International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 2, Issue 5(May 2013), PP. 20-23
www.irjes.com

Hardware Efficient Implementation of Neural Network

Vipin Tiwari[1], Shrikant Vyas[2], Nilay Khare[3]
[1](PhD Scholar, Department of Computer Science, MANIT, Bhopal)
[2](M.Tech. Student, Department of Computer Science, MANIT, Bhopal)
3(Faculty, Department of Computer Science, MANIT, Bhopal)

ABSTRACT : *Neural network is being used in many real life applications like prediction, classification etc. In this paper we present an FPGA hardware efficient implementation of neural network where we have implemented a multilayer feed forward network. We have used CORDIC algorithms of HP-35 scientific calculator for this purpose. The CORDIC algorithm is an approximation technique to compute trigonometric functions. Although a minor modification enables us to realize many useful functions. We have modified these algorithms to make them suitable for binary arithmetical operations.*

Keywords – *CORDIC, FPGA, Hardware, Neural Network.*

I. INTRODUCTION

An artificial neural network is interconnection of several processing units arranged in layered manner, where each processing unit is known as artificial neuron. Each neuron takes input from all or some neurons in the previous layer through weighted links, process the data taken as input and transfers it to next layer as output. These weighted links are referred to as synaptic strengths. This weighted summation is then transferred to a function often referred as activation function. Every neural network contains two or more layers of neurons, these layers may be fully or partially connected. Depending on the interconnection topology and data flow through the network, artificial neural networks can be classified as feed-forward or recurrent. Neural networks have been applied to problems such as classification, prediction and approximation.

Rest of the paper has been organized as follows. Section 2 gives the brief introduction of neural networks. Section 3 reviews previous approaches to hardware implementation of neural networks. Section 4 proposes new methodology to implement neural networks and section 5 shows results of experiments and finally section 6 concludes the work.

II. ARTIFICIAL NEURAL NETWORK

An Artificial Neural Network (ANN) is an information paradigm that is inspired by the biological nervous system such as brain. The key element of this paradigm is the novel structure of the information processing system. It is composed of large number of highly interconnected processing elements (neurons) working as union to solve specific problems. ANN are the type of Artificial Intelligence that attempts to imitate the way a human brain works rather than using a digital model in which all computations manipulate zero's and one's. A neural network is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects:

a. Knowledge is acquired by the network through a learning process.
b Inter neuron connection strength, known as synaptic weights, are used to store the knowledge.

2.1 NEURAL NETWORK ARCHITECTURE

The arrangement of neurons into layers and the pattern of connection within and in-between layer are generally called as the architecture of the net. The neurons within a layer are found to be fully interconnected or not interconnected. The number of layers in the net can be defined to be the number of layers of weighted interconnected links between the particular slabs of neurons. If two layers of interconnected weights are present, then it is found to have hidden layers. ANN come in many different shapes and sizes. One commonly used model of neuron is as shown in figure 1.

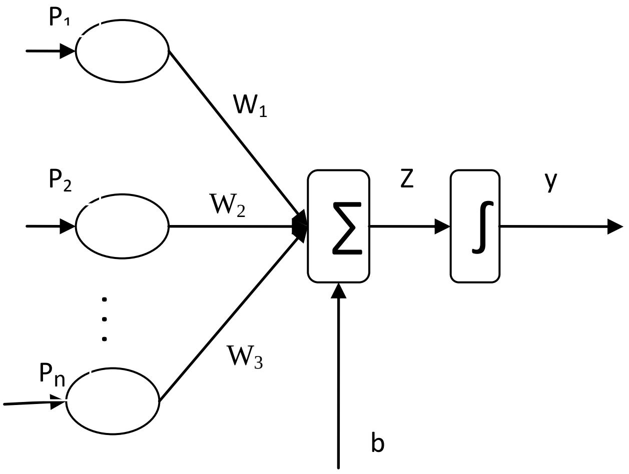

Fig.1 Architecture of a single neuron

In above figure the weighted sum often referred as net is fed into an activation function. This activation function is chosen on the basis of problem to be solved.

III. RELATED WORK

Since last two decades many researchers have been worked on neural networks and they have been implemented in both digital and analog platforms. Now much focus is being given to the FPGA based hardware realization of neural networks. In [1], an FPGA based pipelined implementation of back propagation network has been presented. In [3], authors have presented a hardware implementation of neural network with discretized activation function which is stored in look up table. [2] shows a MAC (multiplication and accumulator) based implementation of neural network using FPGA. In [4], an architecture REMAP has been introduced where highly parallel neural computer has been implemented using only FPGA's. In [8] spiking neural network has been implemented using FPGA, where the design of liquid state machine for speech recognition has been discussed. A distributed system for process and condition monitoring based on PIC microcontroller was developed in [9]. A high bit resolution FPGA implementation of feed-forward neural network with hyperbolic activation function was designed in [10].

Some analog hardware implementation of neural networks have also been reported which were very fast and low power in comparison to their digital counterparts. But analog implementations have some common problems like precision, robustness and learning capability. Analog implementation generally used for application specific purposes as they are very expensive and provide less flexibility. In [5], [6] and [7] some such analog implementation of NN have been presented. Although fast and hardware efficient networks can be designed using linear approximations but in that case it will be highly inaccurate.

IV. PROPOSED WORK

In this paper we have presented a digital implementation of neural network in hardware and time efficient manner using CORDIC architecture while maintaining high accuracy.

4.1 NUMERICAL REPRESENTATION

In this implementation we have used fixed point notation to implement decimal number. We used 32 bit numbers in input as well as for output, where most significant 11 bits are to store integer part of decimal number and remaining 21 bits for fractional part as shown in figure 2. We have also used some 64 bits registers to store Intermediate results of multiplication. In this way we achieved accuracy with hardware efficiency.

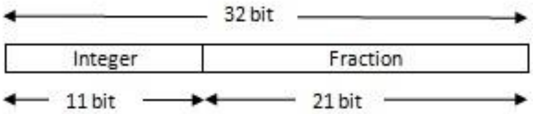

Fig. 2 FXP Representation of decimal number

This representation allows us to implement decimal calculations in hardware in efficient manner as compared to floating point representation.

4.2 ACTIVATION FUNCTION

In any hardware implementation of neural network the most expensive part is activation function. In this paper we have used log sigmoid function as an activation function. The expression for logsig function can be represented as:

$$\text{logsig}(x) = \frac{1}{1+e^{-x}}$$

For implementation of log sigmoid function in hardware linear approximation is a common practice. Although it can be easily done in less hardware but a significant flaw is its accuracy. Due to this reason we have used modified CORDIC algorithm of HP-35 scientific calculator which is well known first CORDIC algorithm based scientific calculator. In order to achieve our activation function we have used algorithms to calculate exponential and reciprocal of given number. As HP-35 works properly on BCD representation of decimal numbers, so we modified basic operations of these algorithms to work on binary arithmetic. In this way we have implemented our activation function in significantly less hardware resources.

4.3 ARCHITECTURE OF NEURAL NETWORK

To design a neural network in hardware first we have implemented a single neuron using only adders, shifters and registers for multiplication and addition to calculate net input to the neuron and then activation function as discussed in 4.2 using similar type of hardware resources. The architecture of neural network we have implemented is of 5-5-2. Here 5-5-2 means 5 inputs to 5 neurons of hidden layer and 2 output layer neurons

V. RESULTS

We have classified our results on the basis of four base criteria as time, accuracy, hardware usage and power consumption.

5.1 TIMING DIAGRAM

This implementation generates the results for the given inputs in approximate 49000 nano seconds on the rising edge of clock having period of 20 nano seconds. This clearly shows the results are obtained in significantly less time as compared to software approach.

5.2 ACCURACY

Our implementation gives highly accurate results suitable for verity of real life problems. It gives results correct up to 6 digits after decimal point. The results of our activation function are highly accurate results in significantly less hardware.

5.3 HARDWARE

We have targeted our implementation on vertex series 5, xc5vsx95t-3ff1136 device. This implementation uses approximately 5% of slice registers and approx. 6% of total LUT's. Similarly approx. 58% of total flip-flops and 40% of bonded I/O's. This clearly indicates the efficiency we have achieved while taking accuracy in consideration. Also it allows this device to be used in any portable device.

5.4 POWER CONSUMPTION

This is an important issue to be considered while designing hardware devices. This implementation consumes approximately one watt of power. The circuitry of design requires maximum voltage of 2.5 V which is equivalent to any portable device.

This clearly shows the efficiency of our design in terms of power consumption.

VI. CONCLUSION

In this paper we have seen that with the help of CORDIC algorithm we can implement a neural network with significantly less hardware resources while achieving highly accurate results.

REFERENCES

[1] Gadea, R. Ballester, F. Mocholi, A., & Cerda, J. Artificial neural network implementation on a single FPGA of a pipelined on-line backpropagation. In Proceedings of the 13th international symposium on system synthesis, IEEE, Los Alamitos (2000).

[2]]Nedjah, N., Martins, R. S., Mourelle, L. M., & Carvalho, M. V. S. (2009). Dynamic MAC based architecture of artificial neural networks suitable for hardware implementation on FPGAs. Neurocomputing (Vol. 72, pp. 2171–2179). Amsterdam: Elsevier (10–12).

[3] Canas, A., Ortigosa, E. M., Ros, E., & Ortigosa, P. M. (2008). FPGA implementation of a fully and partially connected MLP – Application to automatic speech recognition. In A. R. Omondi & J. C. Rajapakse (Eds.), FPGA Implementations of Neural Networks. Springer.

[4] Linde, A., Nordstrom, T., & Taveniku, M. (1992). Using FPGAs to implement a reconfigurable highly parallel computer. In Selected papers from: Second International Workshop on Field Programmable Logic and Applications. Berlin: Springer-Verlag (pp. 199–210).

[5] Choi, Y. K., Ahn, K. H., & Lee, S.-Y. (1996). Effects of multiplier output offsets on onchip learning for analog neuro-chips. Neural Processing Letters, 4, 1–8.

[6] Montalvo, A., Gyurcsik, R., & Paulos, J. (1997). Towards a general-purpose analog VLSI neural network with on-chip learning. IEEE Transactions on Neural Networks, 8(2), 413–423.

[7] Nedjah, N., Martins, R. S., & Mourelle, L. M. (2011). Analog hardware implementations of artificial neural networks. Journal of Circuits, Systems, and Computers, 20(3), 349–373.

[8] Benjamin Schrauwen, Michiel D'Haene, David Verstraeten, Jan Van Campenhout, Compact hardware liquid state machines on FPGA for real-time speech recognition, Neural Networks 21 (2008) 511–523

[9] Frankowiak, M.R., Grosvenor, R.I., Prickett, P.W., A petri-net based distributed monitoring system using PIC microcontroller. Microprocess. Microsyst. 29, 189–196(2005)

[10] Pedro Ferreira, Pedro Ribeiro, Ana Antunes, Fernando Morgado Dias, A high bit resolution FPGA implementation of a FNN with a new algorithm for the activation function. 0925 -2312/$ - see front matter r 2007 Elsevier B.V. All rights reserved. doi:10.1016/j.neucom.2006.11.028

[11] Ray Andraka, A survey of CORDIC algorithms of FPGA based computers.