Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol.9, No.7, 2018
www.iiste.org

Survey paper on Advanced Equipment Execution of ANN for FPGA

Saima Kanwal[1], Arslan Yousaf[1], Maria Imtiaz[1], Jalil Abbas[1*], Arslan Ali[2]

1. Department of Computer Science & IT, Govt. College University Faisalabad (Layyah Campus), Punjab, Pakistan
2. Department of Computer Science , Guizhou Normal University, Guiyang, China

* E-mail of the corresponding author: sjshah786@gmail.com

Abstract

Artificial intelligence is the area of computer science that aims at to create the intelligence machine. Artificial neural network is network that has different processing element. This survey paper recommends the implementation of Artificial Neural Network (ANN) in Field Programmable Gate Array (FPGA) and activates it through sigmoid function. This paper also proposes the implementation of new sigmoid function method in FPGA that combines the Look-Up Table (LUT) and Second Order Nonlinear Function (SONF). By this proposed method ANN works speedily, uses less resource and achieves high accuracy.

Keywords: ANN, FPGA, sigmoid function, look up table, second order nonlinear function.

1. Introduction

Artificial intelligence is advanced technology that covers a lots of disparate problem areas. It is also an area of computer science that aims at to create intelligent machine and now a days it has become an integral part of our industry. It provides a machine that works and acts like a being human and it is special designated speech recognition, learning planning and solves problem[19]. However, the most advanced interesting areas is sensor controlled behavior, in which a machine using sonar's and cameras collects the information in the real world.

Firstly, artificial neural network came in 1943, as a neuropsychologist Waren McCulloch a mathematician, Walter Pittes worked on how neurons works. A neural network is a system of hardware or software patterned like same neurons in the human brain [1]. Neural network is also called artificial neural network. The major and cardinal feature of artificial neural network is the ability to learn and observe from environments. Artificial neural network is the first topology that described as how neurons are interconnected with each other and how they transmit information from one to another. It also describes the strength of inter neurons change with external output [1] [2].

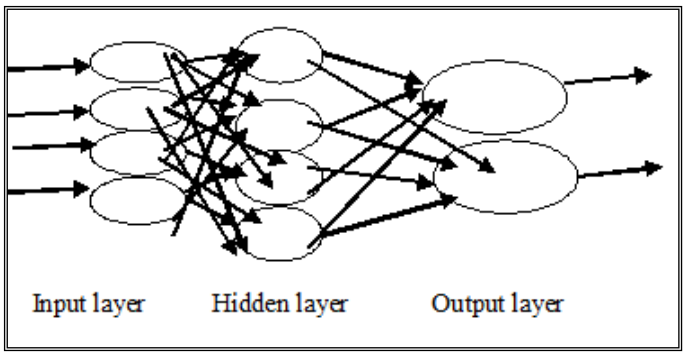

Figure 1.1: Structure of Neural Network

Neural network consists of layers in its depth, and describe as how many layers they have between input and

output, this is called hidden layers. It can also be described by the numbers of hidden nodes in terms as how many input and output each node is consisted of.

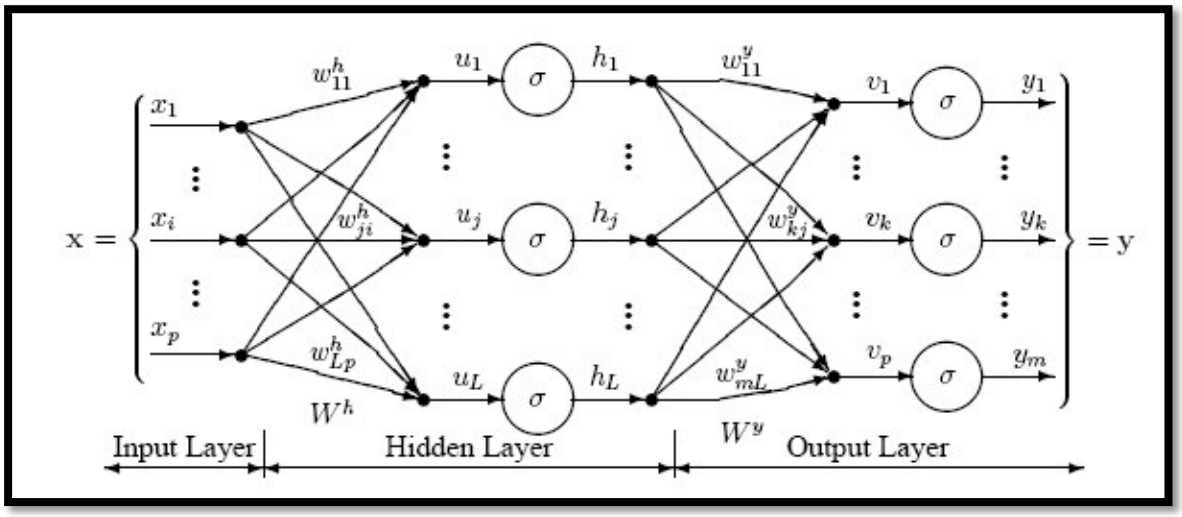

Figure 1.2: Three Layers of ANN

It usually consists of multi-layer perception network. Artificial neural network mostly consists of three layers which are inter-connected with each other. The first layer consists of input neurons and sends data to the next layer. The second layer receives the input and sends in turn to the third layer [17].

In forward propagation, we get the output and compare it with real value. But a back propagation is the method used in ANN [18]; after data is processed it checks each neuron and calculates the error at the output and distributes back through the network layers. Back propagation is also called backward propagation of error [5].

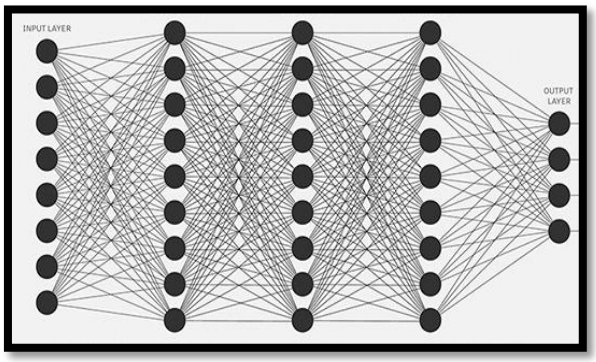

Figure 1.3: Deep Neural Network

2. Characteristics of Artificial Neural Network

A very large number of simple neural network includes in processing elements. ANN is the first topology that describes how neurons are interconnected with each other and how they transmit information from one to another. It also describes how the strength of inter-neuron change with external output. And important feature of ANN is ability to learn and observe from the environments [4]. Safety, accuracy, reliability used for safety in different environments.

3. Usage of Artificial Neural Network

The primary purposes behind utilizing an Artificial Neural Systems are as per the following. It provides Massive Parallelism. A distributed representation of any framework can be produced with upgrade learning ability and generalization ability of the system. It is also used in Robotics, auto driving, auto pilot and different type of environments. It has been used in many fields

to solve the varies problem mostly speed estimation [11].

4. Training Method of Neural Network

Once the neural network is constructed, the next step is to train the network this is done by training method. Two approaches are used for training the network. One of them is supervised training and other is unsupervised training.

4.1 Supervised Training

Supervised training deals with both input and output, in which network processes the input and compares its resulting output with desired output. Errors occur then propagated back by adjusting the weights through which network is controlled [14]. This process is repeated again and again as the weights are continually tweaked. The commercial network development package provides such tools which define the coverage of ANN and define how ANN predicts the right answers. These tools stop working when system reaches some statistically desired point. If a network doesn't solve this problem then designer reviews the I/O, Number of layers, connection between layers and some changes, which are the art of ANN. When the system trained correctly then no further changes are required, and this weight will be frozen.

4.2 Unsupervised Training

This unsupervised training consists of input but not desired output. When some particularities like identifying subsets or cluster found between elements then network needs to be organized itself. The learning machine adjusts the weight in order to reflect the clusters within the network [20].

5. Implementation of ANN

ANN implements in both software and hardware but hardware implementation is better than software [12]. With software implementation we can't gain the maximum efficiency of ANN but with the hardware we can achieve the maximum efficiency because the processor works with the sequential instruction in software implementation of ANN but in hardware ANN works with parallelism. Another reason to implement ANN in software is not portable but in hardware it is portable. Hardware implementation of ANN gaining importance [10].

The ANN can be implemented using the Analog system or digital system. The digital implementation is much better than analog because of its higher accuracy, better reliability, lower noise sensitivity, better testability, higher flexibility and compatibility with other types of preprocessor[15] [16].

The digital implementation of ANN further classified as

i) FPGA (Field programmable Gate Array)
ii) DSP (Domain specific part)
iii) ASIC (Application specific integrated circuit)

5.1 The Proposed Implementation of ANN

We have three ways to implement the ANN in hardware but FPGA is better than other three because DSP work sequentially but ANN work parallel so we can't get the maximum efficiency with DSP and ASIC doesn't not offer the reconfiguration and its design is complex for implementation. However implementation of ANN into FPGA is not easy task like implementation in software it requires complex calculations and involving a large number of neurons [7].

FPGA provides the reconfiguration and offer the portability. FPGA is completely programmable so it replaces the microprocessor [8]. Moreover its work is parallel, with the parallel architecture of ANN the certain task complete faster you ever observe [9].

5.2 Activation Function

We cannot easily implement the ANN in FPGA like in software because it involves a large number of neurons and complex calculation like activation function [15]. The activation function is the function that activates the neuron. There are many activation functions available. Generally we divide them into Two.

First is the linear activation function and the second is nonlinear activation function.

We use the nonlinear activation function in FPGA. The higher accuracy and precision must be in activation function because if any error occurs in output of any neuron activation function it will amplified in the other neurons [16].Optimal configuration achieved by using sigmoid function [6].

We use the nonlinear activation function because it satisfies the specification of training algorithm. Again there are three types of commonly used activation function for implementation of FPGA. First is hard limit activation function, second is saturating linear activation function and third is sigmoid activation function.

Table 1: Some Important Activation Function

Name	Function
Sigmoid function	$f(x) = \frac{1}{1+e^{-x}}$
Linear function	$f(x) = x$
Hard limit function	$x = f(a)\begin{cases} 0, & if\ a < 0 \\ 1, & if\ a \geq 0 \end{cases}$
Saturating linear function	$x = f(a)\begin{cases} -1, for & a < 0 \\ a, for & -1 \leq a \leq 1 \\ 1, for & a \geq 1 \end{cases}$

5.2.1 Hard Limit Activation Function

In this if the function is less than zero then its output is zero and if function input is greater than zero then its output is 1.

$$x = f(a)\begin{cases} 0, & if\ a < 0 \\ 1, & if\ a \geq 0 \end{cases}$$

5.2.2 Saturating Linear Activation Function

It is also called piecewise linear function. It either be bipolar range or the binary. Its means that if the function is less than 0 then output will be -1, if the function is in between the 1 and -1 then output will be same function and if the function is greater and equals to the 1 then output will be 1.

$$x = f(a)\begin{cases} -1, for & a < 0 \\ a, for & -1 \leq a \leq 1 \\ 1, for & a \geq 1 \end{cases}$$

5.2.3 Sigmoid activation function

In whatever your input its map between 0 and 1.

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

5.2.4 The proposed activation function

We choose the sigmoid function for implementation of neural network in FPGA because sigmoid function provides the smooth transaction between input and output and sigmoid function improves the response time of the neural network. Sigmoid function also used in back propagation algorithm.

5.2.5 Sigmoid Function Implementation

There are many ways to implement the sigmoid function in hardware that are look-up table (LUT), coordinate rotation digital computer (CORDIC) algorithm, piecewise linear approximation and piecewise nonlinear approximation[3].

5.2.6 Look-Up Table (LUT)

The easy and simple way to implement the sigmoid function is the use of a LUT. Many implementations used in a LUT. In this method the value curve of a sigmoid function is divided into many segments and it stores into the table. Because it divides into different parts hardware memory is require for LUT. If we need high accuracy we should divide the value into very small segments or more segments.

5.2.7 CORDIC Algorithm

This algorithm purposed by JACK.E in 1959. It is commonly used for FPGA. This is another alternative to calculate the sigmoid function. The basic idea of CORDIC algorithm is rotating the vector on the plane that is to start from the initial position until it coincides with desired target position. The vector rotates by using the fixed number of angular steps which is executes in sequence. The problem with CORDIC is it inherently a sequential approach.

5.2.8 Piecewise Linear Approximation

Three PWL methods' recommended by the scientists. Which is A-law based computing technique, PWL of nonlinear function (PLAN) and recursive algorithm based on the centered recursive interpolation (CRI) [13].All the recommended methods obtain the low value of both average error and maximum with low computing complexity. In A-law based method the gradient of each linear segment is expressed by power of two. Meanwhile the PLAN approximation divides the segment function into 5- segments. The CRI method improves the accuracy recursively.

5.2.9 Piecewise Nonlinear Approximation

In general, piecewise nonlinear approximation used the equation to calculate the sigmoid function.

$$f(x) = c_0 + (c_1 + c_2 * x)$$

Another method is a simple second order nonlinear function (SONF) to approximate the sigmoid function that has been presented by the Zhang. Zhang divides the sigmoid function into 4-segments. This method operates faster than PWL and CORDIC but this method gives the limited accuracy.

LUT and CORDIC function show the higher accuracy but LUT needs higher hardware resources and CORDIC function requires the huge amount of computation time.

PWL and piecewise nonlinear approximation require the less hardware resources and less amount of the time to complete its operation but they can't achieve the higher accuracy.

6. The Proposed Method

This proposed method we present the new sigmoid function method for implementation. The LUT achieves the higher accuracy and SONF need less resources, operates fast. In this proposed method LUT and SONF are combined. Therefore, the method uses fewer resources, gives higher accuracy and performs operation fast.

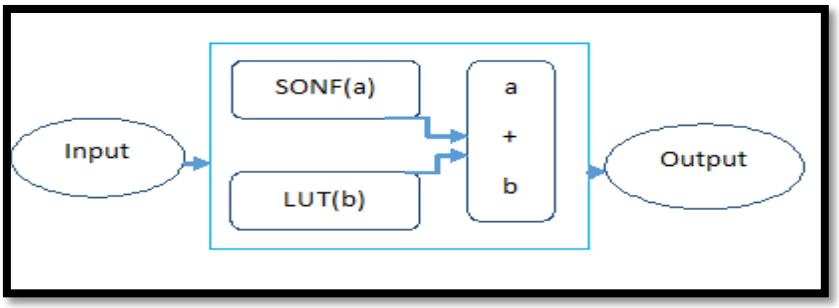

Figure 6: Implementation of Proposed Method

By using this method, high accuracy and speed can be achieved because it uses less clock cycle and less amount of memory.

Table 2: Implementation of Sigmoid Function

Implementation	Clock cycle	Memory use
LUT	3	16kbits
CORDIC	50	0
sSONF	10	0
Purposed method	13	320bits

Conclusion

This research paper discusses about ANN, working of ANN and training of ANN. This also compares the different method of implementation of ANN and proposed ANN implementation in hardware because it is more efficient. ANN can implement in ASIC, DSP and FPGA. The proposed method is FPGA because it is parallel, portable and reconfigurable. FPGA can be activated by different activation functions but proposed method is sigmoid function because sigmoid function improves the response of neural network. For implementation of sigmoid function in FPGA the recommended method is new sigmoid function method which combines the LUT and SONF through which neural network operates fast, achieve higher accuracy and uses less resources. This future proposed idea can be used in other hardware implementation of ANN. Because of its high speed and accuracy this purpose method work effectively where there is need of quick response like robotics.

REFERENCES

[1] H. Namin, K. Leboeuf, R. Muscedere, H. Wu, and M. Ahmadi. 2009. Efficient hardware implementation Of the hyperbolic tangent sigmoid, function. IEEE Int.Symp. Circuits Syst., pp. 2117–2120.

[2] S. Hariprasath and T. N. Prabakar. 2012. FPGA Implementation of Multilayer Feed Forward Neural Network Architecture using VHDL. In: International Conference on Computing, Communication and Applications (ICCCA), pp. 1–6.

[3] Z. Xie. 2012. A non-linear approximation of the sigmoid function based on FPGA In: IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI), pp. 221–223.

[4] J. Juang, L. Chien, and F. Lin. 2011. Automatic Landing Control System Design Using Adaptive Neural Network and Its Hardware Realization.System Journal, vol. 5, no. 2, pp. 266–277.

[5] Sahin and I. Koyuncu. 2012. Design and implementation of neural networks neurons with radbas, logsig, and tansig activation functions on FPGA. Electron. Electr. Eng., vol. 4, no. 4, pp. 51–54.

[6] T. G. Tan, J. Teo, and Patricia Anthony. 2014. A comparative investigation of non-linear activation functions in neural controllers for search-based game AI engineering. Artif. Intell. Rev., vol. 41, no. 1, pp.1–25.

[7] E. Z. Mohammed and H. K. Ali. 2013. Hardware Implementation of Artificial Neural Network Using Field Programmable Gate Array. Int. J. Comput.Theory Eng., vol. 5, no. 5, pp. 780–783.

[8] K. P. Lakshmi and M. Subadra. 2013. A Survey on FPGA based MLP Realization for On-chip Learning.Int. J. Sci. Eng. Res., vol. 4, no. 1, pp. 1–9.

[9] L. Kim, S. Asaad, and R. Linsker. 2014. A Fully Pipelined FPGA Architecture of a Factored RestrictedBoltzmann MCHINE Artificial Neural Network. vol.7, no. 1.

[10] T. M. Jamel and B. M. Khammas. 2012.Implementation of a Sigmoid Activation Function for Neural Network Using FPGA. In: 13th Scientific Conference of Al-Ma'moon University College.

[11] T. Orlowska-Kowalska and M. Kaminski. 2011.FPGA implementation of the multilayer neural network for the speed estimation of the two-mass drive system. IEEE Trans. Ind. Informatics, vol. 7, no.3, pp. 436–445.

[12] Rafid Ahmed Khalil Sa'ad Ahmed AI-Kazzaz "Digital Hardware implementation of Artificial Neurons models using FPGA" , Springer U.S.,2012.

[13] Hiroomi Hikawa , "A Digital Hardware Pulse-Model Neuron With Piecewise Linear Activation Function,"*IEEE Trans.Neural Networks, vol.14,no.5*,pp.1028-1037,Sept.2003

[14] R. Sathya and Annamma Abraham, "Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification", International Journal of Advanced Research in Artificial Intelligence (IJARAI), Vol. 2, No. 2, pp. 34-38, 2013.

[15] Ngah, S., & Bakar, R. A. (2017). Sigmoid Function Implementation Using the Unequal Segmentation of Differential Lookup Table and Second Order Nonlinear Function. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, *9*(2-8), 103-108.

[16] Muthuramalingam, A., Himavathi, S., & Srinivasan, E. (2008). Neural network implementation using FPGA: issues and application. *International journal of information technology*, *4*(2), 86-92.

[17] Thilagavathy, A., & Kanth, K. V. (2013). Digital Hardware Implementation of Artificial Neural Network for Signal Processing. *International Journal of Scientific & Engineering Research*, *4*(3).

[18] Akkar, H. A., & Ali, A. H. Design and Implementation of Artificial Neural Networks for Mobile Robot based on FPGA.

[19] Maung, H. M. Y., Tun, H. M., & Naing, Z. M. Implementation of Neural Network Algorithm for Face Detection Using MATLAB.

[20] Maind, S. B., & Wankar, P. (2014). Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication*, *2*(1), 96-100.