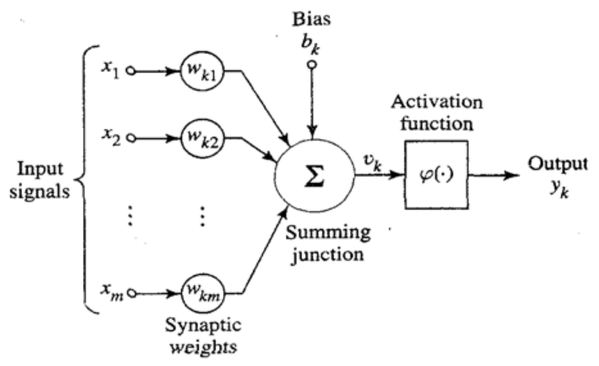


Fig. 1 Nonlinear model of a neuron

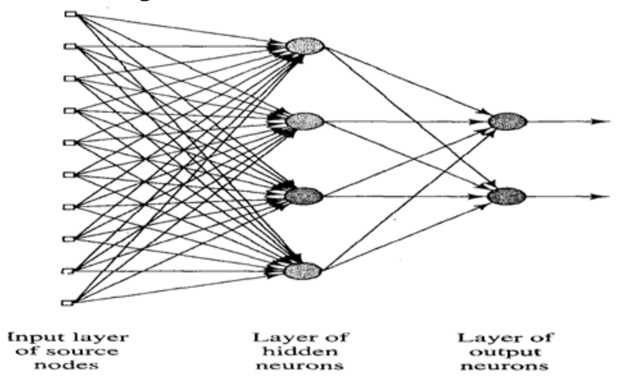


Fig. 1 Feed forward network

The feed forward neural network contains an input layer and an output layer, and the layers that are between the input and the output are called hidden layers.

$$E(t) = \frac{1}{2}\sum_{k=1}^{c}(d_k(t) - y_k(t))^2 \qquad (1)$$

The Back propagation [6] is an iterative algorithm based on minimizing the error given in eq. 1.
The detailed algorithm is shown by the following pseudo-code [7]:

```
Given a labeled data set $X = \{x_1, x_2, x_3, \dots, x_n\} \subset R^n$:
    1. Initialize the synaptic weights to small random
       values [-0:5; +0:5];
{
    2. Randomly arrange the training data;
    3. For i = 1 to n do {
       (a) calculate $y_j(n)$ for j = 1 to c
```

$$y_j(n) = \frac{1}{1 + e^{(-b+\sum_{k=1}^{n} w_k \times x_k)}} \qquad (2)$$

Where m is the total inputs number of neuron j and b its bias (fixed to -1).

(b) By retro-propagating the resulting errors, adjust the weights of each neuron j using the delta rule:

$$w_{ji}(n) = w_{ji}(n-1) + \eta \times \delta_j(n) \times y_i(n) \qquad (3)$$

With

$$\delta_j(n) = y_j(n) \times (1 - y_j(n)) \times e_j(n) \qquad (4)$$

if j $\in$ output layer or

$$\delta_j(n) = y_j(n) \times (1 - y_j(n)) \times \sum_{k \in \ Next\ layer} \delta_k(n) \times w_{kj}(n) \qquad (6)$$

Otherwise. Where $0 < \eta < 1$ is a fixed learning rate and yi(n) the output of neuron i of the precedent layer, if it exists, or the i^{th} component of x otherwise.

```
    }
    1. Repeat steps 2 and 3 until E becomes smaller
       than a specified threshold, or until a maximum
       number of iterations is reached.
```

3. Dataset

3.1 Origin

The dataset used for the training of our ANN was extracted from [8].

3.2 The characteristics of the dataset

3.2.1 General information

The dataset contains 699 instances, each on has 10 attributes.
In this work attributes 2 through 10 have been used to represent instance; after calculation the possible results are malignant or benign.

3.2.2 Attribute information

The attributes extracted from the X ray images describe different proprieties of the breast and its composition; table 1 describes each attributes.

We will not focus on this part in this paper.

Table 1: Attributes

Attribute	***Domain***

Sample code number	id number
Clump Thickness	1 – 10
Uniformity of Cell Size	1 – 10
Uniformity of Cell Shape	1 – 10
Marginal Adhesion	1 – 10
Single Epithelial Cell Size	1 – 10
Bare Nuclei	1 – 10
Bland Chromatin	1 – 10
Normal Nucleoli	1 – 10
Mitoses	1 – 10
Class	2 for benign, 4for malignant

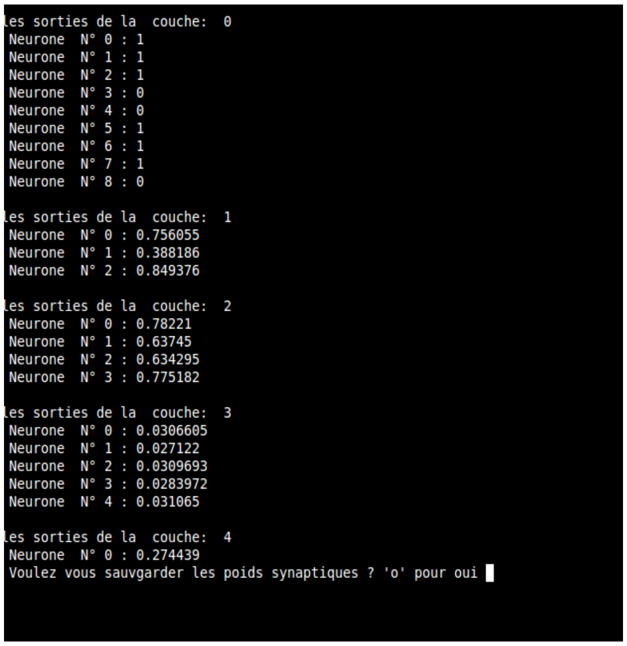
3.2.3 Implementation of dataset

We have divided the dataset in three groups, the first one (400 samples) is used for training process, the second one (100 samples) for validation and 100 samples are reserved for testing our application.
We have used C++ language to program our ANN, we created three classes: a) Neuron, b) Layer, and c) Network.
We have tried multiple architecture in order to create our neural network, but we have fixed the first one (inputs) to 9 neurons, and the last one (outputs) has one neuron. The choice of the 9 neurons in the input layer was governed by the fact that the number of attributes is 9 (we didn't use the first attribute because it has no effect on the result), also the choice of one neuron in the output layer is justified by the reason that we want our system to give us only one result (the cancer is malignant or benign).
We used the classifier error rate τ (%) as a test to choose the best architecture. It is considered as the number of misclassifications in the training (test) phase over the total number of training (test) instances.

Table 1: architecture and corresponding rate of error

ANN	*Arcchitecture*	τ (%) error rate
ANN_	9-1-1	10.5
ANN_	9-25-10-1	4.5
ANN_	9-6-1	4.71
ANN_	9-1	8.1
ANN_	9-10-25-1	4.2

From table 2we can see that best balance between number of neuron and the error rate is the (9-6-1) architecture.



```
les sorties de la  couche:  0
 Neurone  N° 0 : 1
 Neurone  N° 1 : 1
 Neurone  N° 2 : 1
 Neurone  N° 3 : 0
 Neurone  N° 4 : 0
 Neurone  N° 5 : 1
 Neurone  N° 6 : 1
 Neurone  N° 7 : 1
 Neurone  N° 8 : 0

les sorties de la  couche:  1
 Neurone  N° 0 : 0.756055
 Neurone  N° 1 : 0.388186
 Neurone  N° 2 : 0.849376

les sorties de la  couche:  2
 Neurone  N° 0 : 0.78221
 Neurone  N° 1 : 0.63745
 Neurone  N° 2 : 0.634295
 Neurone  N° 3 : 0.775182

les sorties de la  couche:  3
 Neurone  N° 0 : 0.0306605
 Neurone  N° 1 : 0.027122
 Neurone  N° 2 : 0.0309693
 Neurone  N° 3 : 0.0283972
 Neurone  N° 4 : 0.031065

les sorties de la  couche:  4
 Neurone  N° 0 : 0.274439
 Voulez vous sauvgarder les poids synaptiques ? 'o' pour oui
```

Fig. 1 Training process and calculation of weights

We made also other architectures (4 layers, 5 layers (Figure 3) …) but we found that the result is the same so we decided to keep only the architecture of three layers in order to save memory.

4. Implementation

To implement this calculated weights we used a NIOS processor in order to beneficiate of the Hardware/Software design.
We start by creating the Hardware core by using the tool SOPC Builder of Quartus, the system (Figure 4) contains an NIOSII as a processor, an SDRAM to store the attributes of the images, an LCD controller to show the result, keys to command the program and an AVALON BUS to ensure the communication between all these modules…

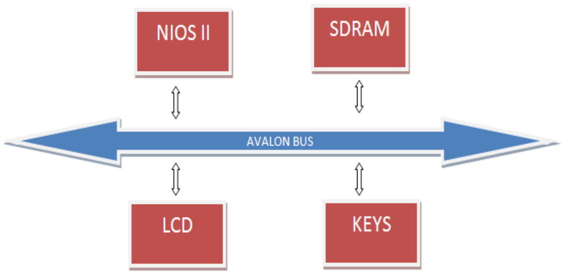
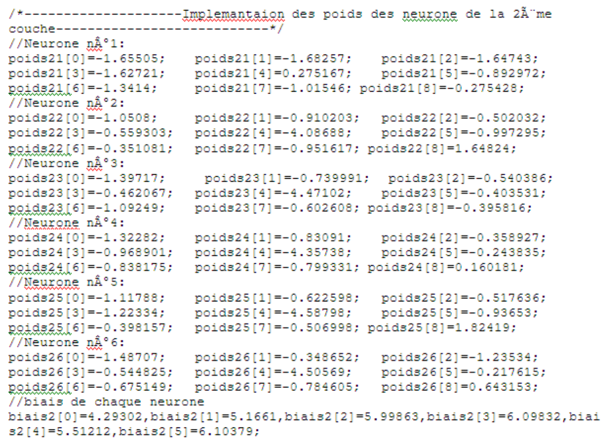


Fig. 1 HW core

The software part has been made by Eclipse, where we create a C program that contains the calculated weights (Figure 5); it starts by charging the attributes into the SDRAM, the attributes of one case are processed and the type of cancer is showed in the LCD.



```
/*--------------------Implemantaion des poids des neurone de la 2Ã¨me
couche----------------------------*/
//Neurone nÂ°1:
poids21[0]=-1.65505;    poids21[1]=-1.68257;    poids21[2]=-1.64743;
poids21[3]=-1.62721;    poids21[4]=0.275167;    poids21[5]=-0.892972;
poids21[6]=-1.3414;     poids21[7]=-1.01546; poids21[8]=-0.275428;
//Neurone nÂ°2:
poids22[0]=-1.0508;     poids22[1]=-0.910203;   poids22[2]=-0.502032;
poids22[3]=-0.559303;   poids22[4]=-4.08688;    poids22[5]=-0.997295;
poids22[6]=-0.351081;   poids22[7]=-0.951617; poids22[8]=1.64824;
//Neurone nÂ°3:
poids23[0]=-1.39717;     poids23[1]=-0.739991;  poids23[2]=-0.540386;
poids23[3]=-0.462067;   poids23[4]=-4.47102;    poids23[5]=-0.403531;
poids23[6]=-1.09249;    poids23[7]=-0.602608; poids23[8]=-0.395816;
//Neurone nÂ°4:
poids24[0]=-1.32282;    poids24[1]=-0.83091;    poids24[2]=-0.358927;
poids24[3]=-0.968901;   poids24[4]=-4.35738;    poids24[5]=-0.243835;
poids24[6]=-0.838175;   poids24[7]=-0.799331; poids24[8]=0.160181;
//Neurone nÂ°5:
poids25[0]=-1.11788;    poids25[1]=-0.622598;   poids25[2]=-0.517636;
poids25[3]=-1.22334;    poids25[4]=-4.58798;    poids25[5]=-0.93653;
poids25[6]=-0.398157;   poids25[7]=-0.506998; poids25[8]=1.82419;
//Neurone nÂ°6:
poids26[0]=-1.48707;    poids26[1]=-0.348652;   poids26[2]=-1.23534;
poids26[3]=-0.544825;   poids26[4]=-4.50569;    poids26[5]=-0.217615;
poids26[6]=-0.675149;   poids26[7]=-0.784605;   poids26[8]=0.643153;
//biais de chaque neurone
biais2[0]=4.29302,biais2[1]=5.1661,biais2[2]=5.99863,biais2[3]=6.09832,biai
s2[4]=5.51212,biais2[5]=6.10379;
```

Fig. 1 SW implementation of the calculated weights

This process is synthesized in the Cyclone II of the Terasic DE2-70 board.
The table Two shows the occupation memory of our system and the execution time.

Table 1: results of implementation

System	Total Logic elements	Execution time (ms)
ANN	7500 (11%)	1250

5. Conclusion

In This work, we have proposed a hardware/software implementation of an Artificial Neural Network. This method is divided into two parts.
The first one: is the training step by using Linux and C++ for coding the Back propagation algorithm and the extraction of the weights of the trained system and the second step: by using VHDL and FPGA for the implementation of the calculated weights.
The results confirm the interest to use FPGA and embedded systems in the implementation of intelligent systems used for decision support.
As perspectives, we have tried to use inputs extracted from X rays images of Moroccan patients

References

[1] Association Lalla Salma de Lutte Contre le Cancer,Guide de protection precoce des cancers du sein, pp 10-13, Edition 2011.

[2]Anil K Jain and J Mao: “Artificial Neural Network: a tutorial”, IEEE, March 1996, pp31-44.

[1] Rumelhart, D. E., McClelland, J. L. and the RDP Research Group, 1986, Parallel Distributed Processing: Exploration in the Microstructure of Cognition, vol. MIT Press, Cambridge? Massachusetts, 1986.

[2] McCulloch, W. S. and Pitts, W., 1943, A Logical Calculus of the Ideas Immanet in Nervous Activity. “Bulletin of Mathematical Biophysics”, vol. 5, 115- 133, 1943.A.B. Smith, C.D. Jones, and E.F. Roberts, “Article Title,” in *Proc. IEEE Int. Symp. Circuits and Systems*, Monterey, CA, pp. 11-14, June 1998.

[3] V. Kecman “Learning andsoft computing: support vectormachines, neural network, and fuzzy logic models”, chap 3, pp 193- 253.

[4] V. H. C. de Albuquerque, A. R. de Alexandria, P. C. Cortez, and J. M. R. S. Tavares, Evaluation of multilayer perceptron and self-organizing map neural network topologies applied on microstructure segmentation from metallographic images,NDT & E International, vol. 42, no. 7, pp. 644-651,Oct. 2009.

[5] Y. Safi and A. Bouroumi, Prediction of Forest Fires Using Arti_cial Neural Networks, Applied Mathematical Sciences, vol. 7, no. 6, pp. 271-286, 2013

[6] K Jain and J Mao: “Artificial Neural Network: a tutorial”, IEEE, March 1996, pp31-44.A.B. Smith, C.D. Jones, and E.F. Roberts, “Article Title,” in *Proc. IEEE Int. Symp. Circuits and Systems*, Monterey, CA, pp. 11-14, June 1998.

[7] http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsi