A NEW HARDWARE ARCHITECTURE FOR FPGA IMPLEMENTATION OF FEED FORWARD NEURAL NETWORK

V.A.Sumayyabeevi[1], JaimyJames Poovely[2], Anju George[3]

Abstract

New chips for machine learning applications appear, they are turned for specific topology being efficient by using highly parallel designs at the cost of high power or large complex devices. Although, the computational demands of deep neural networks require flexible and efficient hardware architectures able to fit different applications, neural network types, number of inputs, outputs, layers and units in each layer , making the conversion from software to hardware easy.

Keywords: *Feed forward neural networks - FFNN, systolic hardware architecture, FPGA implementation*

Introduction

A neural network is a network or circuit of neurons and an artificial neural network is composed of artificial neurons. Thus a neural network is either a biological neural network, made up of biological neurons, or an artificial neural network for solving artificial intelligence problems. Artificial intelligence is the intelligence exhibited by machines, in contrast to the natural intelligence displayed by humans. Therefore artificial neural networks are inspired by, but not identical to biological neural network that constitute animal brains. Such systems perform tasks by considering examples without being programmed with task-specific rules.

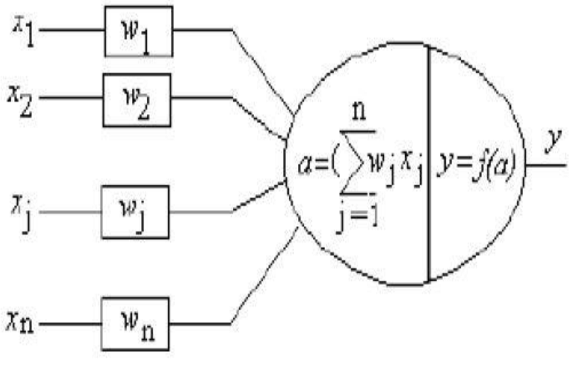

Fig. 1. Classification of Neural Networks

[1] First year M.Tech VLSI and Embedded Systems student in APJ Abdul Kalam Technological university. Kerala. She received her B.tech degree in Electronics and Communication from University of Calicut Kerala.

[2] Asst. Professor in the Department of Electronics & Communication at Adi Shankara Institute of Engineering & Technology, Kalady. She received her B.Tech in Electronics and Communications from MG university.

[3] Asst. Professor in the Department of Electronics & Communication at Adi Shankara Institute of Engineering & Technology, Kalady. She received her B.Tech in Electronics and Communications from MG University.

Received: 26 Apr 2020 | Revised: 24 May 2020 | Accepted: 28 Jun 2020

In an artificial neuron, the most fundamental computational unit is modeled based on the basic property of a biological neuron. This type of processing unit performs in two stages: weighted summation and some type of non- linear function. It accepts a set of inputs to generate the weighted sum, the passes the result to the non-linear function to generate output. The artificial neuron presented in the figure1 have „n inputs denoted as w1, w2, w3. . . .wn respectively. The activation function, which determines whether the neuron is to be fired or not, is given by the formula: The output is y = f(a) An ANN system consist of a number of artificial neurons and huge number of interconnections among them.

IMPLEMENTATION OF ANN[4]

A large amount of work has been done for developing simulation environments for ANNs on sequential machines. ANN can be implemented in two ways. One is software simulation and the other is hardware approach like FPGA implementation.

Table1: Comparison between software and hardware approach for the implementation of ANN[5,6,7]

Aspect	Software Simulation	Hardware Approach
Calculation Complexity	High for large net-work	Low
Execution time	Increases exponentially	Relatively lower
Interpretation of Results	Difficult to interpret results	Easy to interpret results
Cost	No margins for reducing system cost	Provide margins for reducing system cost
Graceful degradation	Stops functioning when there is faults in the system	Allow applications to continue functioning through reduced performance

The performance of conventional von-Neumann processors, for example the Intel Pentium series, continues to improve dramatically. When the particular task at hand does not require super-fast speed, most designers of neural network solutions find a software implementation on a PC or work station with no special hardware add-ons a satisfactory solution. However, even the fastest sequential processor cannot provide real time response and learning for networks with large number of neurons and synapses. Parallel processing with multiple simple processing elements on the other hand can provide tremendous speedups. Some specialized applications uses of hardware neural networks. When implemented in hardware , neural network can take full advantage of their inherent parallelism and run orders of magnitude faster than software simulations. A large number of hardware architectures have been proposed for the implementation of the ANN. ANN may be realized using analog systems as well as the digital systems. In addition some of the existing platforms available for the hardware implementation ANN are Digital Signal Processing (DSP) chips, Application Specific Integrated Circuits (ASICs), Graphical Processing Units (GPU), or Field Programmable Gate Array(FPGA).

HARDWARE IMPLEMENTATION OF ANN

Hardware devices designed to realize artificial neural net work(ANN) architectures and associated learning algorithms especially taking advantages of inherent parallelism in the neural processing are referred as hardware neural networks. Specialized ANN hardware (which either support or replace software) offers appreciable advantages mentioned as follows: 1.Speed: Specialized hardware can offer very computational speed 2.Cost : Reduce system cost by lowering total component count and decreasing power requirement. 3.Graceful degradation: Allow applications to continue functioning though with slightly reduced performance(graceful degradation) even in the presence of faults in some component.

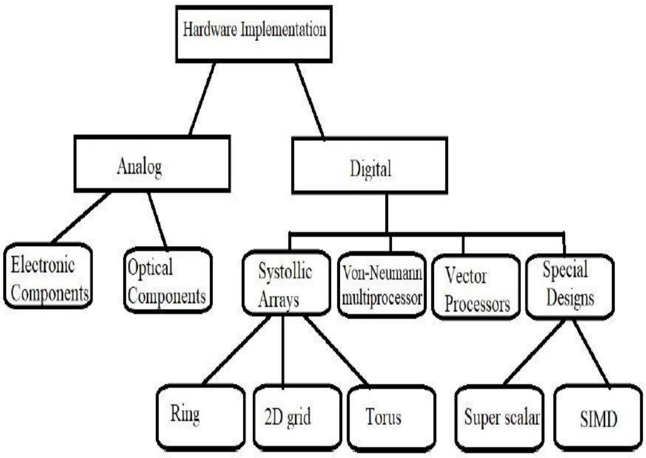

Fig:2 Classification of hardware implementation of ANN

Table.2 Comparison between analog and digital implementation of ANN[10,11]

Analog	**Digital**
Signals are represented by magnitude of current, voltage and frequency	Signals are represented as digital or discrete values
Implemented with electronic or optical components networks are distributed in microprocessor system	Implement weight using shift registers, latches etc
Memory elements are capacitors, transistors, charge coupled devices	Memory elements are latches, flip flops etc
Noise effect is more	High signal to noise ratio
Cost may increase	Cheap fabrication
Less power	Possibility of working with virtual network

Received: 26 Apr 2020 | Revised: 24 May 2020 | Accepted: 28 Jun 2020

Complex	Complexity reduced
Less precision	Greater precision
Dependent on components	Flexibility is more

Digital implementation of ANN[12,13]

In digital neuro computers signals and network parameters are encoded and processed digitally. The precision of the circuits can increased by increasing the word length of the machine. Analog neuro computers are affected by the strong variation of the electrical characteristics of the transistor, even when they have been etched on the same chip. According to the kind of analog component the arithmetic precision is typically limited to eight to nine bits. There is another reason for the popularity of digital solutions in the neuro computing field. Present computers work digitally in many cases the neural network is just one of the pieces of a whole application. Having everything in the same computer makes the integration of the software easier. This has led the development of the several boards and small machines connected to a host as a neural coprocessor. In a digital neuron synaptic weights are stored in shift registers, latches or memories. A digital implementation entails advantages like simplicity, high signal-to-noise ratio, easily achievable, cascadability and flexibility, and cheap fabrication, along with some demerits like slower operation.

Advantages

- A digital neural network are robust for drift, mismatch, noise etc
- A digital network can be generated from a logic description of its function
- Loading digital weights is relatively easy, no feedback is required
- Digital circuits scale very well with new processors, and virtually no redesign is required

Architecture for digital neural network

Artificial neural network exhibit high level of parallelism in computations. In parallel computing many calculations or the execution of processes are carried out simultaneously. Large problems are divided into smaller ones and each of them solved at the same time. There are several forms of parallel computing:

1. Bit-level parallelism
2. Instruction or data level parallelism
3. Task parallelism Based on this parallel computation methods the architectures of digital neural network can be classified into:

1. Bit-slice
2. Single instruction multiple data (SIMD)
3. Systolic arrays

FPGA IMPLEMENTATION OF ANN [14,15,16,17]

FPGA is an integrated circuit that can be configured by customers after manufacturing. It contain an array of programmable logic blocks and a hierarchy of reconfigurable interconnects that allow the blocks to be wired together like many logic gated that can be inter wired in different configuration. Artificial neural networks (ANN), with numerous applications typically running under PC based software system. However, when fast processing time is required in real time applications or fast prediction, decision or classification, a PC based system might not be able to provide enough throughputs. Nowadays, this situation becomes very common since ANN size are growing due to the complexity of the problems to be solved and big data applications, with an increasing number of inputs, neuron units, and the number of layers. Moreover, a power consumption and computational speed is an important issue; CPUs and GPUs can process data at a high speed, but the use of power and resources is higher than FPGA and other custom embedded hardware platforms.

Table 3. Comparison between various digital neural network architectures

Bit-slice	**SIMD**	**Systolic arrays**
Technique for constructing a processor from modules of processors of smaller bit width	Describes computers with multiple processing elements that performs the same operations on multiple data points simultaneously	Homogenous network of tightly coupled data processing units called cells or nodes
Operations and operands are bit related	Exploits data level parallelism but not concurrency	The parallel input data flows through an network of hardwired processor nodes
Provide cheap and simple building blocks for constructing large networks	Large register files, so area and power consumption is high	Faster than general purpose processors and scalable

Advantages of FPGA implementation

- FPGA device is a good candidate to be used as an independent device, receiving inputs directly form the process, computing them and sending the output to a real process
- FPGA devices are one of the best options for the hardware implementation of ANN since required computation are based on the sum of products, which can fit very well into the FPGA internal slices
- Use of FPGA devices allows the parallelization of neural networks by using concurrent computing of multiple units
- Reconfigurable FPGAs provide an effective pro- grammable resource for implementing hardware neural network
- Reconfigurable FPGAs provide an effective pro- grammable resource for implementing hardware neural network
- FPGA allow different design choices to be evaluated in very short time
- FPGA based implementations are low cost, readily available
- Offer software like flexibility
- Provide high throughput gain than that of software implementation

Proposed FPGA Architecture[18]

By considering typical FPGA resources, a novel hardware architecture can be proposed. It is versatile and universal systolic massive parallel architecture (SYMPA) for feed for- ward neural networks, based on computationally independent neural processing elements. The resulting hardware structure is a combination of fine grained and coarse grained with parallel input processing and time multiplexed input. The SYMPA architecture allows the implementation of arbitrary size and arbitrary type FFNN. The proposed architecture can adopt any type of FFNN like multilayer perceptron(MLP), auto encoder(AE), Logistics regression(LR). It can scale up to arbitrary size, only limited by the available resources. A single activation function block(AFB) is required for whole FFNN and provide great versatility.

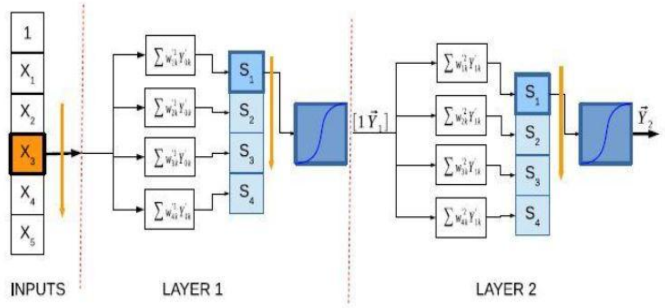

Fig. 3. Computation Procedure in the proposed architecture

Figure3 graphically shows the proposed layer wise parallel feed forward architecture for a 544 FFNN. Initially, the inputs in a layer are serially processed by the layer computation blocks, feeding one input each clock cycle . After obtaining the results of the units in one layer, its layer output values (s1,s2. . . . . .) are stored in a memory and the same hardware can be iteratively reused to compute all layers. It is important to note that the stored values si correspond to the sum of products result without activation function evaluation. It is only before entering in to the next layer that they are evaluated by the activation function. Applying this mechanism, together with the serial input processing, a single activation function block can serve for all the FFNN structure since one value per clock cycle is used, no matter the FFNN size.
Computation procedure for the proposed layer wise parallel feed forward architecture with serial input is shown in the figure 3. The input vector x is serially entered and processed by the array of multiply and accumulators to calculate the weighted sum of products. As the output layer values S are generated, they enter the activation function block, generating Y vector as the output of the layer. When one layer is finished, the computation is repeated for the next layer.

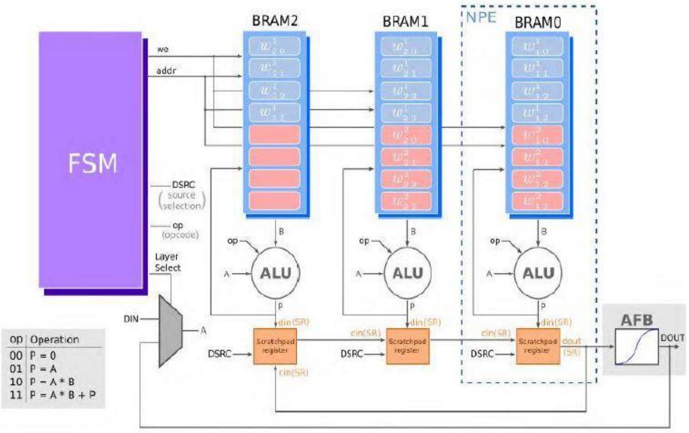

Fig. 4. Artificial Neuron Model

The proposed architecture, shown in figure is a generic architecture that can be arbitrarily extended in number of layers and units per layer as long as hardware resources are available. Since the layers reuse the hardware, the main limiting factor is the number of units in the biggest layer, and number of input. Each NPE act as a neuron unit of the FFNN for each layer computation. At most one NPE will be reused as many times as layers exist in the FFNN. Weight values corresponding to all units that will be computed by the NPE are stored in its internal local memory. Each NPE contains the weight values of one unit per layer, at most. The NPE architecture is designed to accumulate the partial sum of products of the current unit in the ALU accumulator register, and, when finished, the resulting value is moved into the corresponding SR register so that they can be shifted through the SR chain into Activation function block (AFB), and, simultaneously , the NPEs can compute the next layer values by feeding back the AFB output values to NPEs. The order of the operations is defined by finite state machines (FSM) block. The data input in the system is serially performed through DIN port.

CONCLUSION

For implementing artificial neural network, two methods are available: one is software simulation and other is hard- ware approach. From this two, hardware implementation is preferred due to its less complexity and easiness in the analysis of results. For the hardware implementation of ANN, two techniques are available: one is analog implemen- tation and other id digital implementation. From this digital implementation is preferred. Incorporating FPGA into the digital implementation of ANN, will provide an excellent ANN .This implementation is preferred because of high speed and re-configurability. For accelerating the FPGA implementation a novel architecture is proposed. Which can implement any feed forward neural network like multilayer perceptron, auto encoder and logic regression. Its systolic nature and pipelined design make it possible to obtain linear scalability in resource occupation. The architecture uses a single activation function for the whole FFNN, apart from the obvious resource savings, this fact allows customization op- tions for the activation function model. It combines concepts from matrix computation fundamentals, mixed serial parallel computer architecture, and specific hardware availability in current FPGA devices as ALUs and distributed as RAM.

REFERENCES

[1] Botros, N. M., Abdul-Aziz, M. (n.d.). Hardware implementation of an artificial neural network. *IEEE International Conference on Neural Networks*,1993

[2] Uhrig, R. E. (n.d.). Introduction to artificial neural networks. *Proceedings of IECON '95 - 21st Annual Conference on IEEE Industrial Electronics.doi:10.1109/iecon.*1995

[3] Yihua Liao"Neural network in hardware: A survey," Department of computer science University of California, Davis One Shields Avenue, Davis, CA 95616liaoy@cs.ucdavis.edu

[4] Botros, N. M., Abdul-Aziz, M.Hardware implementation of an artificial neural network using field programmable gate arrays (FPGA's). ,1994

[5] M. Minsky and S. Papert, Perceptrons, The MIT Press, Cambridge, MA,*Proceedings of the 2004 American Society for Engineering Education Annual Conference Exposition Copyright* 2004

[6] Y,Zhou,W.Wang and X Huang, "FPGA design for PCANet, deep learning network", in Proc.*IEEE 23rd Annu.Int.Symp.Field-program custom* comput Mach,*May* 2015

[7] V. Calayir, T. Jackson, A. Tazzoli, G. Piazza, and L. Pileggi, "Neurocomputing and associative memories based on ovenized aluminum nitride resonators," in Neural 27 Networks (IJCNN), *The International Joint Conference on. IEEE*, 2013

[8] Misra, J., Saha, I. (2010). Artificial neural networks in hardware: survey of two decades of progress. Neuro computing,2010

[9] J. Tang, M.R. Varley and M.S. Peak," hardware implementations of multi-layer feed forward neural networks and error backpropagation using 8-bit pic microcontrollers"with the Department of Electrical and Electronic Engineering,University of Central Lancashire, *Preston PRl 2HE*, United Kingdom.

[10] Morales Morales, C., Flores, U., Adam Medina, M., Diaz Salazar, M., Abiel Caballero,J., Criado Cruz, D., Pavoni Oliver, S.Digital Artificial Neural Network Implementation on a FPGA for data classification. *IEEE Latin America Transactions,*, 2015

[11] Mada, S., Mandalika, S.Analog Implementation of Artificial Neural Networks Using Forward Only Computation. *Asia Modelling Symposium (AMS)*,2017

[12] Artificial neural networks in hardware: A survey of two decades of progress Article in neuro computing December 2010

[13] Lu, L., Liu, W., O'Neill, Swartzlander, E. E. (2013). QCA Systolic Array Design.*IEEE Transactions on Computers*,2011

[14] Sahin, S., Becerikli, Y., Yazici, S. (2006). Neural Network Implementationin Hardware Using FPGAs. Lecture Notes in Computer Science,2006

[15] Zhu, J., Sutton, P.FPGA Implementations of Neural Networks –A Survey of a Decade of Progress. Lecture Notes in Computer Science,2003

[16] Iranpour, E., Sharifian, S. (2016). An FPGA implemented brain emotional learningintelligent admission controller for SaaS cloud servers. Transactions of the Instituteof Measurement and Control, 2016

[17] Saichand, V., M., N. D., S., A., Mohankumar, N.FPGA Realization of Activation Function for Artificial Neural Networks.*Eighth International Conference on Intelligent Systems Design and Applications.*,2008

[18] Leandro d. Medus, taras iakymchuk , jose vicente frances-villora,manuel batallermompe ´an , and alfredo rosado-mu˜noz, "A Novel Systolic Parallel Hardware Architecture for the FPGA Acceleration of Feedforward Neural Networks," in *Proc.IEEE Annu. Int. Symp. Field-Program. Custom Comput. Mach.*, May 2019