ISSN:0975-9646

Neural Network Implementation Using FPGAs

Dhirajkumar S. Jinde[1], Samrat S. Thorat[2]

[1]*E&TC, GCOEA, Amravati,* [2]*E&TC, GCOEA, Amravati*
Maharashtra, India

***Abstract*— In this paper a hardware implementation of an artificial neural network on Field Programmable Gate Arrays (FPGA) is presented. Digital system architecture is designed to realize a feed forward multilayer neural network. The designed architecture is described using Very High Speed Integrated Circuits Hardware Description Language (VHDL). In this we have categorized the any standard data according to their given input specification. Classification is accomplished by using ANN. It is implemented on FPGA..**

Keywords—**Field Programmable Gate Array (FPGA),VHDL, ANN.**

I. INTRODUCTION

Artificial neural networks (ANN) have found widespread deployment in a broad spectrum of classification, perception, association and control applications. Any kind of standard data can be categorized by using the hardware implementation.

Artificial neural networks (ANNs) have been mostly implemented in software. This has benefits, since the designer does not need to know the inner workings of neural network elements, but can concentrate on the application of the neural network. However, a disadvantage in real-time applications of software-based ANNs is slower execution compared with hardware-based ANNs.

Most of the data or applications are based on a Real-time. Hardware is more susceptible than software implementation. In Artificial Neural Network (ANN) MATLAB is used for software implementation and VHDL is used for Hardware implementation mostly. In our paper we used both MATLAB as well as VHDL. Using MATLAB we are find out the weights of the standardized data which is taken from net. By using this calculated weights and inputs from standardized data we can categorize the standardized data.

The simulated waveform are observed in active HDL and implemented in FPGA.

In this project we select the IRIS data sets for classification from net. In this , classification model use multilayer feed forward neural network .The NN has 2 input neuron ,6 hidden neuron ,2 output neuron for iris data sets. The hidden neuron may be change according to obtain the nearest neural network output. Network is designed and trained in software using MATLAB Neural Network processing toolbox. Once network is trained, correct weights are determined, it has to hard coded on FPGA. The VHDL code is compiled, synthesized and implemented in Quartus II.

II. CLASSIFICATION OF IRIS

The iris can be classified into 3 categories i.e. iris setosa, iris versicolour and iris virginica. Depending on the types of attributes i.e. sepal length in cm, sepal width in cm, petal length in cm and petal width in cm.

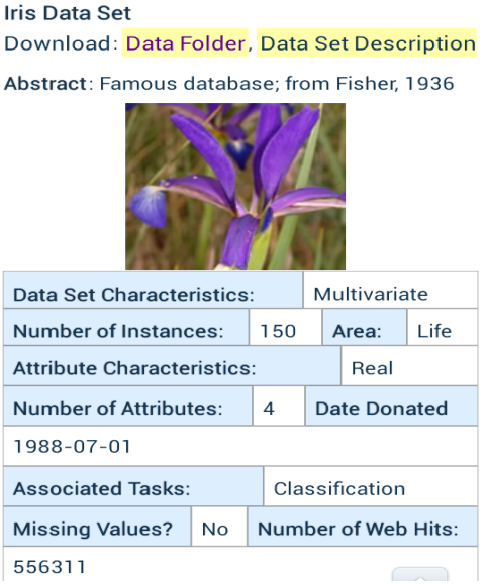

Fig. 1 Iris data set information

III. INPUT AND OUTPUT DATA SETS OF IRIS

The standard iris data 4 inputs are sepal length in cm, sepal width in cm. petal length in cm. and petal width in cm. resp. and in output we get iris setosa, iris versicolour and iris virginica according to the different inputs. (In standard data total 144 inputs is given but it is not possible to written in paper so inputs are used in shorter format).

Sr. no	Sepal length in c.m	Sepal width in c.m	Petal length in c.m	Petal width in c.m	Type of iris
1.	5.1	3.5	1.4	0.2	1
2.	4.9	3	1.4	0.2	1
3.	4.7	3.2	1.3	0.2	1
…					…
…					…
…					…
51.	7	3.2	4.7	1.4	2
52.	6.4	3.2	4.5	1.5	2
53.	6.9	3.1	4.9	1.5	2
…					…
…					…
…					…
97.	6.3	3.3	6	2.5	3
98.	5.8	2.7	5.1	1.9	3
99.	7.1	3	5.9	2.1	3
…					…
…					…
…					…
144.	5.9	3	5.1	1.8	3

Table1. Input and output data for iris

From the iris input and output data it is clear that from 1 to 50 the output is iris setosa (type 1), from 51 to 96 the output is iris versicolour (type 2) and from 97 to 144 the output is iris virginica with different input values.

IV. IMPLEMENTATION IN MATLAB

In Artificial Neural Network weights are really important without finding the weights hardware implementation of Neural network is not possible. So for finding the weights firstly the iris input and output data is implemented in MATLAB and find the weights.

In MATLAB the training of Neural Network is done. In Fig 2.it is shown that the Neural Network in trained for different hidden layers in that input and output layers must be constant. In above GUI shows that 1 to 50 gives the output iris setosa(type 1), 51 to 96 gives the output iris versicolour(type 2) and 97 to 144 gives the output virginica(type 3).

The above GUI also shows that the Neural Network output is attained the same behaviour as the target attained (according to the 3 types of iris). Every time when network is trained the different weights obtained so when the perfect NN output is obtained then save the weights and used for hardware implementation.

From the MATLAB the weights of the trained neural network is calculated and by using these weights the VHDL coding has been written

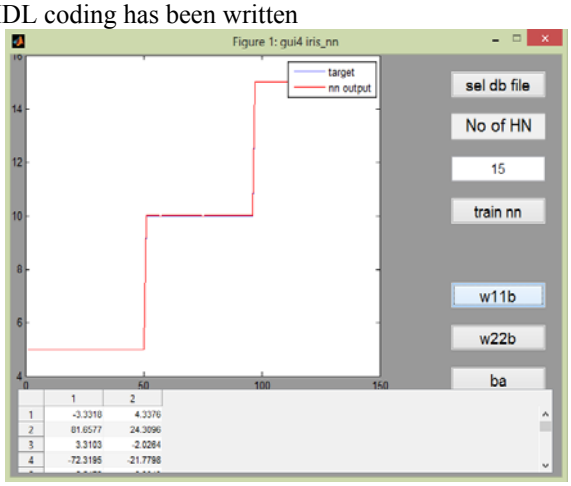

Fig. 2 training of neural network in Graphical Interface Unit (GUI)

V. IMPLEMENTATION IN VHDL

In our design we used the four block given below

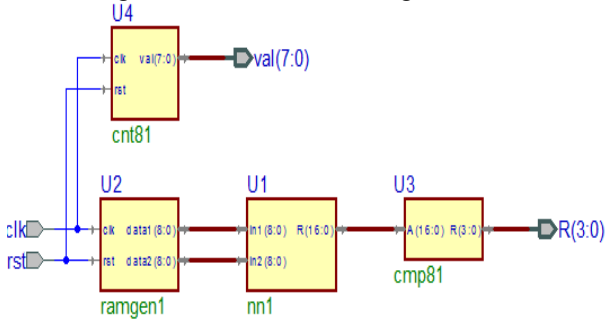

Fig. 3 Final design of hardware implementation of neural network

A. Counter

In the project we prefer the 8 bit counter for giving the inputs because it is not possible to give the total 144 inputs manually. So we are applying the inputs as sepal length, sepal width, petal length and petal width. So we will see the outputs iris setosa when counter counts values from 1 to 50, iris versicolour when counter counts values from 51 to 96 and iris virginica when counter counts value from 97 to 144

B. Ram generator

Ram generator is used as a pattern generator. In the project we categorized the iris data into three types according to given inputs but instead of using the four input we used only two inputs. The Neural Network for only two inputs. So when we see the plots of inputs sepal length and sepal width are the not good classify inputs and petal length, petal width are good classify inputs so we take both. In ram generator the petal length and petal width are separated and created the array of 144 values for both. Ram generator gives 2 inputs data values in each clock. It is for testing Neural Network

C. Neural Network

In the hardware implementation of neural network this block is most important. In this block number of multipliers and number of adder/substractor is used. The 1st block is 8 bit multiplier U1 multiplication of 1st input of petal length and its weight, 2nd input of petal length and its weight similarly multiplication is done . In block U2 similar multiplication happen 1st input of petal width multiplied with its 1st its associated weight, 2nd input of petal width and its weight, similarly with all inputs multiplication is done.

2nd block is 16 bit adder/substractor block. In that block addition of 1st input of U1 and 1st input of U2 is done. Similarly addition done with all remaining inputs of U1 and U2.

3rd block U4 is also 16 bit adder/substractor block. In this block the addition of output weight and output of U3 block is done serially U4.

4th block U5 is also adder block. In this block addition of 1st and 2nd output of U4 block is done. similarly 3rd, 4th and so on. After that output of above both addition again added and similarly 3rd and 4th also and so on. And finally we get only single output.

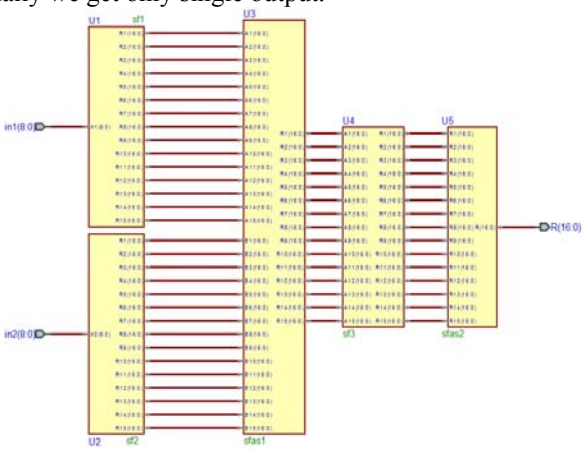

Fig. 4Neural network architecure

D. Comparator

In our design we used the 16 bit comparator. The 16 bit output data comes from U5 is compared with input data i.e. if output data is in between 1 to 50 so we get final output is iris setosa, output data is in between 51 to 96 so final output is iris versicolour similarly for iris virginica also. And finally we get the categorized outputs. Comparator is used as a sigmoid function.

From simulation it is cleared that, FPGA implementation of ANN clearly classifying iris data into 3 categories .1st represent iris setosa, 2nd represent iris versicolour and 3rd represent iris virginica.

VI. SIMULATION RESULTS

From simulation it is cleared that, FPGA implementation of ANN clearly classifying iris data into 3 categories .1st represent iris setosa, 2nd represent iris versicolour and 3rd represent iris virginica.

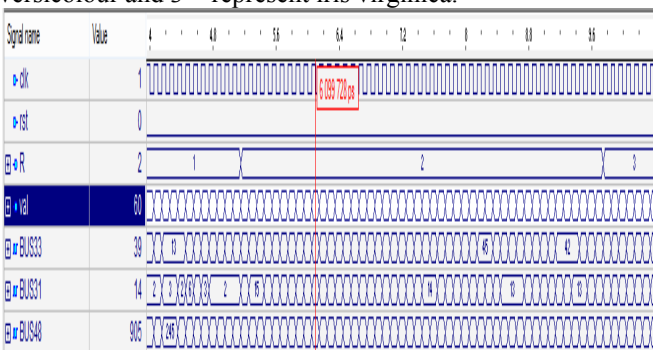

Fig.5 simulation results for hardware implementation of neural network (iris data sets)

VII. CONCLUSION

This paper has presented the implementation of neural networks by FPGAs. The weights and inputs are really important and hence classify. The proposed network architecture is modular and is categorized the iris data (any kind of data NN can categorized) and it is being possible to easily increase or decrease the number of neurons as well as layers because neural networks are inherently parallel structures, parallel architectures always result faster than serial ones.

REFERENCES

[1] Rafid Ahmed Khalil, "Hardware Implementation of Backpropagation Neural Networks on Field programmable Gate Array (FPGA)", Al-Rafidain Engineering, Vol.16, No.3, Aug.2008.

[2] Aydoğan Savran, Serkan Ünsal," Hardware implementation of a feedforward neural network using FPGA".

[3] Esraa Zeki Mohammed and Haitham Kareem Ali "Hardware Implementation of Artificial Neural Network Using Field Programmable Gate Array" International Journal of Computer Theory and Engineering, Vol. 5, No. 5, October 2013

[4] Aniket u trivedi & 2vm umale "Hardware implementation of neural network for vehicle classification using FPGA

[5] A muthuramalingam, S. Himavathi, E. Srinivasan "Neural Network Implementation Using FPGA: Issues and Application" International Journal of Information Technology 4:2 2008

[6] Suhap Sahin, Yasar Becerikli*, and Suleyman Yazici "Neural Network Implementation in Hardware UsingFPGAs