Abstract

Background/Objectives: Recent research focuses on low-power design techniques. This has been mainly inspired by the demand of hand-held electronic devices which have to consume less power. Neural network based classifiers are widely used in speech recognition which needs higher power. Methods/Statistical Analysis: A hybrid approach for designing architecture of Multi-Layer Perceptron (MLP) based Neural Network (NN) for speech recognition is proposed using bipartite tabular method and banking organization method. This approach is prototyped in Xilinx xc3s1200. The Back propagation neural network is trained in MATLAB using TIDIGITS corpus. Using the optimized weights from MATLAB, the proposed prototyped low-power architecture is evaluated in Xilinx. Findings: The outcome shows a considerable reduction both in area and power. Conclusion/Improvements: The reduction in switching power is by 33% and the average power is by 25% are noted. There was 2% reduction in the resources used by the proposed architecture.

Keywords: Banking Organization Method, Bipartite Tabular Method, Low Power Architecture, Multi-Layer Perceptron Neural Network, Speech Recognition

1. Introduction

Artificial Neural Network (ANN) has been widely used in pattern recognition \(^1\), signal processing, classification and control systems etc. Most of the research work in this field till now consists of software simulation, investigating the capabilities of ANN model or developing a new algorithm. However, hardware implementation is also essential for applicability and for taking the advantage of Neural Network's (NN) inherent parallelism \(^2\). To realize any required function in NN, hardware implementation is important. An ANN system consists of a number of artificial neurons and a huge number of interconnections among them. The network is formed by connecting a number of these neurons using weighted connections. ANN is then trained to perform a useful function by adjusting its weight using a learning algorithm. Widely used neural classifier is Multi-Layer Perceptron (MLP) \(^3\).

In modern VLSI technology, low power consumption design techniques have become more and more essential. This has been mainly motivated by the demand of portable communication and computation equipment which must consumes as little power as possible to extend the battery life. Furthermore, increasing use of ANN in embedded devices motivate the development of specialized NN hardware. As a result, there has been considerable research interest in the hardware implementation of ANN. In recent years, FPGA technology has made very significant advancement enabling the implementation of highly complex systems. In this paper, we use Verilog as the Hardware Description Languages (HDL) to implement ANN architecture in FPGA.

In speech recognition, pre-processing of speech is done to extract the useful features from speech which is called as Feature extraction method. Feature extraction of speech is done using several different techniques, the most common being Linear Predictive Coding (LPC), Perceptual Linear predictive (PLP) and Mel frequency Cepstral Coefficients (MFCC). LPC is a time-domain technique which varies in the amplitude of the speech signal due to noise. The preferred technique for feature extraction is MFCC with 13 coefficients \(^4\,\,\,5\) where the

*Author for correspondence
features are generated by transforming the signal into frequency domain. In general, Cepstral features are more compact and de-correlated. The experiment based on TIDIGITS corpus demonstrates the effectiveness of proposed techniques leading to higher speech recognition accuracy. The training of Back propagation NN is done in MATLAB using TIDIGITS. In this paper, fixed-point representation is adopted to represent and optimized NN weights and the feature vectors.

In this research, the architecture of serial MLP is implemented on FPGA, in which the architecture generally follows different modules. RAMs are used to store weight and speech vector, activation and functional units. In this paper these modules were realized with different low power methodology which is explained in section-2. This paper consists of the following sections: A brief introduction about MLP NN and digital hardware implementation of serial MLP are discussed in section 2. Section-3 presents the proposed design architecture of Serial MLP NN and briefs about the bipartite tabular and banking organization method. In section-4 comparisons were made between the previous result of serial MLP hardware architecture and the proposed architecture in terms of area and power.

2. Overview and Implementation of MLP

2.1 Multi-Layer Neural Network

The most common NN model is the MLP Neural Network. It is a kind of supervised network which requires a desired output in order to learn. The purpose of this network is to create a model that appropriately maps the input to the output using historical data so that the model can be used to produce the output when the desired output is unknown. The MLP NN model consist a network of processing element or nodes arranged in layers. Typically, it requires three or more layers of processing node, an input layer which accepts the input features used in classification procedure, one or more hidden layer, and output layer as shown in Figure 1. In the block diagram of MLP which is shown in Figure 1 in which the data from input pattern are entered first at the input layer and network node performs calculations in the successive layer until an output value is computed at each of the output node. The output signal should indicate the appropriate data of the input data. For example only one output node should give the high output value, while all other nodes give the low value.

Every processing node in one particular layer is usually connected to every node in the layer above and below it. The weighted connections define the behaviour of the network and are adjusted during training through a supervised training algorithm. In a feed forward network each input pattern vector is presented to input layer. For successive activation, the input to each term is the summa-tion by their respective weight which is given in equation

\[ \text{sum}_i = \sum_{j=0}^{n} w_{ij} \text{out}_j \] (1)

where \( w_{ij} \) is the weight connecting from node \( j \) to node \( i \) and \( \text{out}_j \) is the output from node \( j \). The output of a node \( i \) is \( \text{out}_i = f(\text{sum}_i) \), which is then sent to all node in the next layer. This process is continues through all the layer of the network until the output layer is reached and the output vector is computed. The function \( f \) denotes the activation function of each node. A sigmoid activation function is frequently used in equation (2):

\[ f(\text{sum}_i) = \frac{1}{1 + e^{-\text{sum}_i}} \] (2)

2.2 Hardware Implementation

Figure 2 shows the generalized block diagram of MLP proposed in this work. Implementation of MLP-based system in hardware is shown in Figure 3, which has 13 inputs of nodes, 15 nodes of hidden layer and 9 output nodes.
The function of the computational neuron can be implemented by means of main blocks such as RAM to store speech and synapses weights, Multiplier to multiply speech vector with synaptic weight, Adder and accumulator for summation of multiple neuron, multiplexer is used to select RAM during computation either from input to hidden or from hidden to output, 13-bit counter used to select particular data (weights, feature vector) from specific address. As shown in Figure 3 in Serial MLP design, the maximum power been utilize or dissipate mainly during switching, reading and writing data into RAM and module such functional unit and activation unit dissipate power during switching because this module enable continuously. So, To design an area and power efficient architecture, two relevant methods are chosen, one is banking organization and another is bipartite tabular method. Proposed architecture discuss in next section.

3. Proposed Serial Multi-Layer Perceptron Neural Network

As shown in Figure 4, the Low power serial MLP architecture is with two new approaches. In Figure 4(a) this architecture synapses weights are stored in different set of RAM module to reduce power using banking organization. Bipartite tabular method is used in activation unit, with this technique two tables (Positive half and negative half table shown in Table 1) are accessed in parallel and the outputs of the two tables provide an approximation to the function in carry save form. These two tables must be combined with a carry propagate adder two’s complement approximation to the function. When compared to any other technique, this tabular method requires less memory and these methods are explained in detail in section 4.

As shown in Figure 4(b) we adopted one more technique, instead of using simple multiplier we use booth multiplier to reduce the number of partial product because in the main block speed of multiplication is

### Table 1. PWL approximation for the 15 segment bipolar sigmoid function

| Input Range | Output Range          |
|-------------|-----------------------|
| -8 -7 64     | -1 → -0.9921875       |
| -7 -6 65     | -0.9921875 → -0.984375 |
| -6 -5 66     | -0.984375 → -0.96875  |
| -5 -4 67     | -0.96875 → -0.9375    |
| -4 -3 68     | -0.9375 → -0.875      |
| -3 -2 69     | -0.875 → -0.75        |
| -2 -1 70     | -0.75 → -0.5          |
| -1 0 71      | -0.5 → 0              |
| 0 1 72       | 0 → 0.5               |
| 1 2 73       | 0.5 → 0.75            |
| 2 3 74       | 0.75 → 0.875          |
| 3 4 75       | 0.875 → 0.9375        |
| 4 5 76       | 0.9375 → 0.96875      |
| 5 6 77       | 0.96875 → 0.984375    |
| 6 7 78       | 0.984375 → 0.9921875  |
| 7 8 79       | 0.9921875 → 1         |
depends on addition of partial products. More the number of bits, multiplier/multiplicand is composed of more are the number of partial products, longer is the delay in calculating the product. The critical path of the multiplier depends upon the number of partial products. For example, if two “n” bits number are multiplied, there will be ‘n’ partial products to add. Booth’s multiplication is an answer in reducing the number of partial products. Using Booth’s multiplier, the number of partial products is reduced to ‘n/2’, hence reduction in area.

3.1 Banking Organization
Banking is an organization technique that target total switching capacitance to achieve reduced power and to improve the speed. The values of optimized weight vector extracted from MATLAB are exported to RAM module with different address locations. An unfortunate outcome of such methods is that any access of address causes the entire RAM to be enabled and the address which is not toggled will dissipate power. To reduce unnecessary switching, we approach a banking method to store synaptic weights in different set of RAM’s. For example consider the total number of weights which are require to multiply with speech vector as per equation (1), 1440 such weights are needed and to store and access this, it requires 1440 addresses. When RAM is enabled only one address will be selected and the rest of addresses will be in active mode but they will not be selected and thus they will dissipate power. Thus storing 360 addresses in each set of bank and select one of them simultaneously dissipate less power.

Clearly splitting the memory into a set of smaller memories is an efficient way to reduce the total switched capacitance as shown in Figure 4. An additional set of decoder is required to select the one of the bank and store the optimized weights into a set of RAMs, While an additional delay is incurred in selecting the bank, it is offset by the much lower capacitance that switched per bank, which is clearly the total power must be divided by four (average power/4). This technique is mainly preferred for large RAMs, as is reduce the overall power consumption.

3.2 Bipartite Table Method
Elementary functional approximation is important in several areas, including digital signal processing, image processing and scientific computing etc. For the application that requires low-precision elementary function, approximation is implemented by using look-up table whereas if we use application that uses high-precision, the look-up table needs more memory.

In the Hardware implementation, there are number of approaches to implement the sigmoid function. One of the method recently been developed for approximating elementary function is based on bipartite tabular method. As shown in Table 1 exponential term varies in a given output range for a particular input. The Table 1 forms symmetry for positive as well as negative values.

As shown with the input range of –8 to 8 and outside of these limits we approximate the outputs to the values 1 or –1 which is shown in Table 1. We call this solution as PWL sigmoid mode. The solution is symmetric if we use a sign-magnitude representation; therefore we can implement only half of the Table 1 by using the structure as shown in Figure 6.

As shown in Figure 5 the first block checks MSB sign bit, if the MSB sign bit of input is “1”, it shows a negative value and it converts the magnitude of MSB sign bit. If the MSB sign bit of input is “0”, it shows a positive value and magnitude conversion doesn’t take place. According to MSB bit the multiplexer bypasses the input value from half-positive table (Table 1). Output from table is always positive so if MSB sign bit of input is “1”, it converts the output available from table into two’s-complement form. This method proves that instead of using look-up table to

---

**Figure 5.** Banking organizations for storing synaptic weights.

**Figure 6.** Sigmoid function block diagram.
store the values in RAM this conventional method occupy less area. The disadvantage of this method is accuracy decreases up to certain level.

4. Experimental Result

The implementation of Multi–Layer Perceptron with bipartite method and banking organization are implemented using Verilog and prototyped using XC3S1200 FPGA. The simulation results at the output node for TIDIGIT corpus for digit-One and for digit-2 are shown in Figure 7(a) and (b).

In Figure 7(a), the purple line indicates the nine output nodes of the output layer in which data_out-1 is at logic-1 and rest of outputs are equal to logic-“0” corresponding to TI digit “ONE” and similarly Figure 7(b) for TI-digit “TWO”. The results at output nodes are also compared with MATLAB and Xilinx to find accuracy as shown in Figure 8.

As shown in Figure 8 the variations between MATLAB and Xilinx output value at output layer and average power is calculated using X-Power estimation tool and the area table.

5. Conclusion

Neural network architectures are widely used in pattern recognition specifically speech recognition. In this paper, Low-power Serial MLP Neural Network architecture is proposed for isolated digit recognition. The low power techniques used are banking organization method, bipartite tabular method and booth multiplier. Back Propagation Neural Network is trained with TIDIGITS

![Figure 8](image1.png)

**Figure 8.** Compared values at output node after passing through activation unit in MATLAB and Xilinx.

| Table 2. Resource utilization of the XC3S1200 FPGA |
|-----------------------------------------------|
| Resources           | Resources used in [6] | Resources used in Proposed design | Maximum Available | Percentage Utilization |
|---------------------|------------------------|----------------------------------|-------------------|------------------------|
| Occupied slices     | 1530                   | 1497                             | 8672              | 17%                    |
| 4-Input LUTs        | 2178                   | 2141                             | 17344             | 12%                    |
| Number of IOBs      | 198                    | 157                              | 304               | 51%                    |
| Number of BUFMuxs   | 0                      | 1                                | 24                | 4%                     |
database in MATLAB and the optimised weights of the NN are exported to FPGA and evaluated the serial MLP NN in FPGA. With the improved design, the result shows the required computation power and the area are drastically reduced by 25%.

### 6. References

1. Zebardast B, Maleki I, Maroufi A. A novel multilayer perceptron artificial neural network based recognition for kurdish manuscript. Indian Journal of Science and Technology, 2014 Mar; 7(3):343–51.

2. Keles F, Yildirim T. Low voltage power neuron circuit design based on subthreshold FGMOS transistor and XOR implemetation. SM2ACD, IEEE; 2010 Oct 4–6. p. 1–5.

3. Canas A, Ortigosa EM, Ros E, Ortigosa PM. FPGA implementation of a fully and partially connected MLP. Springer; 2006. p. 271–96.

4. Manikandan J, Venkataramani B, Avanthi V. FPGA implementation of support vector machine based isolated digit recognition system. New Delhi: IEEE; 2009 Jan 5–9. p. 347–52.

5. Abdalla MI, Ali HS. Wavelet-based mel-frequency cepstral coefficients for speaker identification using hidden markov models. Journal of Telecommunication, 2010 March; 1(2):16–21.

6. Canas A, Ortigosa EM, Ros E, Ortigosa PM. Hardware description of multi-layer perceptrons with different abstraction level. Microprocessor and Microsystem. Elsevier; 2006 Nov; 30(7):435–44.

7. Haykin S. Neural networks: a comprehensive foundation. Englewood Cliffs, NJ: Prentice-Hall, New York; 1998.

8. Son YH, Seongil O, Ro Y, Lee JW, Ahn J. Reducing memory access latency with asymmetric DRAM bank organization. ACM (ISCA’13); Tel-Aviv, Israel. 2013 Jun; 41(3):380–91.

9. Gadea-Girones R. FPGA implementation of non-linear predictors. Springer; 2006. p. 297–323.

| Design          | Switching power (nw) | Net power (nw) | Leakage power (nw) | Average power (nw) | Clock (MHZ) |
|-----------------|----------------------|---------------|-------------------|-------------------|-------------|
| [6]             | 60586.3              | 32607         | 183.31            | 31125.53          | 250         |
| Proposed design | 40592.21             | 28520.31      | 139.83            | 23084.031         | 250         |