Investigation on efficacy of neural network in signal processing

JOTHI.M,
ME VLSI DESIGN
Srinivasan Engineering College
[1]joemano.jothi6@gmail.com

DR.B.KARTHIKEYAN
Professor
Srinivasan Engineering College
bcarthik@gmail.com

***ABSTRACT*:** With the advent of new technologies and advancement in medical science we are trying to process the information artificially as our biological systems perform inside our body. Artificial intelligences through a biological word are realized based on mathematical equations and artificial neuron. Our main focus is on the implementation of Neural Network Architecture (NNA) with on a chip learning in analog VLSI for generic signal processing application. In the proposed paper analog components like Gilbert Cell Multiplier (GCM), Neuron activation Function (NAF) are used to implement artificial Neural Network Architecture. The analog components used are comprises of multiplier and adders, along with the tan-sigmoid function circuit using MOS transistor in sub threshold region. Design and verification of the proposed design is carried out using microwind simulation tool.

Keyword: Neural Network Architecture, Back Propagation Algorithm.

1. INTRODUCTION

1.1 Artificial Intelligence

Intelligence is the computational part of the ability to achieve goal in the world. Intelligence is a biological word and is acquired from past experience. The science which defines intelligence mathematically is known as Artificial Intelligence. Artificial Intelligence (AI) is implemented by using artificial neurons and these neurons comprises of several analog component. The proposed paper is a step in the implementation of neural network architecture using back propagation algorithm for data compressions. The neuron selected is comprises of multiplier and adder along with the tan-sigmoid functions. The training algorithm used is performed in analog domain thus the whole neural architecture is a analog structure.

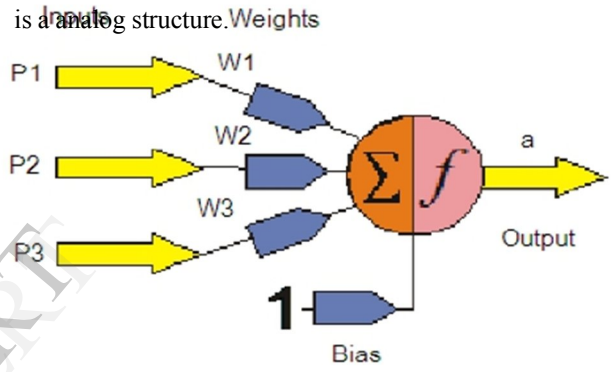

Figure 1: Neural Network

Figure1: can be expressed mathematically as

a = f (P1W1+P2W2+P3W3+Bias)

Where „a is the output of the neuron & „p is input and „w is neuron weight . The bias is optional and user defined. A neuron in a network is itself a simple processing unit which has an associated weight for each input to strengthening it and produce an output. The working of neurons is to add together all the inputs and calculating an output to be passed on. This neural architecture is trained using back propagation algorithm and also it is a feed forward network. The designed neurons are suitable for both digital and analog applications.

The proposed neural architectures is capable of performing operations like sine wave learning amplification & frequency multiplication and can also be used for analog signal processing activities.

1.2 Multiple Layers of Neurons

When a set of single layer neuron are connected with each other it forms a multiple layer neuron, as shown in the figure 2.

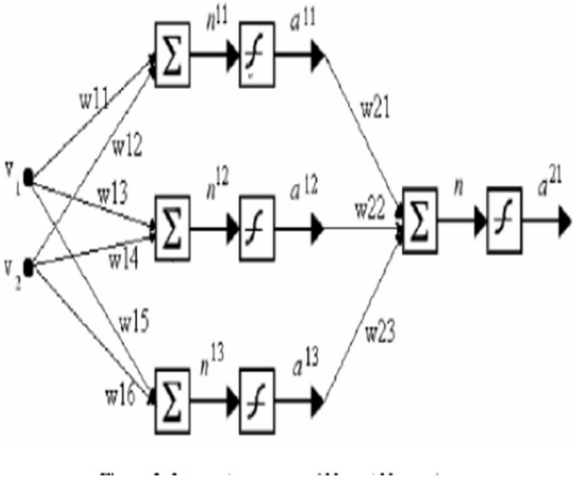

Figure 2: Layer structure of Neural Network

As it is clear from the above figure that weights w11 to w16 are used to connect the inputs v1 and v2 to the hidden layer. So weights w21 to w23 transferred the output of hidden layer to the output layer and the final output is a21.

2. ARCHITECTURE

2.1 Analog Components for Neural Architecture

The inputs to the neuron v1 and v2 as shown in figure 2 are multiplied by the weight matrix& the resultant output is summed up and passed through a Neuron Activation Function (NAF). The output of the activation function is then passes to the next layer for further processing. Blocks used are Multiplier block, Adders, Neuron Activation Function (NAF) block with derivative.

2.1.1 Multiplier Block

The Gilbert cell is used as the multiplier. The schematic of the Gilbert cell multiplier is as shown in the figure 3.

2.1.2 Adders

The output of the Gilbert cell is in the form of current (Transconductance).The node of the Gilbert cell connecting the respective outputs act as adder itself.

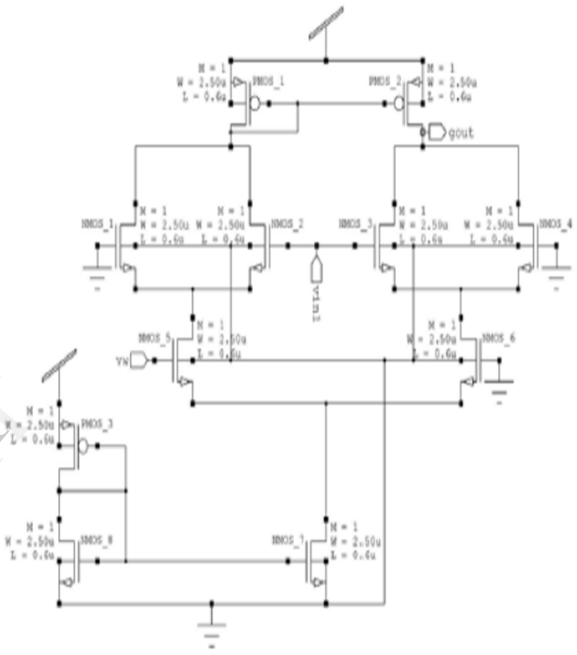

Figure 3: Gilbert cell multiplier with current mirror circuit

2.1.3 Neuron Activation Function (NAF)

The designed neuron activation function is tan sigmoid. The proposed design is really a differential amplifier modified for differential outputs. Same circuit is capable of producing output the neuron activation function and the differentiation of the neuron activation function. Here two designs are regarded for NAF

1. Differential amplifier as NAF

2. Modified differential amplifier as NAF with differentiation output.

2.1.3.1 Differential Amplifier Design As A Neuron Activation

Function (Tan)

This block is named as tan in the final schematics for Neural network Architectures. Differential amplifier when design to work in the sub-threshold region acts as a neuron activation function.

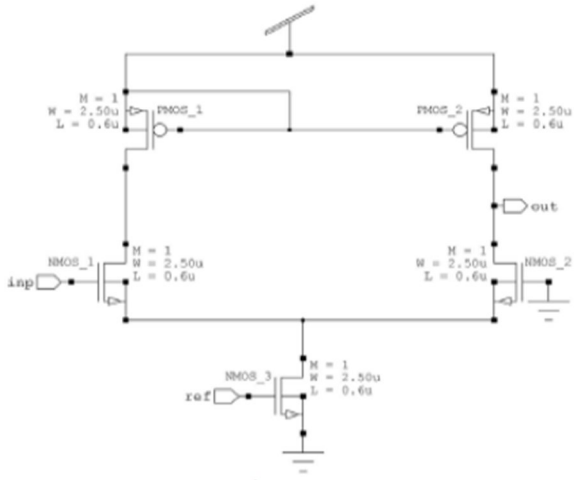

Figure 4: Schematic of NAF

2.1.3.2 Modified Differential Amplifier Design for Differentiation Output (fun)

Schematic of the design shown in the figure 5 is used for the tan sigmoid function generator with modification for the differential outputs. The NAF function can be derived from the same differential pair conformation. The structure has to be modified for the differential outputs.

2.2 Realization of Neural Architecture using Analog Components

The components designed in the previous section are used to implement the neural network architectures. The tan block is the differential amplifier designed in section 2.1.3.1. And this block is used as the neuron activation function as well as for the multiplication purpose. The multiplier is the Gilbert cell multiplier designed in section 2.1.1.The fun is the Neuron activation function circuit with differential output designed in section 2.1.3.2.

Figure 5 shows exactly how the neural network architecture of figure 2 is implemented using analog components. The input layer is the input of 2:3:1 neuron. The hidden layer is joined to the input layer by weights in the first layer called as w1i.

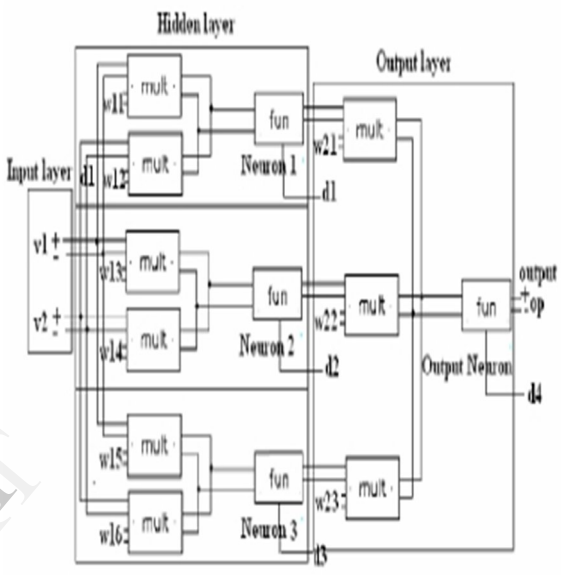

Figure 5: Implementation of the Neural Architecture using Analog Blocks

The output layer is joined to input layer through weights w2j. The op is the output to the 2:3:1 neuron. The schematic circuit for image compression is shown in figure 6.

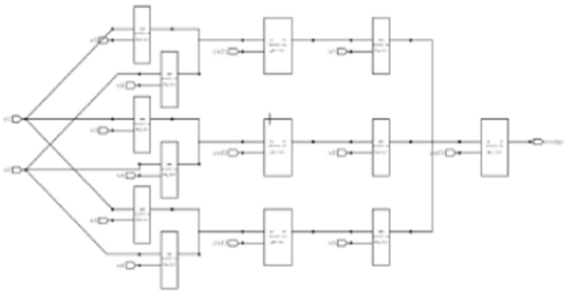

Fig 6: Schematic of Compression Block

2.3 Neuron Application -Image Compression and Decompression

The above proposed NA is used for image compression. Image comprising of pixel intensities are fed to the network shown in Figure 6 for compression and then this compressed image act as input for the decompression block. The 2:3:1 neuron proposed has an inherent capability of compressing the inputs; there are two inputs and one output. The compression achieved is 97.30%. Since the inputs are followed in the analog form to the network there is no need for analog to digital converters.

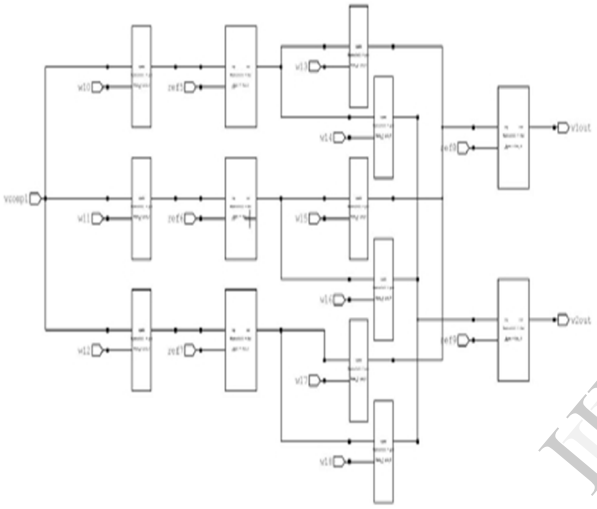

Figure 7 shows the schematic of decompression block

This is one of the major advantages of this work. A 1:3:2 neural networks are designed for the decompression intention. The neural network has 3 neurons in the hidden layer and two in the output layer.

Figure 8 shows the compression and decompression outline. The training algorithm used in this network is Back Propagation algorithm. The error distributes from decompression block to the compression block.

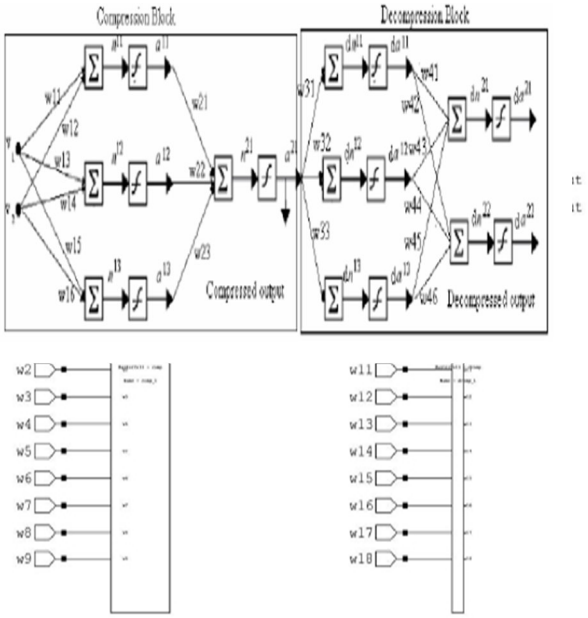

Figure 8: Image Compression and Decompression using proposed neural architecture

Formerly the network is trained for different inputs the two architectures are classified and can be used as compression block and decompression block independentlyFigure 9: Combined Schematic of Compression and Decompression Block

3. BACK PROPAGATION ALGORITHM

Back propagation is the almost common method of training of artificial neural network so as to minimize the objective function. It is the induction of delta rule and mainly used for feed forward networks. The back propagation algorithm is understand by dividing it into two stages. First stage is propagation and second is weight update.

1. Propagation

a) Forward propagation of training input patterns through with the neural network in order to generate the propagations output activations.

b) Backward propagation of the propagations output activations through with the neural network using the training patterns target in order to give the deltas of all output and hidden in neurons.

2. Weight Update

a) For each weight junction it multiplies its output delta

and input activation to get the gradient of the weight.

b) Add the weight in opposite direction of the gradient by subtracting a ratio of it from the weight.

This ratio determines the speed and quality of learning rate. The sign of the gradient of a weight shows where the error is increasing that is why the weight must be updated in the opposite direction. The phases goes on repeating until the performance is of the network is satisfactory.

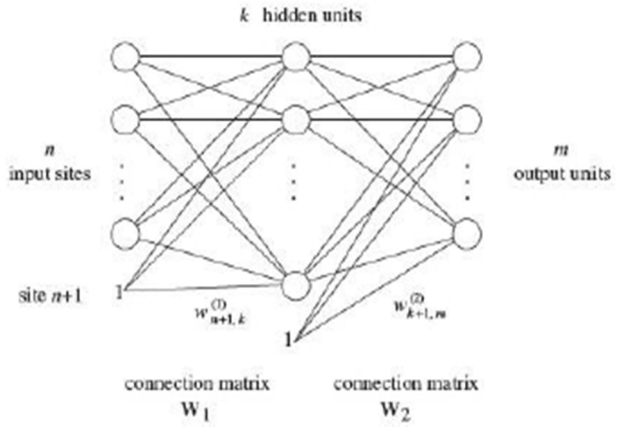

Figure 10: Notation for three-layered network

There are (n + 1) × k weights between input sites and hidden units and (k +1) ×m between hidden and output units. Let W1 denote the (n+1) ×k matrix with component $w_{ij(1)}$ at the i-th row and the j-th column. Similarly let W2 denote the (k + 1) × m matrix with components $w_{ij(2)}$ ij . We use an over lined to emphasize that the last row of both matrices represents to the biases of the calculating units+1 The matrix of weights without this last row will be needed in the back propagation step. The n-dimensional input vector = (o1, . . . , on) is extensive, transforming it to ˆo = (o1, . . . , on, 1). The excitation netj of the j-th hidden unit is given by:

$$o_j^{(1)} = s\left(\sum_{i=1}^{n+1} w_{ij}^{(1)}\,\hat{o}_i\right)$$

The activation function is a sigmoid and the output $O_{j(1)}$ of this unit is thus

After choosing the weights of the network randomly the back propagation algorithm is used to compute the essential corrections. The algorithm can be decayed in the following four steps:

- Feed-forward computation.
- Back propagation to the output layer.
- Back propagation to the hidden layer.
- Weight updates.

The algorithm is blocked when the value of the error function has become sufficiently small.

4. EXPERIMENTAL RESULTS

In this section, we present experimental results shows that the proposed back propagation algorithm it improves the compression and decompression efficiency. Analog implementation of the neural network reduce the circuit complexity and also avoids the of real time analog samples to digital signal. The back propagation algorithm works in much the same way as the name suggests: After propagating an input through the network, the error is calculated and the error is propagated back through the network while the weights are adjusted in order to make the error smaller. When I explain this algorithm, I will only explain it for fully connected ANNs, but the theory is the same for sparse connected ANNs. Although we want to minimize the mean square error for all the training data, the most efficient way of doing this with the back propagation algorithm, is to train on data sequentially one input at a time, instead of training on the combined data. However, this means that the order the data is given in is of importance, but it also provides a very efficient way of avoiding getting stuck in a local minima.

A neural network has to be configured such that the application of a set of inputs produces (either 'direct' or via a relaxation process) the desired set of outputs. Various

methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a priori knowledge. Another way is to 'train' the neural network by feeding it teaching patterns and letting it change its weights according to supervised and unsupervised learning rule. By using this algorithm I expect the better compression ratio compare to previous work.

5. CONCLUSION

Neural network with their remarkable ability to derive meaning from complicated or imprecise data can be used to extract patterns and to detect trends that are too complex to be noticed by either humans or other computer technics. Due to its adaptive learning, self-organizations, real time operations and fault tolerance via redundant information coding properties it can be used in Modeling and Diagnosing the Cardiovascular System and in Electronic noses which has several potential application in telemedicine’s. Some other application developed was “Instant Physician” which represents the best diagnosis and treatment. This work can be further led to implement neuro fuzzy system with high speed low power, CMOS circuit in current mode as well as for nano scale circuit simulation using Double Gate MOSFET (DG MOSFET) modeling.

REFERENCES

[1] Arne Heittmann, “An Analog VLSI Pulsed Neural Network for Image Segmentation using Adaptive Connection Weights” Dresden University of Technology, Department of Electrical Engineering and Information Technology, Dresden, Germany, 2000

[2] Ben Hommounda et al .H., “Neural Based Models Of Semiconductor Devices for Spice Simulator, “American J. of Applied Science, vol. 5, 2008, pp. 385-391

[3] Cyril Prasanna Raj P & S.L. Pinnae “design and analog vlsi implementation of neural network architecture for signal processing” European Journal of Scientific Research ISSN 1450-216X Vol.27No.2 (2009), pp.199-216

[4].F. Djeffal et al., “Design and Simulation of Nano electronic DG MOSFET Current Source using Artificial Neural Networks,” Materials Science and Engineering C, vol. 27, 2007, pp. 1111-1116

[5]R.A. Jacobs, Increased rates of convergence through learning rate adaptation , Neural Networks , 1 ,295-307, (1988)

[6]D. Nguyen a and B. Wid row, Improving the learning speed of 2-layer neural network by choosing initial values of the adaptive weights, IEEE First International Joint Conference on Neural Networks ,3, 21-26, (1990).

[7]Rafid Ahmed Khalil &Sa'ad Ahmed Al-Kazzaz “Digital Hardware Implementation of Artificial Neurons Models Using FPGA”pp.12-24

[8]Wai-Chi Fang et al, “A VLSI Neural Processor for Image Data Compression using Self-Organisation Networks” IEEE Transactions on Neural Networks, Vol. 3, No. 3, May 1992, pp. 506-517

[9]Ranjeet Ranade & Sanjay Bhandari & A.N. Chandorkar “VLSI Implementation of Artificial Neural Digital Multiplier and Adder” pp.318-319

[10]Bose N.K., Liang P.,Neural Network Fundamentals With graphs, algorithms and Application, TataMcGraw hill, New Delhi, 2002,ISBN0-07-463529-8.

[11] T. P. Vogl, J. K. Man gis, J.K. Rigler, W. T. Z ink and D .L. Alkon, Accelerating the convergence ofthe back-propagation method , Biological Cyberne tics, 59