Backpropagation Algorithm and its Hardware Implementations: A Review

Shivani Kuninti, Rooban S
Department of Electronics & Communication Engineering,
Koneru Lakshmaiah Education Foundation, Vijayawada. A.P., India.
shivani.ry@gmail.com

Abstract. Nowadays, artificial intelligence is gearing up at faster pace. Hardware integration with Artificial Neural Networks (ANN) have paved the way to different applications in areas like control engineering, robotics and navigation. There has been extensive research in the field of machine learning to use the system in advanced applications. The combination of hardware and software modules help to build systems in a robust manner. This paper focuses mainly on back propagation algorithm and its different hardware implementation using FPGA, ASIC, Memristor and Microcontroller. Implementation of back propagation algorithm on Field programmable gate arrays is analysed in detail. FPGAs parallel processing feature makes it unique for neural networks. The parallel processing models implement back propagation algorithms for calculating errors in the hidden layers of neural networks. This review helps the researchers to understand the implementation of various hardware with back propagation algorithm also the comparative analysis on the parameters helps to identify the suitable hardware based on their requirements.

1. Introduction
Artificial intelligence is seen in most of the software technologies and hardware implementations. The artificial neural networks form the basis of modern era applications. ANNs mimic the nervous system of the human body. It is similar to the neurons in brain. As the neurons in the brain are connected by different nerves, artificial neural networks have different nodes which are connected by rules. As shown in Figure 1, a simple neural network has two inputs, hidden nodes, and two outputs. The nodes can be an input, output or hidden. The neuron nodes are connected by rules or algorithms.

ANNs incorporate many algorithms for different kind of applications [1]. Algorithms are implemented by following specific rules of ANN. Different algorithms have been devised to satisfy the required specific feature. Depending on the functionality or feature, many interconnected layers are present. The hidden nodes are interconnected in the form of a pattern and rules are applied in the form of equations in feed forward networks, and these equations are iterated multiple times. Feed-forward networks follow delta rule. Errors occurred at the output nodes are calculated and these errors are back propagated and distributed across the hidden nodes. Back propagation algorithm is one of the algorithms extensively used in ANNs. The back-propagation algorithm can be implemented in hardware using FPGA or ASIC or memristor or microcontrollers.

Figure 1 Simple Neural Network
Any neural structure depends on the performance and stability of the system. Machine learning configures many algorithms to implement different functionalities on hardware. Few functionalities can be done on software and few functionalities on hardware modules. Combination of both software and hardware modules improve the system capabilities like speed, performance, area utilization, cost and memory. This paper is divided into five sections. Section II and III describes the artificial intelligence and back propagation algorithm respectively. Section IV described the hardware implementation and discusses the comparative analysis. Section V concludes the review work on hardware implementation.

2. Artificial Intelligence

Neural network follows a specific pattern or implements a learning rule. In smaller networks, when the inputs nodes are activated, synaptic weights are updated and they are forwarded to output nodes. These kinds of synapses require to follow the rules [2]. There are different training algorithms are available like back propagation, genetic and krill herd algorithms [2], [3].

Calculations of synaptic weights requires the training of hidden nodes through functions, rules, and algorithms. When input nodes are excited, information is processed by the hidden layers and the processed information is given to output nodes. Back propagation algorithms are used in machine learning applications.

Machine learning (ML) is a branch of artificial intelligence. In modern era of software, machine learning depends on prediction of data sets based on experience. It employs various algorithms for different modules in the software.

ML is one of the approaches to Artificial Intelligence which has applications in many areas including EDA.[4] Moreover, ML takes samples from different methods and dynamically builds timing models automatically. [4] Machine learning employs advanced algorithms with complex data sets and it analyses data which are multi-dimensional in nature. It forms a known pattern or unknown pattern from multidimensional data. [4] Back propagation algorithm is extensively employed in machine learning in feed forward networks.

3. Back Propagation Algorithm

Backpropagation (BP) algorithm has extensive applications in artificial neural networks.[5] An algorithm can be executed in software or in hardware. The complexity of algorithms execution in software increases, when the modules require parallel processing, and it takes more iterations. When parallel processing is required [6], implementation of algorithm on hardware improves the speed [7]. Timescale option in hardware gives flexibility for concurrency.

The feed forward networks and recurrent networks do not have feedback connections. Perceptron and Adaline are single layer feed forward networks.[2] In multi-layer feedforward networks, there are different hidden layers. A node in the hidden layer takes the data coming from below layer and sends data directly to the above layer, and there are no connections within a particular layer.[2] The nodes must be trained using a particular learning rule. There are two learning rules to train single layer networks. They are perceptron rule and delta rule which are iterative procedures to adjust the weights of the linear networks. The functionalities implemented by single layer networks are very limited, so multi-layer networks are employed for various features.[2]

In 1969, Minsky implemented a feed-forward network, which has paved a way to a new era of designers to construct ANNs. In single layer, there are no hidden layers whereas multi-layer network has hidden layers. But the problem of adjustment in weights across the nodes from input to hidden nodes was not solved in [2].

The solution for adjustment of weights is done by backpropagating the errors. While distributing the weights across the nodes, there are some errors in the calculations of a function. When a learning rule is followed or a teaching pattern is applied, the errors from output layers must be distributed across the hidden nodes with the help of back propagation property. For feed-forward, linear activation function,
delta rule is applied. For multiple layers and for non-linear activation, generalized delta rule is applied. Generalized delta rule is commonly known as back propagation rule.[2]

Two rules are applied for multi-layer networks they are i) generalized delta rule and ii) chain rule. In feed forward network, hidden nodes are modelled with non-linear functions. While calculating and updating the weights, few errors usually occur. These errors are distributed across all the hidden nodes according to chain rule. The generalized delta rules have two stages, in the first stage, for every input propagated through the network, output is calculated, and the calculated output is compared with the desired value. Second stage includes a backward pass through the network, where the error signal is passed to each node in the network and appropriate weight changes are calculated then synaptic weights for both the layers are updated. Before applying back propagation process on the hidden nodes, weighted delta value from each output node and the activation function of the hidden unit must be considered [8]. Flow of back propagation algorithm is shown in Figure 2.

![Flow of Back Propagation Algorithm](image)

**Figure 2.** Flow of Back Propagation Algorithm
Flow chart of back propagation algorithm steps are,

Step 1. The neural network is initialized, and all the weights are initialized.

Step 2. Input activation functions are given.

Step 3. The weight for the hidden nodes is calculated

Step 4. The errors in desired to actual values of the outputs are calculated

Step 5. Weights of the hidden nodes are updated.

Step 6. Check for the error and go to step 3 until error becomes less than 0.00001.

Back propagation algorithm has low convergence rate and the system get struck in local minima. Various advanced algorithms based on same learning rule as back propagation have been developed. The local minima problem can be rectified to some extent in generalized back propagation algorithm and the convergence rate of generalized back propagation is improved as it uses ordinary differential equation.[9]

4. Hardware Implementation

Back propagation algorithm can be implemented in different hardware modules. In this paper, microcontroller, memristor, ASIC and FPGA implementations are studied and discussed. Portion of the algorithm can be implemented on software and remaining portion is on hardware. Software part of BP algorithm is done mostly in MATLAB or C or HDL code is written.

A. Microcontroller Implementation

In [24],[25] neural network implements BP algorithm for multi-layers on microcontroller boards. Off-line training of the network is done by tangent-sigmoid function for activating the neurons and real time implementation is done by tangent-sigmoid function by piecewise linear approximation. Microcontroller 89C52 can be used to realize perceptron for applications like a robotic car [10]. Extra circuitry is required to implement the neural hardware [11].

MATLAB has Neural network toolbox program, the firing function of neurons is done by piece wise approximation of tangent sigmoid function, so that it can be implemented on microcontroller. Neural network is selected on the trial and error basis. Hidden layer is activated by tansig-sigmoid function, and for the output layer a pure linear function is applied. Different frameworks like TensorFlow can be employed to design AI system for new modules, ideas, and interfaces.[12] Ternarized backpropagation algorithm is aimed particularly for processing of edge technology.

As shown in the Figure 3, the input sensors data has to be learned. There is a transfer function in simulation tool which converts the inputs on to the neural network which is based on microcontroller. Microcontroller board has the memory block. The neural network must be trained with BP learning algorithm and the outputs obtained are displayed through output ports of the microcontroller.

![Figure 3. Microcontroller based ANN](image-url)
B. Memristor Implementation

Memristor has been in research for the past decade. Its name implies memory resistor, a nanoscale device[13]. Its memory storage property makes it unique, as it has very simple structure, utilizes less power, more speed. Memristors are non-volatile, good scalable and high density [14],[15]. Memristors are like synapses in the brain of the human body. So, in various ANN applications, memristors can be implemented as synapses.[16] Memristor has been realized as laboratory device in 2008 by Stanley group [8],[17]. A single memristor can be used in the analog multiplication which is helpful in synaptic processing [16].

In 2014, memristor, made of metal oxide crossbars is experimented to implement a perceptron. A memristor is used to design a single layer neural network [18]. The memristor experiment was extended to three-layer perceptron, and the perceptron was simulated using back propagation algorithm [19],[5]. Two series memristor works as half bridge rectifier. As shown in Figure 4, according to the input voltage applied, memristances changes depending on the polarity. The memristances act as weights in neural hardware [8].

![Figure 4. Memristor bridge circuit](image)

To implement a memristor in neural network, learning rule is applied in MATLAB. A host computer updates the synapse weight on the memristor. Backpropagation algorithm is applied on memristor bridge synapse [16]. The weights are trained by back propagation algorithm. Here, backpropagation algorithm trains the weight in software. Memristances changed by suitable width of the pulses are used as coefficients of the weights. The limitation is the implementation of memristor crossbars array. As the hardware complexity increases, implementation of memristor crossbar arrays is difficult [8]. The number of memristors is reduced by half when memristor based synapse is implemented in comparison with traditional memristor bridge circuit.[8]

C. ASIC Implementation

ASIC design flow [20] in simple form is shown in the Figure 5. The register transfer level code of the VHDL code written and is synthesized [21], then a gate level netlist is generated. The hardware implementation in ASIC has additional steps than FPGA, such as floor planning, placement and routing, and physical implementation. These additional steps will take longer time when compared to FPGA.
There are various methods of hardware implementations on FPGAs. For prototyping applications FPGAs are the best option. The FPGA tools for simulation and synthesis, and various FPGA boards are easily available. The code written in Hardware Description Languages like Verilog, VHDL, System Verilog can be simulated and synthesized in tools like Xilinx and Cadence. After assigning all the pin connections for the selected FPGA board in the tool, by externally connecting the FPGA board, the implementation of the code written can be verified on the FPGA board. There are diverse FPGA generic libraries created for ANN algorithm. Xilinx has python based PYNQ Community where different artificial intelligence-based projects can be hosted on Zynq boards.

The Zynq 7000 series FPGA architecture consists of SoC, it integrates ARM based processor with hardware programmable FPGA. On a single device, it has combination of central processing unit, digital signal processor, ASSP. The 7 series architecture is complex than older families. Moreover, its performance is better than previous families. All Xilinx FPGAs consist of slices, memory, multipliers, programmable interconnects, clock buffers. The neurons in the artificial intelligence network are
concurrent. They require parallel processing while implementing on hardware. The core feature of FPGA being parallel processing, the implementation of neural hardware is done.[22]

Figure 6 shows the flow [23] from High Level Language like C, Java to FPGA. Many compilers are available online. SeaCucumber software is a synthesizing compiler. Circuits are generated by SeaCucumber, accepting Java class files as input [18]. Similarly, online compilers can be used to convert from C code to hardware module interface.[24] The digital design engineers codes in HDL. IEEE Standard 1364, is applied in Verilog HDL for describing various components and circuits [25]. MATLAB can be used to design algorithms and can be simulated with Simulink. With the help of HDL coder, VHDL or Verilog code can be generated in MATLAB.

FPGAs have programmable logic blocks array. As shown [4] in Figure 7, It has several blocks of I/O, memory, multipliers, general logic, with a routing fabric which can be programmed. The I/O blocks connects the FPGA chip to the external world. The FPGAs can be programmed even after the fabrication is completed. By writing a code in HDLs, FPGAs can be programmed for a particular function or task.

![Figure 7. Basic FPGA Structure](image)

FPGAs are re-programmable and with different FPGA architecture, machine language algorithms can be employed [26], [4]. As the re-programmable characteristic of FPGA is suitable for machine learning type algorithms, Parallel processing implementation has many applications in machine learning [27]

Field programmable gate arrays are preferable to application specific integrated circuit for various reasons. FPGA offer fast programming and testing by the end user. FPGAs are best for prototyping, and the ease of shifting to final design is what makes it unique. The re-programmable feature of FPGAs gives flexibility for the designer. The reusability of FPGAs for other design is possible, and also the design changes/modifications are way easier in comparison to ASICs.

The financial risk associated with the FPGAs is less when compared to ASICs, and turn-around time of ASICs are much longer. In terms of cost, FPGAs incur low costs in small volumes, that means it offers lower start-up costs. The design tools used for FPGA are less expensive in comparison to ASIC. The major disadvantages of FPGA compared to ASICs are slower, larger area, uses more transistors per logic, power consumption [25]. FPGA requires more area than ASIC, has speed performance slower than ASIC and consumes dynamic power more than ASIC [20],[28].
E. Performance of Back Propagation Algorithm on FPGA

There are diverse FPGA generic libraries created for ANN algorithms. While coding back propagation algorithm, these libraries will be useful. A 2x2xl neural network of two inputs, one output and two hidden nodes has been designed. The backpropagation learning algorithm is designed completely on FPGA and resource utilization is optimized [28]. To get the output neuron in particular range [-1,1], tansig is used as activation function for hidden layer. To limit output neuron in range [0,1], Sigmoid activation function is used [28]. To train the neural network, tansig and sigmoid functions has been applied iteratively and programmed on FPGA. [28]

Sigmoid function implementation on an FPGA effectively is the most tough task on part of the designers. In other cases, alternative methods of sigmoid are employed.

F. Comparative Analysis of various Hardware implementation

|                    | FPGA                                      | ASIC                                      | Microcontroller | Memristor                      |
|--------------------|-------------------------------------------|-------------------------------------------|-----------------|--------------------------------|
| Back Propagation   | Implemented using different features      | Implemented but time taking               | Implemented     | Difficult when hardware        |
| algorithm          |                                           |                                           | But not         | Complexity increases           |
|                    |                                           |                                           | customizable    |                                |
|                    |                                           |                                           | fully           |                                |
| Area               | 20 to 30 times more area than ASIC       | Moderate                                  | Less area       | Very less                      |
| Cost               | Less expensive                            | Expensive in small volumes               | Moderate        | Less                           |
| Time to produce    | One or two days                           | 18 months                                 | Readily available| Less                           |
| Hardware Complexity| Less as boards are available              | Less                                      | Moderate as     | More as size increases         |
|                    |                                           |                                           | boards are      |                                |
|                    |                                           |                                           | available       |                                |
| Programming        | Verilog, VHDL                             | Verilog, VHDL                             | C, C++          | MATLAB for Simulation          |
| Languages          |                                           |                                           |                 |                                |
| Power              | 10 times more dynamic power than ASIC    | Less power                               | Power Optimized | Low power                      |
|                    |                                           |                                           | than FPGA       |                                |

As shown in the above table 1, back propagation algorithm can be easily implemented easily on FPGA, microcontroller in comparison to ASIC and memristor.

FPGAs occupy less area but less expensive in comparison to ASIC. As memristor is a nanoscale device, back propagation can be implemented for less nodes but difficult when memristor crossbars are employed. As microcontroller boards are readily available, microcontroller boards offer less flexibility in comparison to FPGA boards when implementing back propagation algorithm. Production time is very less for FPGA and microcontroller, but FPGAs offer more flexibility in comparison to microcontroller. Hardware complexity is less for FPGA and ASIC. The coding languages used for FPGAs, ASIC are Verilog, VHDL and microcontroller uses C, C++. Memristor simulation can be done in LTSpice, MATLAB. Power consumption is less for ASIC, Microcontroller, memristor in comparison to FPGA.

5. Conclusions

Various back propagation algorithms on FPGA architecture, Microcontroller, ASIC, memristor is studied and analyzed. Memristor based neural networks occupy less area but as hardware complexity
increases, back propagation learning rule cannot be applied completely. Back propagation algorithm on FPGA has great scope of applications. ASIC takes longer shelf-time, therefore applications are limited in ANN. Microcontroller based applications are not that customizable and flexible in comparison to custom based. Back propagation algorithms combined with other algorithms in neural networks can increase the convergence rate of BP algorithm. Back propagation algorithms combined with other algorithms have greater scope of application in real time. So, back propagation algorithm combined with FPGA can give many real time solutions.

6. References

[1] Hunt K J, Sbarbaro D, Zbibkowski R and Gawthrop P J 1992 Neural networks for control systems-A survey Automatica 28 1083–112
[2] Montavon G 2020 Introduction to Neural Networks Lecture Notes in Physics 968 37–62
[3] Cheerla S, Ratnam D V, Vinay C, Avinash N, Anish I and Imran S B 2019 A semi empirical path loss model by using artificial neural networks (Ann) International Journal of Innovative Technology and Exploring Engineering 8 835–41
[4] Raman M, Abdallah N and Dunoyer J 2019 An Artificial Intelligence Approach to EDA Software Testing: Application to Net Delay Algorithms in FPGAs Proceedings - International Symposium on Quality Electronic Design, ISQED 2019-March 311–6
[5] Pandya V, Areibi S and Moussa M 2005 A handel-C implementation of the back-propagation algorithm on field programmable gate arrays Proceedings - ReConFig 2005: 2005 International Conference on Reconfigurable Computing and FPGAs 2005
[6] Bataineh A al, Kaur D and Jarrah A 2018 Enhancing the Parallelization of Backpropagation Neural Network Algorithm for Implementation on FPGA Platform Proceedings of the IEEE National Aerospace Electronics Conference, NAECON 2018-July 192–6
[7] Kansal S, Sikri M, Gupta A and Sharma M 2019 A prospect of achieving artificial neural networks through FPGA 2018 International Conference on Computing, Power and Communication Technologies, GUCON 2018 358–63
[8] Vo H M 2018 Training on-chip hardware with two series memristor based backpropagation algorithm 2018 IEEE 7th International Conference on Communications and Electronics, ICCE 2018 179–83
[9] Ng S C, Leung S H and Luk A With Convergence Analysis 0–3
[10] Farooq U, Hasan K M, Raza A, Amar M, Khan S and Javaid S 2010 A low cost microcontroller implementation of fuzzy logic based hurdle avoidance controller for a mobile robot Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010 9 480–5
[11] Rooban S, Kumar K S, Shankar K R and Rao N U B 2019 D6228048419 882–6
[12] Kaneko T, Momo H and Asai T 2019 An FPGA Accelerator for Embedded Microcontrollers Implementing a Ternarized Backpropagation Algorithm 2019 International Conference on Reconfigurable Computing and FPGAs, ReConFig 2019
[13] Rooban S, Swathi K L, Monica C and Shivaramakrishna B 2019 An odd parity genertor design using nano-electronics International Journal of Engineering and Advanced Technology 8 597–601
[14] Junsangsri P, Member S and Lombardi F 2013 <Design of a Hybrid Memory Cell Using Memristance.pdf> 12 71–80
[15] Zhang Y, Wang X and Friedman E G 2018 Memristor-Based Circuit Design for Multilayer Neural Networks IEEE Transactions on Circuits and Systems I: Regular Papers 65 677–86
[16] Kim H, Sah M P, Yang C, Roska T and Chua L O 2012 Memristor bridge synapses Proceedings of the IEEE 100 2061–70
[17] Strukov D B, Snider G S, Stewart D R and Williams R S 2008 The missing memristor found Nature 453 80–3
[18] Świątek J 2014 Preface Advances in Intelligent Systems and Computing 240 V–VI
[19] Zhu R, Lin P, Lv Y, Chang S, Wang H, Huang Q and He J 2018 Back-Propagation Neural Network based on Analog Memristive Synapse 2018 IEEE International Conference on Electron Devices and Solid State Circuits, EDSSC 2018 2–3
[20] Kuon I and Rose J 2007 Measuring the gap between FPGAs and ASICs IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 26 203–15
[21] Ranzini S M, Parahyba V E, Vilela T, Schneider E O, Tardelli J B, Oliveira J C R F, Lopes E P, Lima T C, Reis J D, Oliveira J R F and Fraidenraich G 2015 Digital back-propagation ASIC design for high-speed coherent optical system SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference Proceedings 2015-Decem
[22] Anon Xilinx - Adaptable. Intelligent.
[23] Anon c-to-verilog.com - c to verilog Resources and Information.
[24] Lynch M L and Singer S M 1997 Next generation diagnostic ATPG system using the Verilog HDL Proceedings - IEEE International Verilog HDL Conference 1708 56–63
[25] Kuon I, Tessier R and Rose J 2007 FPGA architecture: Survey and challenges Foundations and Trends in Electronic Design Automation 2 135–253
[26] Krishna B M, Kumar M L, Tanmayee M, Krishna C A and Charan P R S 2019 FPGA based coin recognition system International Journal of Innovative Technology and Exploring Engineering 8 1589–94
[27] Anitha A, Rooban S and Sujatha M 2019 Implementation of energy efficient gates using adiabatic logic for low power applications International Journal of Recent Technology and Engineering 8 3327–32
[28] Vo H M 2017 Implementing the on-chip backpropagation learning algorithm on FPGA architecture Proceedings - 2017 International Conference on System Science and Engineering, ICSSE 2017 538–41