FPGA Implementation of Neuron Block for Artificial Neural Network

ZhaoFang Li and Yu-Jung Huang
Department of Electronic Engineering
I-Shou University
Kaohsiung 84008, Taiwan
e-mail:yjhuang@isu.edu.tw

Wei-Cheng Lin
Medical Engineering Department
E-DA Hospital
Kaohsiung City 824, Taiwan

***Abstract*—This paper presents the FPGA implementation of neuron block units based on a sigmoid activation function for artificial neural networks (ANNs) applications. The Coordinate Rotation Digital Computer (CORDIC) algorithm has been employed for the approximation of sigmoid activation function. The proposed design was simulated using ModelSim XE II and synthesized using Altera's Quartus II with a Cyclone IV EP4CE115 FPGA device. The functionality of neuron block unit was successfully verified using the trained weight on MNIST dataset.**

Keywords—Sigmoid activation function; CORDIC; FPGA

I. INTRODUCTION

Artificial Neural network (ANN) is a field of artificial intelligence (AI) used to models the human brain activities [1]. The concept of deep learning comes from the study of artificial neural network, multilayer perceptron which contains more hidden layers in a deep learning structure. In recent years, deep learning based on the artificial neural network has achieved great success in pattern recognition fields, such as speech recognition, image classification, and face recognition. The software-based ANNs have a disadvantage of slower execution compared with hardware-based ANNs in real-time applications [2].

ANN may be realized by using analog systems or digital systems. In addition, some of the existing platforms available for hardware implementation of ANN are Digital Signal Processing (DSP) chips, Application Specific Integrated Circuits (ASICs), Graphical Processing Unit (GPU) [3] or Field programmable gate array (FPGAs) [4]. As the parallel structure of FPGAs matches the topologies of ANNs, they are quite suitable for the implementation of ANNs [5]. This paper concentrates on FPGA implementation of digital artificial neuron block based on Coordinate Rotation Digital Computer (CORDIC) algorithm for the approximation of activation functions.

II. NEURON MODEL

The multilayer feed-forward ANN structure with the neuron block unit used in the hidden layer is shown in Fig. 1. The training data consists of a set of X_j input patterns where j represents the pattern number and W_{ij} the corresponding trained weight in the ith hidden layer. The more internal neuron a network has, the better that network will be at representing complex solutions. On the other hand, too many internal neurons may cause training to diverge, or lead to overfitting. As shown in Fig. 1, for example, the hidden layer can have 300 units and the output contains 10 units in accordance with 10 different classes.

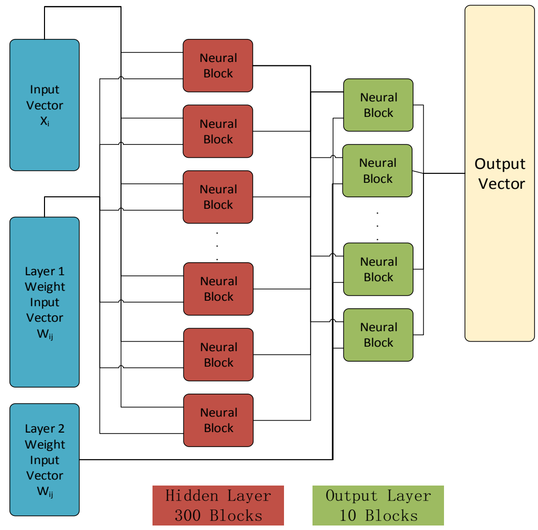

Fig. 1 Neuron block units of hidden layer in ANN

A neuron forms the basis for designing the ANNs. The output activation *f* for the neuron is described by the following equations

$$o_i = f(\sum_{j=1}^{m} w_{ij}x_j + b_i) \tag{1}$$

The w_{ij} denotes the weights connecting the jth input unit to the ith hidden unit. The weighted summation adds up the products of previous neurons multiplied by the corresponding weights, and then, the sigmoid function $f(\vartheta)$ is operated to calculate the output. The bias b_i or threshold can be represented as simply another weight (w_0) with a constant input of 1 (x_0=1). The nonlinear activation is typically chosen to be the sigmoidal function

$$f(\vartheta) = \frac{1}{1+\exp(-\vartheta)} \tag{2}$$

In the presented work, Look-Up-Table (LUT) and Coordinate Rotation Digital Computer (CORDIC) based approximations

[6] have been united for the implementation of the sigmoid transfer function used in the hidden layer of ANN on FPGA platform.

The architecture of a neuron block process unit is shown in Fig. 2. In the proposed work, the FSM control unit is responsible for generating various control signals at the right time. When a set of data x_j and the corresponding trained weight w_{ij} are completely sent, the control line Acc_EN will be set to 0, and the accumulator will send the result to the logic sigmoid function. At the same time, Acc_done will be set to 1, start the operation of the logic sigmoid function unit. The calculation of sigmoid function $f(\vartheta)$ value has been performed using CORDIC-LUT-based approximation algorithm in 32-bit single precision IEEE-754-1985 floating-point number standard.

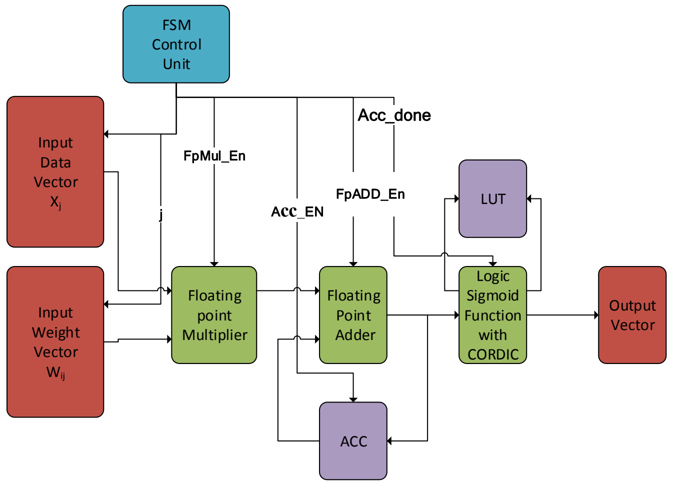

Fig. 2 A neuron block unit

III. FPGA Implementation

FPGA-based ANN has been evaluated using the obtained values of trained weight and bias coded in Verilog. The MNIST dataset provides a training set of 60,000 handwritten digits and a validation set of 10,000 handwritten digits [7]. In the data set, each handwritten digit is represented in a 28 × 28 pixel image. The trained weight matrix obtained from Matlab simulations is used as the input vector of the ANN on the present FPGA implementation. To evaluate the hardware implementation of the neuron block units, the design has been simulated and synthesized using logic synthesis tool Altera Quartus II with an Altera Cyclone IV EP4CE115 FPGA device. Table1 shows the FPGA logic utilization of the proposed design.

Table 1 FPGA logic utilization

Total logic elements	6,618/114,480(6%)
Total Combinational Function	5,906/114,480(5%)
Dedicated Logic Resisters	2,772/114,480(2%)
Total pins	111/529(21%)
Total Memory Pins	0/3,981,312(0%)
Embedded Multiplier 9-bit elements	21/532(4%)

The simulation results of the output of the neuron block on ANN-based architecture with 28x28 inputs and trained weight matrix is shown in Fig. 3. The parameter j indicates the current data x[j] from MNIST dataset and Matlab trained weight Weight_Wij read from the input vectors for ANN simulation. Neural_Output is the output of neurons implemented based on the present proposed architecture. The output results of neuron block units are consistent with the results from Matlab simulation, which confirm our successful implementation.

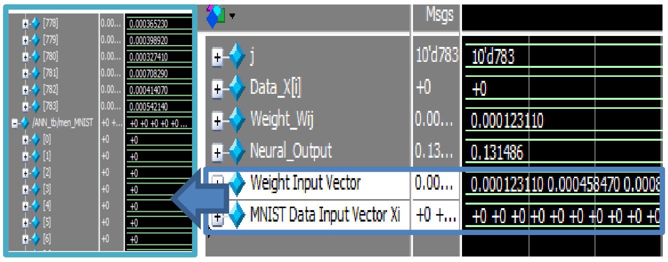

Fig. 3 Simulation result of neuron block unit in ANN

IV. CONCLUSION

The proposed ANN consists of one hidden layer with 300 neurons and one output layer with 10 neurons to recognize 10 different handwritten digits. This work has the conclusion of successful FPGA implementation of neuron block unit using CORDIC algorithm for the approximation of activation functions. The Matlab trained weight vector for the neuron block unit from MNIST is used to verify our output of the present proposed ANN architecture.

Acknowledgment

This work was supported in part by the I-Shou University under Grants ISU-105-ICU-02.

References

[1] Simon Haykin, Neural Networks: A Comprehensive Foundation, 2ed. Addison Wesley Longman (Singapore) Private Limited, Delhi, 2001

[2] D.S. Jinde, S.S. Thorat, Neural network implementation using FPGAs, Int. J. Com-put. Sci. Inf. Technol. 5 (3) (2014) 3431–3433.

[3] F.O. Zamorano, J.M. Jerez, G. Juarez, J.O. Perez, L. Franco, High precision FPGA implementation of neural network activation functions, Intelligent Embedded Systems (IES), Orlando, FL. IEEE (2014) 55–60.

[4] B. Li, E. Zhou, B. Huang, Ji. Duan, Y. Wang, N. Xu, J. Zhang, H. Yang, Large scale recurrent neural network on GPU, Neural Networks (IJCNN), 2014 International Joint Conference on, 2014: 4062-4069

[5] R. Raeisi, A. Kabir, Implementation of Artificial Neural Network on FPGA, American Society for Engineering Education, Illinois, USA, 2006.

[6] TH Tsai, CH Yeh, YJ Huang, (2008), CORDIC Architecture for Hough Transform Applications, in IEEE Asia Pacific Conference on Circuits and Systems, pp. 442–445.

[7] Y. LeCun, MNIST Handwritten Digit Database, [Online]. Available: http://yann.lecun.com/exdb/mnist