Implementation of Back Propagation Algorithm in Verilog

Neelmani Chhedaiya
neelmanichhedaiya@gmail.com
SSGI Bhilai (C.G)
Prof. Vishal Moyal
vishalmoyal@gmail.com

ABSTRACT

In this paper, a design method of neural networks based on Verilog HDL hardware description language, implementation is proposed. A design of a general neuron for topologies using back propagation algorithm is described. The sigmoid nonlinear activation function is also used. The neuron is then used in the design and implementation of a neural network using Xilinx Spartan-3e FPGA. The simulation results obtained with Xilinx ISE 9.2i software.

The backpropagation algorithm is one of the most useful algorithms of ANN training. In this paper, we present the neuron implementation for the in topologies that are suitable for this algorithm. The neuron is then used in a multilayer neural network. For the implementation, Verilog HDL language is used. Verilog HDL is a hardware description language which simplifies the development of complex systems because it is possible to model and simulate a digital system form a high level of abstraction and with important facilities for modular design.

The purpose of this work is to suggest and analyze several neuron implementations, show a way for the integration and control of the neurons within a neural network, and describe a way to implement a simple feedforward neural network trained by BP algorithm using Xilinx 9.2i.

Key words: Artificial Neural Network, Backpropagation, Verilog HDL.

INTRODUCTION

A Neural Network is a powerful data-modeling tool that is able to capture and represent complex input/output relationships. The motivation for the development of neural network technology stemmed from the desire to develop an artificial system that could perform "intelligent" tasks similar to those performed by the human brain. Neural networks resemble the human brain in the following two ways:

- A neural network acquires knowledge through learning.
- A neural network's knowledge is stored within inter-neuron connection strengths known as synaptic weights.

Following are the advantage of neural network:
1. An ability to learn how to do tasks based on the data given for training or initial experience.
2. An ANN can create its own organization or representation of the information it receives during learning time.
3. ANN computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.
4. Partial destruction of a network leads to the corresponding degradation of performance. However, some network capabilities may be retained even with major network damage.

Learning in neural network-

Learning is a process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place. All learning methods used for neural networks can be classified into two major categories:

Supervised learning which incorporates an external teacher, so that each output unit is told what its desired response to input signals ought to be. During the learning process global information may be required. Paradigms of
supervised learning include error-correction
learning (back propagation algorithm),
reinforcement learning and stochastic learning.
**Unsupervised learning** uses no external teacher
and is based upon only local information. It is
also referred to as self-organization, in the sense
that it self-organizes data presented to the
network and detects their emergent collective
properties. Paradigms of unsupervised learning
are Hebbian learning and competitive learning.

**Neuron Implementation**

The common mathematical model of a neuron is
shown in Fig. (1)

![Mathematical model of artificial neuron.](image)

The neuron output can be written as:

$$a = f \left( \sum_{j=1}^{R} w_j p_j \right)$$

where $w_j$ are inputs, $p_j$ are weight coefficients, $f$
activation function, and $a$ neuron output.

**BACKPROPAGATION NETWORK**

Back propagation is a form of supervised
learning for multi-layer nets, also known as the
generalized delta rule. Error data at the output
layer is back propagated to earlier ones, allowing
incoming weights to these layers to be updated.
It is most often used as training algorithm in
current neural network applications. The back
propagation algorithm was developed by Paul
Werbos in 1974 and rediscovered independently
by Rumelhart and Parker. Since its rediscovery,
the back propagation algorithm has been widely
used as a learning algorithm in feed forward
multilayer neural networks.

**VERILOG HDL**

In the semiconductor and electronic design
industry, Verilog is a hardware description
language (HDL) used to model electronic
systems. Verilog HDL, not to be confused with
Verilog HDL (a competing language), is most
commonly used in the design, verification, and
implementation of digital logic chips at the
register-transfer level of abstraction. It is also
used in the verification of analog and mixed-
signal circuits.

Verilog HDL is a programming language that
has been designed and optimized for describing
the behavior of digital systems. Verilog HDL has
many features appropriate for describing the
behavior of electronic components ranging from
simple logic gates to complete microprocessors
and custom chips. Features of Verilog HDL
allow electrical aspects of circuit behavior (such
as rise and fall times of signals, delays through gates, and functional operation) to be precisely described. The resulting Verilog HDL simulation models can then be used as building blocks in larger circuits for the purpose of simulation. Verilog HDL is also a general-purpose programming language: just as high-level programming languages allow complex design concepts to be expressed as computer programs, Verilog HDL allows the behavior of complex electronic circuits to be captured into a design system for automatic circuit synthesis or for system simulation.

METHODOLOGY

The proposed neural network has two layers with four neurons each and sixteen synapses per layer. Different situations may need neural networks of different scales. Such a situation can be overcome by combining a certain number of such unit neural networks. Back propagation algorithm logic has been divided into individual modules and these modules have been implemented in Verilog HDL using behavioral modeling.

The different modules are: synapse, neuron, error generator at the input, error generator at the output, weight update unit and a weight transfer unit. At the synapse inputs are multiplied by the corresponding weights and the weighted inputs are summed together to get four outputs. After the synapse is the neuron that calculates the output in accordance with the transfer function sigmoid and its derivative has also been calculated. Further the neuron is followed by the synapse and the neuron because in this paper two layer network has been considered. Then the network is followed by the error generator at the output, which compares the output of the neuron with the target signal for which the network has to be trained. Similarly, there is error generator at the input, which updates the weights of the first layer taking into account the error propagated back from the output layer.

Finally, a weight transfer unit is present just to pass on the values of the updated weights to the actual weights. Then a final entity having structural modeling has been formulated in which all the entities are port mapped. The results constitute simulation of Verilog HDL codes of different modules in MODELSIM. The simulation of the structural model shows that the neural network is learning and the output of the second layer is approaching the target.

CONCLUSION

The proposed neural network has two layers with four neurons each and sixteen synapses per layer. Different situations may need neural networks of different scales. Such a situation can be overcome by combining a certain number of such unit neural networks.

Back propagation algorithm logic has been divided into individual modules and these modules have been implemented in Verilog HDL by using any one modeling. This paper describes the Verilog HDL implementation of a supervised learning algorithm for artificial neural networks. The algorithm is the Error Back propagation learning algorithm for a layered feed forward network and this algorithm has many successful applications for training multilayer neural networks. The simulation of the any one model by using Verilog HDL shows that the implementation creates a flexible, fast method and high degree of parallelism for implementing the algorithm.

FUTURE SCOPE

An algorithm similar to back propagation algorithm, which may be used to train networks whose neurons may have discontinuous or non-differentiable activation functions. These new algorithms should also have the capability to speed up the convergence of back propagation algorithm. Modified forms of back propagation algorithm such as Quick BP, RPROP, SAR-PROP, and MGF-PROP can provide a great help.
REFERENCES

[1]. Hardware Implementation of Backpropagation Neural Networks on FPGA- Rafid Ahmed Khalil-University of Mosul, College of Engineering, Mosul, Iraq


[3]. FPGA realization of backpropagation for stock market prediction by Wan-De Weng; Rui-Chang Lin; Nat. Yunlin Univ. of Sci. & Technol., Nantou in the year 2007


[5]. Neural Network Implementation Using FPGA: Issues and Application by A. Muthuramalingam, S. Himavathi, E. Srinivasan