Artificial Neural Networks as building blocks of Mixed signal FPGA

Manjunath.R, Dr.K.S.Gurumurthy,
Dept. EC & CSE UVCE, Bangalore
Email: manju_r_99@yahoo.com

Abstract

Ever since the deployment of FPAAs, efforts are on the way to minimize the silicon area to realize an arbitrary system. A relatively new concept which has been tested and tried [1] in this direction is the use of Artificial neural networks (ANNs) as Configurable Analog Blocks (CABs). Conventional ANNs however suffer with lengthy training period. In this paper ANNs with differential feedback technique are explored. It has been found out that they perform better than the conventional ANNs.

1. Introduction

The fact that human brain has astonishing capabilities of learning and memorizing complex functions has led to investigate a similar system on silicon, though not with those complexities. As neurons are the building blocks of Brain, similar Artificial Neural Network can be made as the building blocks of such a system. A good introduction for ANNs in the context of information theory may be found in [2-7]

There are various ways of implementing a function- By storing the lookup table for all possible input output pairs (Not for analog input), by using the actual hardware implementation (analog or digital) or by adaptively learning the function itself. They are examined here

In section 2.techniques for porting a neuron on to silicon is explored. Section.3 examines the differentially fed ANNs. In Section 4 Differentially fed ANN cells are explored along with the driving software requirements.

Sec 5 gives a brief discussion of simulation results and sec 6 concludes along with future work.

2. Neural hardware on silicon

The theoretical advantage of a hardware neural network is that the computations can take place in parallel, asynchronously, and limited only by the propagation delays in circuit. This opens up a large and

diverse area of application for uses such as signal processing, vision systems and system control. An "ideal" artificial neuron [9] is usually represented by the mathematical equation

$$Y = S\left(\sum_{i=1}^{n} x_i w_i\right) ----(1)$$

The neuron is made up of two functional blocks. a) Synapse: - It performs multiplication of an input signal and b) neuron body:-It sums the outputs of the attached synapses mathematically and passes the result through transfer function and on to other neurons in the network. The developed neuron circuit shown in Figure 3. In terms of operation, a single synapse is implemented by a pair of transistors, M1and M2. The current through these is nearly proportional to Vin * Vw. To handle signed weights there is a current mirror included in the neuron body to reverse the effective input from the synapses connected to it. The sign is decided by a pair of pass transistors

2.1.Implementation of Activation function.

For the non-linear function of the neuron, it is possible to use the fact that any logarithmic amplifier can get a non-linear response and produce a function similar to a sigmoid

$$\frac{dS(i_i)}{di_i} = \frac{-25mv * R_{fs}}{R_{fs} i_i + S(i_i) + 25mv}$$

$S(i_i)=-25mv*Ln\{i_iR_{fi}+S(i_i)\}+K_{i0}$

$K_{I0}=LnI_0(R_{is}+R_{fs})$ --(2)

It can be seen in [2] that at small values of i_i slope of the function is proportional to $-R_{fs}$. For large values of i_i the slope tends to zero, resembling a sigmoid function. Equation. (2) describes a family of sigmoid functions whose characteristics depend on the input current and the value of the external components. In another architecture ANNs are ported to FPAAs [1]. The hardware is reconfigurable. Here ANNs are made to learn some classification function and transferred to FPAA rather than making use of OPAMPS directly. Also, OPAMP may be used as accumulator for the partial weighted sums of neuron

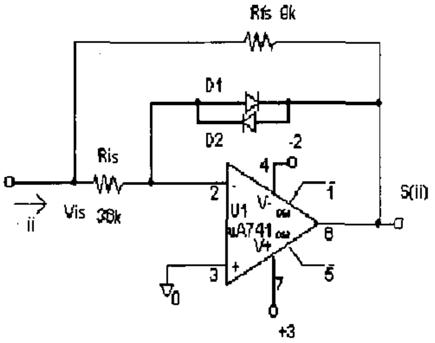

Figure 1.Transfer function Circuit

3. Formalism of differentially fed ANN

ANN works on the concept of function learning. A single hidden layer in general, is enough to catch any function. One of the major drawbacks of the conventional training methodology is that it is iterative in nature and takes a large number of cycles to converge to the pre specified error limit. By intuition, if more information is made to be hidden with the data, it takes less number of iterations to get stabilized to the pre defined error limit. Auto Regressive Moving Average (ARMA) model can be conveniently used in this direction

In a typical ARMA model, the output is the weighted sum of present and previous inputs and previous outputs, which may also be viewed as differential terms. The output y of a neural network but for the nonlinearities is the weighted sum of its *inputs. Again the output may be viewed as a particular* case of ARMA to which differential terms are added. The proposed neural network makes use of the higher order derivatives of its output as the additional inputs. This makes the architecture auto regressive and enhances its predictability. The output y of a neural network but for the nonlinearities can be written as

$y=\Sigma w_i x_i$.-----(3)

Where x_i are the inputs w_i, the corresponding weights. Summation is taken over all inputs. Again the linearity of the output of equation (3) may be viewed as a particular case of ARMA

$y(n+1)=b_0y(n)+b_1y(n-1)+.....+a_0x_n+...$-----(4)

Where b_0.. and a_0.. are constants. The auto regressive terms $b_0...b_n$ may be realized using an implied differential feedback [10]. With differential feedback it has been found out [10] that the number of iterations required for training is reduced as shown in the table I. XOR gate is considered for simulation. Gaussian distributed random input with seed value 1000 is taken as input. With I order different feedback, the output may be written as:

$\Sigma w_i x_i + b_1 y_1$------(5)

y_1 being the I order differential. Table I shows that with increase in the order of differential feedback, the training period reduces. As the order approaches infinity the number of iterations can be brought down .As the order approaches infinity the number of iterations can be brought down substantially. This increases the no. of inputs to be handled. In [10] it has been shown that these terms can be replaced by a single term corresponding to the highest order of the differential considered. The results are given in Table II for II order differential where the two feedback differential terms are replaced by a single II ordered differential term and weight

$W_{eq}=(w_1*(\text{I ordered feedback input})+w_2*(\text{II ordered feedback input}))/y_0$---(6)

4. Neural CAB

A primitive configurable neural block called UNE (unit to compute neurons)[11] is shown in figure.6.

Section III indicates improved performance with differential feedback applied. Any CAB in general should have memory to store weights. Multipliers to multiply with these weights, accumulator to add the partial sums and a non linear transfer function which are all configurable.

Any software driving such a CLB should have a few desirable features. The neural CAB concept is attractive when it's programmed in a real system. For this driving software suit is needed along with a development board for evaluation and rapid hardware design. A rich set of IP (Intellectual property) library need to be provided. Also, the synthesis step of design flow need to learn different functions and replace them by neural hardware (CABs of FPAA or neural CABs if they are available)

4.1. Operation of the neural CABs.

The synapses signals feed the activation function circuit with the accumulated sum of the weighted voltage inputs. The activation function circuit draws a current which is proportional to the sum of products of inputs and weights as its input. The weights for the inputs are represented by resistors. It is possible to adaptively change the weights through switched capacitors.

The cells may be restricted to have limited (say 2) inputs. When the function contains more than 2 inputs, 2 cells can be used with their results cascaded

4.2. Results and discussion

It is well established that, in a control system, proper feedback can improve the stability. Going by a similar analogy, a proportional differential feedback has been tried in an artificial neural network. Two important properties have been observed in a differentially fed artificial neural network-Reduction in the training period and smaller Square error, making them attractive for the online usage. The error in the ANN output is found to be minimal when there exists a differential feedback from the output to the input, which serves as additional input. Also, by increasing the order of the differential, the error can be made arbitrarily small

Conventional ANN suffers with the lengthy training period while the linear models like ARMA fail to catch up the non-linear functions. The features of both these models are combined in the new architecture to overcome their drawbacks. The numbers of iterations required to train the neural network have been reduced.

with differentially fed Artificial Neural Network. The drawback of using OPAMP instead of ANN is that it fails to catch nonlinearities and noisy inputs. Though it can implement functions, it cannot learn them making the required hardware more. For e.g. The XOR function takes one neural cell (CAB or CLB) for its

realization .Direct realization takes more than one OPAMP (and CAB) for realization. Even digital logic can be realized using Neural CABs and used in FPGA cells as ANN itself may be taught of as a lookup table

5. Conclusions and future work

In this paper, neural networks with differential feedback are introduced. The superiority of these differentially fed neural networks over conventional ones in function learning is demonstrated. When such a neural network is used as building block of a programmable analog array, a large number of parameters have to be numerically investigated before finalizing the clear cut advantages. They include Hard ware cost, Noise performance, speed, power loss, Range of inputs for ANN, Component reuse/wear & tear of some CLBs. The performance of conventional FPAAs, Conventional neural networks on FPAAs, neural networks on dedicated CABs, differentially fed neural networks on FPAAs and on dedicated CLBs have to be tested for these parameters making the problem fivefold

The hardware required appreciably changes when the building blocks make use of dedicated CABs consisting neurons and surrounding logic implemented using CMOS and when existing CABs of FPAAs consisting of OPAMP are used. This is because the cell architecture along with the peripherals will be different for with and without Feedback types. Naturally the hard ware used to learn a function, the hardware wasted in the reconfigurable blocks, and the training time/data will be different. The growth of complexity as the number. of inputs increases has to be examined As it is, differential feedback requires more hard ware because of extra feedback inputs used .Sec II shows that these terms can be replaced by a single equivalent term during training. Its impact on range ,SNR and training time has to be examined .Also the number of cells required to perform a specific task will be different. The range restriction on the inputs will be eased on differential feedback method due to their inherent Auto regressive nature. To have same range ordinary ANN requires different number of cells

With different architectures, the degree of Fault tolerance will vary. Overhead involved in adaptive reconfigurability when a fault is detected during training or during commissioning has to be worked out. Also, fault diagnosis complexities will be different. Additionally, the possibility of blending Bio neurons on to silicon in the CABs may be thought

Table 1.Performance with feedback

Order of differential	Square error	Iterations
No feedback	18	1156
I order	18	578
II order	18	289

Table 2.Performance with II order feedback

Order of differential	Square error	Iterations
II order Feedback	18	578
Equivalent Output feedback	18	578

References

[1]. Aguilera Galicia C. R. Realización electrónica de redes neuronales artificiales aplicadas a sensores virtuales de biomasa en circuitos integrados analógicos onfigurables en campo" Master on Industrial Electronics Thesis, ITESO, Guadalajara, 2001.

[2.] Aarts, E.H.L. and Korst, J.H.M.Simulated Annealing and Boltzmann Machines, Chichester:Wiley.1989

[3].S.Amari Information Geometry of the EM and em algorithms for neural networks, Neural networks,8,No.9,1995

[4].Amari.s. Differential geometry of curved exponential families –curvatures and information loss.Annals of statistics.1982

[5].S.Amari, Natural gradient works efficiently in learning, neural computation 9,1998 :251-276

[6].S.Amari and H.Nagaoka, Methods of information geometry, AMS and Oxford University press, 2000.

[7].Information geometry of EM and em algorithms for neural networks, Neural networks 8(9), 1995, 1379-1408

[8].Zetex Group company, Fast Analog solutions LTD® designing and manufacturing the TRAC family U.K, http://www.fas.co.uk/, (vi: 03/1999).

[9].L. Fausett, *Fundamentals of Neural Networks*, Prentice-Hall, 1994.

[10].Manjunath.R and K.S.Gurumurthy, System design using differentially fed Artificial Neural networks, TENCON 2002

[11]. ErezEtzion., NN in Physics, Numerical Recipies,04,02.2002