Modeling of Adaptive Artificial Neural Networks using VHDL is More Appropriate using Bipolar Inputs

Ms. Pooja Verma[1], Ms. Neha Verma[2], Mr. Sanjay Bhandari[1]
1-Associate Professor, Department of Electronics & Communication Engineering, Jodhpur Institute of Engineering & Technology, Jodhpur, India
2-M.E. student, M.B.M Engineering College , Jodhpur, India

***Abstract:*--The training speed of the artificial neural network is affected by choice of initial value of weights and also on the input data representation. Fast learning helps in increased usage of artificial neural networks. This paper discusses the implementation of two artificial neural networks, i.e. single- and multi-layer perceptron, using VHDL. It contains the results of an initial study of supervised training the network with binary and bipolar input data. The results are discussed by comparing the simulations of different logic functions with both types of inputs.**

Keywords - Perceptron, Multilayer perceptron, VHDL, binary and bipolar input patterns, Back propagation.

I. INTRODUCTION

A few decades ago man couldn't think of designing machines which could recognise or classify patterns as efficiently as done by humans. But the advent of artificial neural networks that emulate the functioning of brain, led to the development of intelligent systems that could implement all the tasks like prediction, pattern matching, pattern classification, pattern recognition and many more which was considered to be possible only by living beings.

The most important characteristic of brain is its ability to learn and adapt itself according to the environment.[1] This is done by adapting the free parameters of the network by presenting the environmental stimuli. In the standard neural network learning algorithms, these free parameters basically correspond to the connection strengths, also known as synaptic weights, of the neurons forming the network. The environmental stimuli correspond to a set of input data (or patterns) that is used to train the network. This data can be represented in the binary form ('0' and '1' values) or bipolar form (-1 and +1 values).[2]

Training a network could take a long processing time. The network learns in a supervised or unsupervised manner through an iterative process of weight adjustment. Learning speed of the artificial neural networks can be improved by using bipolar form of training data instead of using binary form of inputs for training. In order to observe the effect of input data representation on training, we have implemented single- and multi-layer perceptron networks VHDL, a hardware description language used to model digital systems.

II. SINGLE-LAYER PERCEPTRON

Frank Rosenblatt, who was a neuro-biologist of Cornell, gave a neuron named as perceptron in the year 1958. This model consisted of only one neuron with synaptic weights that could be modified on the basis of a particular learning mechanism using threshold (signum) function as the activation function. This development introduced the concept of learning in artificial neural networks. Perceptron was used for classification of patterns that could be linearly separated into two categories.

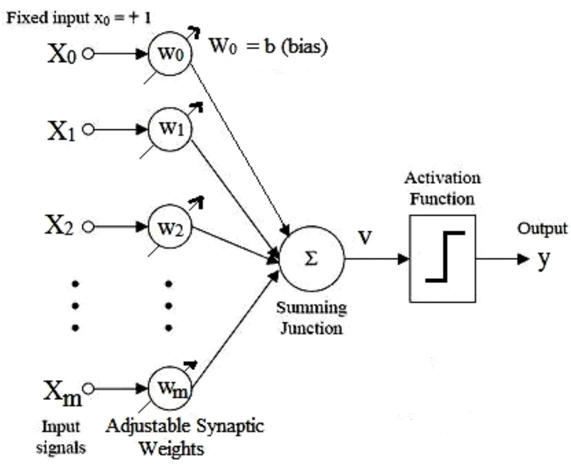

Fig.1. Single-Layer Perceptron

The expression for signum function is given in equation (1).

$$f(v) = \begin{cases} +1, & if \;\; v > 0 \\ -1, & if \;\; v \le 0 \end{cases} \qquad (1)$$

where v is the weighted sum of inputs. The procedure of training of the network, using error correction learning rule [1, 2] is as follows:

Step 1: An input is applied to the network that generates a set of values on the output nodes by flowing through the network.

Step 2: Comparision of the actual and desired output is done such that

- If the two outputs are the same, then no changes are made to the network.
- If the two outputs are different, then synaptic weights are adjusted.

III. MULTI-LAYER PERCEPTRON

The incapability of single-layer perceptron to classify non-linearly separable patterns was dealt with successfully by increasing the number of layers in the network thus giving rise to the multi-layer perceptron model. This network was based on error back-propagation network. [1,3] Werbos was first to develop the back propagation algorithm in 1974. It was then rediscovered by Rumelhart and McClelland in 1986.

This neuron model consists of multiple layers without any limitation on the number of neurons in each layer. There are hidden layers of neurons present in between the input and output layers of the neurons. The input signal traverses through each of the layers of the network.

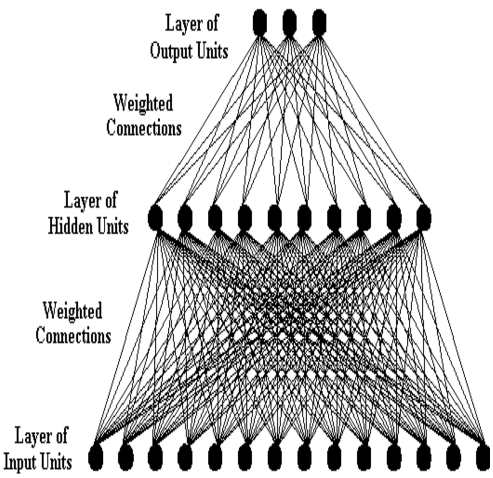

Fig. 2. Multi-layer Perceptron

Each neuron has differentiable nonlinear activation function. A commonly used activation function is sigmoid function given by equation (2).

$$f(v) = \frac{1-\exp(-2\beta v)}{1+\exp(-2\beta v)} \qquad (2)$$

where β is slope parameter and *v* is the weighted sum of inputs. [2]

The back propagation learning algorithm comprises of two steps:

Step1: Computation while traversal of signal from input layer to the output layer

- Training sample is applied to the network.
- This sample traverses through each of the layer of the network until the output pattern is generated at the output layer.
- This output is compared with the desired target and an error is calculated.

Step 2: Computation while traversal of signal from output layer to the input layer:

- The error traverses in the from the output layer towards the input layer, passing each of the hidden layers in the network.
- This error is used to adjust the synaptic weights the actual output approaches the desired value.

IV. VHDL

VHDL is a programming language that is used to describe a digital hardware device model.[4,5] The full form of VHDL is *Very High Speed Integrated Circuit (VHSIC) Hardware Description Language.* One of the advantages of using this language is that the simulation results can be observed in the form of waveforms known as test bench waveforms. [6,7] This allows the designer to analyse several alternatives of the design.

V. SIMULATION RESULTS

Single-layer and multi-layer perceptron networks were designed using Xilinx ISE 12.2 and trained to implement some of the Boolean logic operations. The networks were trained using both binary and bipolar forms of input data with bipolar output. The simulation results were observed with the help of testbench waveforms for each of the logic operation.

A. Training Of Single-Layer Perceptron

The single-layer perceptron was trained to perform logical AND and OR functions. The initial values of synaptic weights of the network were assumed to be 0 with unity learning rate parameter. The results of training are shown in table 1.

TABLE 1. Comparison Results of Training Single-layer Perceptron with Binary/Bipolar Input Patterns

Single-layer Perceptron			
Number of Iterations required for Training the Model			
AND Function		OR Function	
Binary Input Pattern	Bipolar Input Pattern	Binary Input Pattern	Bipolar Input Pattern
6	3	3	2

B. Training Of Multi-Layer Perceptron

The multi-layer perceptron was trained to perform logical XOR and OR functions with binary/bipolar inputs and bipolar output. The fully connected networks, trained using error back propagation learning algorithm, are shown in figure 3. The learning rate parameter was assumed to be 0.1 with momentum constant being 0.6 and unity slope parameter of sigmoid function.

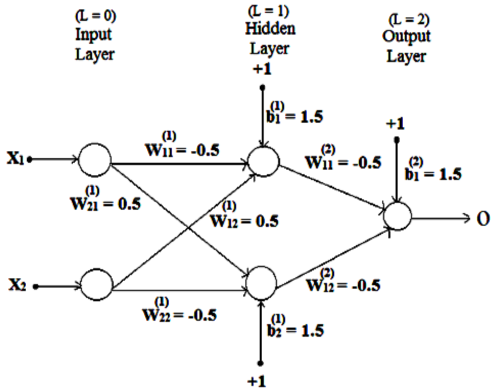

Fig. 3(a) Fully-Connected 2-2-1 Network for Implementing XOR Function

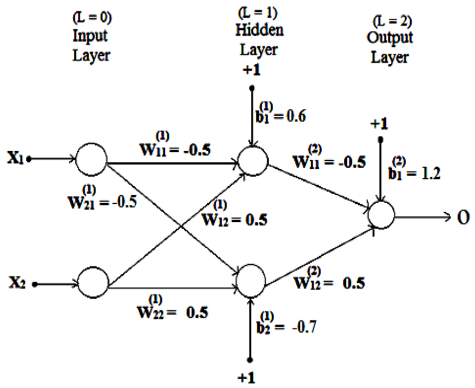

Fig. 3(b) Fully-Connected 2-2-1 Network for Implementing OR Function

The results are summarised in table 2.

Table 2. Comparison Results of Training Multilayer Perceptron with Binary/Bipolar Input Patterns

Multilayer Perceptron				
Logic Function	Binary Input Pattern	Bipolar Input Pattern	Binary Input Pattern	Bipolar Input Pattern
	No. of iterations required for Training	Average Error (*Eavg*)	No. of iterations required for Training	Average Error (*Eavg*)
XOR Function	2	0.003348	2	0.002380
OR Function	2	0.009565	2	0.007290

VI. CONCLUSION

From the tables 1 and 2, it is clear that in case of single-layer perceptron number of iterations required to train the model reduces on changing the data representation from binary to bipolar; while in case of multilayer perceptron though the number of iterations required to train the model remain the same but there is reduction in the value of average error.

Thus it can be concluded that change of training data representation of a neuron model from binary to bipolar form leads to faster training and reduced error.

REFERENCES

[1] Haykin S., "Neural Networks A Comprehensive Foundation", 2nd edition, Pearson Education, 1999.
[2] Fausett L., "Fundamentals of Neural Networks Architectures, Algorithms and Applications", Pearson Education, 1994.
[3] Rumelhart D., Hinton G., Williams R., Learning internal representations by error propagation. In *Parallel Distributed Processing*, *MIT Press*, volume 1, 1986.
[4] Pedroni V., "Digital Electronics and Design with VHDL", Elsevier, 2008.
[5] Perry D., "VHDL Programming by Example", 4th edition, Tata McGraw Hill, 2002.
[6] Ashenden P., "The Designer's Guide to VHDL", 3rd edition, Elsevier, 2009.
[7] Izeboudjen N., Farah A., Titri S. and Boumeridja H., "Digital implementation of artificial neural networks: From VHDL description to FPGA implementation", in proceedings of the Engineering Applications of Bio-Inspired Artificial Neural Networks International Work-Conference on Artificial and Natural Neural Networks, Volume II, IWANN'99 Alicante, Spain, June 2–4, 1999.