VLSI IMPLEMENTATION OF GENERAL PURPOSED CONIC SECTION FUNCTION NEURAL NETWORK

Burcu Erkmen Tulay Yildirim
e-mail: bkapan@yildiz.edu.tr *e-mail: tulay@yildiz.edu.tr*
Yildiz Technical University, Department of Electrical & Electronics Engineering, 34349, Besiktas, Istanbul, Turkey

Key words: Conic Section Function Neural Networks, Mixed mode design, Current mode analog circuits

ABSTRACT

In this paper, a circuit system of General Purposed Conic Section Function Neural Network is presented. The feed-forward analog computational cells have been designed by using the current mode approach. The network is trained in a chip-in-the-loop fashion with a host computer implementing the training algorithm. The network inputs and the feed-forward signal processing are analog. The mixed analog-digital design consists of 16 inputs, 16 hidden layer neuron and 8 outputs. 8 bit precision is selected to store weight, center and angle values on the EEPROM digital memory cells. The implemented feed-forward network circuitry has been tested on a classification problem successfully.

I. INTRODUCTION

Hardware realization of neural networks with their generalization capability is useful for numerous pattern recognition and signal processing applications. Neural network model require a lot of computing time to be simulated on a sequential machine resulting a great difficulty to investigate the behavior large neural networks and to verify their ability to solve problems. The neural system solves complicated problems by parallel operation of neurons. Performed in hardware, the operations performed by these circuits will take place in parallel, and in real-time [1]. As such, they will allow the neural network to converge at a higher speed than software-based counterparts.

In literature, several architectures have been introduced for realization of artificial neural networks. Among the architectures, MLP and RBF probably are the two most widely used neural networks for practical applications. Due to complementary properties of these networks several attempts have been performed to bring MLPs and RBFs under unified framework to make simultaneous use of advantages of both networks. In [2], a hybrid Radial Basis Function-Multilayer Perceptron (RBF-MLP) network was used to improve performance. The Conic Section Function Neural Network (CSFNN) [3] is a unified framework for MLP and RBF networks to make simultaneous use of advantages of both networks.
Fully analog implementations of CSFNN neurons [4] and fully digital implementations of CSFNN [5] exist in literature. In this paper, we propose mixed mode CSFNN architecture with combining the advantages of both analogue and digital realizations. The current mode analogue hardware is used in the forward pass calculation while digital memories are used to store network parameters. Chip-in-the-loop learning technique has been used to train CSFNN circuit.

This paper organized as follows. In Section II the theory of Conic Sections Neural Networks is overviewed. General information of CSFNN circuitry is given in Section III. The circuit design flow is briefly described in Section IV. The feed-forward analog computation of CSFNN is examined in Section V. In Section VI the simulation results of the CSFNN neuron are showed. The classification results of hardware realization for a benchmark problem are presented in Section VII. Finally conclusions are given in Section VIII.

II. CONIC SECTION FUNCTION NEURAL NETWORKS

The conic section function neural network (CSFNN), first described by Dorffner [3], is capable of making automatic decisions depending on the distribution of a given data. Decision boundaries hyperplane and hypersphere are the special cases of CSFNN. These are the decision boundaries of MLP and RBF, respectively. There would be intermediate types of decision boundaries such as ellipses, hyperbolas or parabolas in between those two cases which are also all valid for decision regions. Mathematically, the conic sections are formed from the intersection between a cone and a plane.
The neural computation is different in hidden neurons and output neurons in CSFNN. Hidden neurons realize the propagation rule of CSFNN and sigmoid activation function. The output neurons are inner product type. The propagation rule of hidden neurons can be derived using analytical equations for a cone. The principle of a CSFNN is that the cone of each hidden unit can be adapted so as to

make an automatic decision on the most appropriate region boundary. The following equations are obtained for n-dimensional input space for CSFNN neuron.

$$u_j^p(x) = \sum_{i=1}^{n} (x_i^p - c_{ij}) w_{ij} - \cos \omega_j \sqrt{\sum_{i=1}^{n} (x_{pi} - c_{ij})^2} \qquad (1)$$

$$f_j^p(x) = \frac{2}{1 + e^{-2 \cdot u_j^p}} - 1 \qquad (2)$$

Where x_{pi} refers to input vector for p. pattern, w_{ij} refers to the weights for each connection between the input and hidden layer, c_{ij} refers to center coordinates and ω_j refers to opening angles. i and j are the indices referring to the units in the input and hidden layer, respectively. This equation consists of two major parts analogous to the MLP and the RBF. The equation simply turns into the propagation rule of an MLP network, which is the dot product (weighted sum) when ω is π/2. Second part of the equation gives the Euclidean distance between the inputs and the centers for an RBF network. Figure 1 illustrates the structure of a CSFNN.

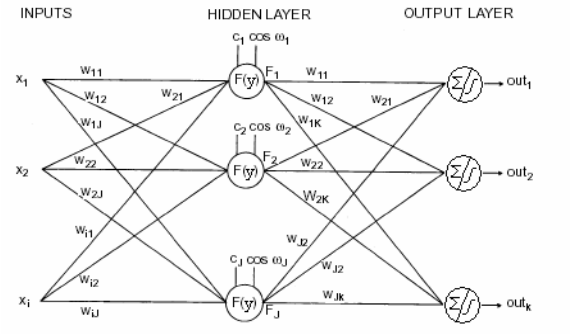

Figure 1. Conic Section Function Neural Network structure

III. GENERAL INFORMATION ABOUT CSFNN CIRCUITRY

The implementation of CSFNN hardware differs in several aspects. In this section, the advantages and disadvantages of each aspect has been examined briefly.

MIXED ANALOG/DIGITAL HARDWARE

In the fully analog implementation of ANN, analog storages for weight values are capacitors, but these can store an analog voltage only for a short time because of charge leakage. The neural signal processing is fully analog, yielding high speed operation, low power consumption and compact circuitry. The main drawbacks of analog systems include sensitivity to ambient noise and to temperature. In the fully digital implementation of ANN, digital memory is good for long term weight storage, but on the other hand, the digital synapse and neuron circuits account for bigger size in silicon area. If high precision is not required, one can take full advantage of using an analog approach to build a neural architecture. By combining the advantages of both analog and digital realization of the ANN, mixed hardware design is a meaningful way to the implementation of ANN. In this paper, feed forward neural signal processing is fully analog; control unit and storage of the synaptic weights are fully digital.

CURRENT MODE DESIGN

Current mode signal processing offers several advantages when used in neural circuits. One of the most apparent advantages is that the summing of many signals is most readily accomplished when these signals are current. Arithmetic operations, such as addition, subtraction and scaling, are typically difficult to implement and it is often area- and power-consuming in a voltage-mode system. Other advantage is increased frequency of operation due to use of low-impedance internal nodes, and increased dynamic range of signals allowed when MOS transistors can be operated over a wide range of signals allowed when MOS transistors can be operated over a wide range, from weak inversion to strong inversion [6]. In this work hardware realization of feed-forward computation is composed of current mode analog circuits.

WEIGHT CENTER AND ANGLE STORAGE

Learning in analog VLSI systems is inherently coupled with the problem of storage of analog information since after learning it is most often desirable to retain the learned weights for an extended period of time. Ideally, weights would be held in a long-term, easily-modified store. In practice, permanence of a stored weight (e.g. on a floating gate) must be traded against the ease of its modification [7]. In this study the weight, center and angle values are stored on on-chip floating gate devices.

WEIGHT PRECISION

Before designing circuits to implement neural computations, it is necessary to determine the precision requirements. Although precision requirements are somewhat problem dependent, in general, five-bit weight registers are sufficient for forward computations, but 12 bit resolutions is necessary for learning [8]. For this work, after doing simulations for several bit resolutions for weight, center and angle values, 8 bit (plus 1 sign bit) precision is chosen to store these values on the digital memory for acceptable network training.

LEARNING MODES

In literature, there are different approaches to train Neural Chips [9].

Off-chip Learning

Here the chip plays no part in the training process. A computer simulation is used to find the solution weight set and these weights are downloaded onto chip. Clearly, the problems associated with this technique arise from the fact that the network implemented on the chip is not identical to that simulated on the computer.

On-chip Learning

In this method, the training takes place entirely on the chip. Only the training data is supplied to the chip, and on-chip weight adaptation hardware modifies the synaptic weights according to the implemented learning rule. Improved performance (training speed) comes at the cost of less flexibility.

Chip-in-the-Loop Learning

This technique relies on a host computer to train the chip. This computer supplies the training vectors, and the chip is used 'in-the-loop' to generate the neural network outputs. The computer reads the outputs and then calculates the weight updates. The updated weights are downloaded to the chip before the next training iteration. Once training is complete, the host computer is no longer required. Since the analog hardware is used in the forward pass calculation, circuit anomalies can be compensated for during the learning process. In this study, chip-in-the-loop learning technique is used to overcome typical analogue process variations. The block diagram of chip in the loop learning technique used for training our hardware is shown in Figure 2.

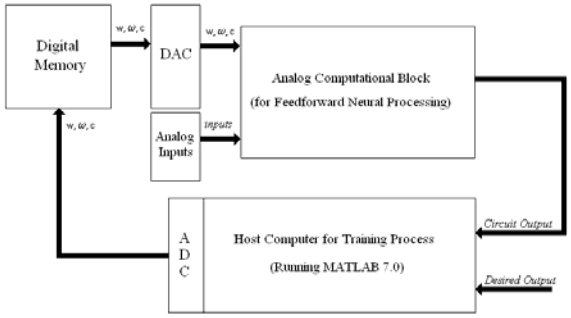

Figure 2. The Block Diagram of Chip-in-the-loop Learning Technique

IV. THE CIRCUIT DESIGN FLOW

The overall circuit is divided into two main parts with regard to their operating modes, i.e. analog and digital. The block diagram of implemented CSFNN circuitry is illustrated as Figure 3. The neural signal processing is fully analog and the synaptic weights, centers and angles are stored as digital form. It is a fully interconnected feedforward structure with 16 current mode analog inputs, 16 hidden layer neurons and 8 outputs. The weight, center and angle values are stored on on-chip floating gate devices. Current mode analog circuits are used for feed-forward neural processing. The digital weights, centers and angels is converted analog signals through the use of D/A circuits. Negative weights is converted the analog signals using the sign bit circuit integrated into presented D/A converter circuit. The general purpose neural-network circuit system is problem independent. Reconfigurability is obtained with switching circuits which has the ability to alter the topology of the neural network. Switching circuit is controlled by digital control unit. Reconfiguration switches inserted in the interconnection between DAC outputs and network inputs.

Digital block contains digital memories, digital control unit and decoders needed to address and to access memories. The weight, center and angle values are stored on on-chip digital floating gate memory cells with 8 bit precision. These digital values are selected, red and written by the row, column and block decoders, read and write circuitry. 704 Byte floating gate memory cells are designed to store these values.

The feed-forward computation is realized in analog block which detail examined in Section V.

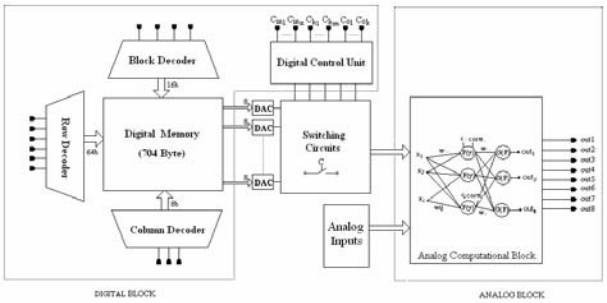

Figure 3. The Block Diagram of Implemented CSFNN Circuitry

V. ANALOG FEED_FORWARD COMPUTATION

In this work, the feed-forward neural computation is fully analog. Analog circuitry is composed of current mode circuits to easily realize the arithmetic operations, such as addition, subtraction. The neural computation is different in hidden neurons and output neurons. Hidden neurons realize the propagation rule of CSFNN (Eq 1). The output neurons are inner product type, and have sigmoid-like activation function. The whole network composed of computation units such as multiplication circuits, square root circuits, squarer circuits, sigmoid generator circuits. A functional diagram of CSFNN neuron circuitry for hidden units is shown in Figure 4. The output neurons are composed of only multiplication circuits, sigmoid generator circuits.

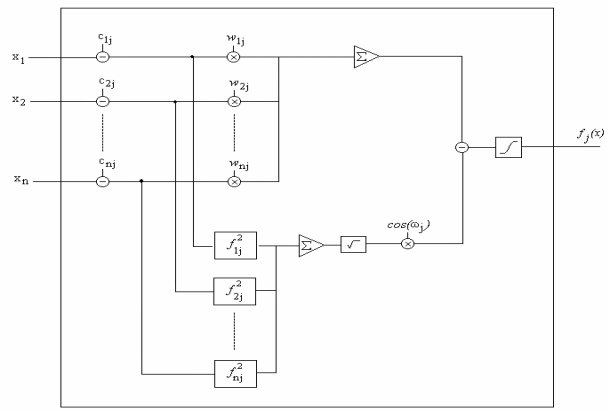

Figure 4. Functional diagram of CSFNN neuron circuitry

VI. THE SIMULATION RESULTS OF THE CSFNN NEURON CIRCUITRY

Cadence software tool has been used to simulate the implemented the circuits performed with Spectre in Analog Artist environment. The simulations have been done to the neuron circuitry to obtain decision boundaries using Cadence with AMIS 0.5µm CMOS transistor model parameters with 5V voltage supply. Different type decision boundaries have been obtained using only one CSFNN neuron circuitry. Open and closed decision boundaries and intermediate types of decision boundaries in between those two cases are realized. The decision boundaries of the straight lines for MLP, the circles for RBF and the transition from the straight line to the circles are shown in Figure 5, Figure 6, Figure7, respectively.

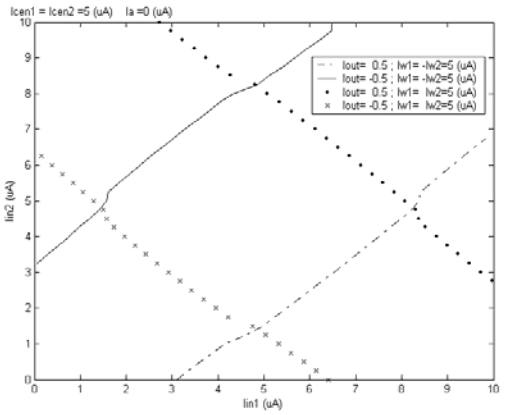

Figure 5. Open decision boundaries (MLP)

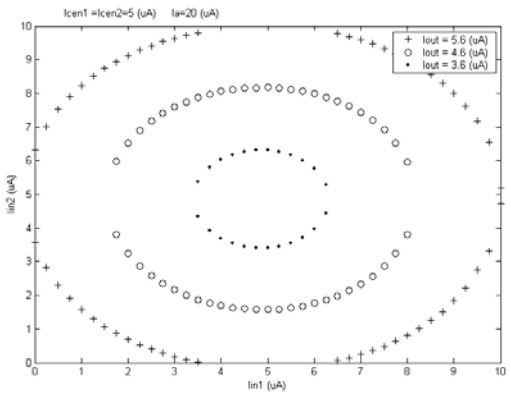

Figure 6. Closed decision boundaries (RBF)

VII. PATERN CLASSIFICATION APPLICATION

The implemented circuit of CSFNN networks has been applied to a highly nonlinear pattern recognition benchmark to ensure performance of designed circuitry. The small version of an IRIS plant classification problem constituted pattern recognition benchmark. The data set contains 3 classes where each class refers to a type of IRIS plant. Each pattern was described with 4 structural attributes of the plant. The training set contains 12 patterns, 4 from each pattern class. The test set contains 75 patterns, 25 from each pattern class.

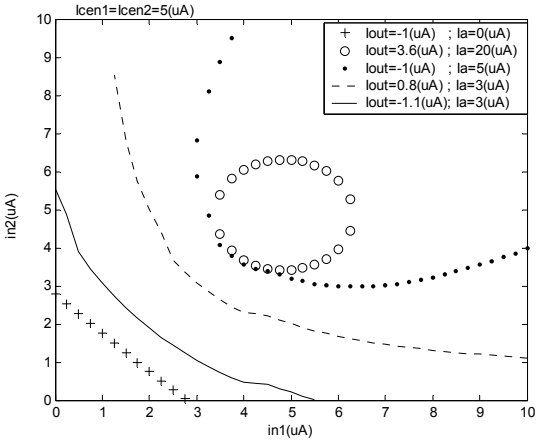

Figure 7. Transition decision boundaries from MLP to RBF

A CSFNN with 3 hidden neuron (i.e. 4-3-2) was used to solve the problem. The circuitry was trained chip-in-the-loop technique. The back-propagation algorithm was implemented on a host computer during the training. Training has been achieved within 60 epochs software-only environment, then 15 epochs with chip-in-the-loop technique. After doing 75 iterations, success rate of 100% has been obtained with training set. The test set presented to the implemented circuitry without using the host computer and achieved 94.67% success rate.

Training and testing process has also been performed at the software-only environment (MATLAB 7.0) using CSFNN network with 75 iterations. The same performance was obtained with chip-in-the-loop technique for the training set and 96% success rate was achieved with test set. This comparison demonstrates that the implemented circuitry and applied chip-in-the-loop learning technique have been successfully used for classification problems on condition with 16-16-8 maximum network size. The classification rates for software-only environment and chip-in-the-loop learning technique are summarized at Table I.

Table I. Simulation results for the classification problem

Chip-in-the-Loop Learning		Software-only Environment	
Train (%)	Test (%)	Train (%)	Test (%)
100	94.67	100	96

VIII. CONCLUSION

In this work, a circuit system of General Purposed Conic Section Function Neural Network is designed. The implemented circuit system is problem independent. The chip architecture allows reconfigurable topologies. The

whole system comprises analog and digital blocks. The neural signal processing is fully analog and the synaptic weights, centers and angles are stored as digital form. Feed-forward computation realized with current mode analog subcircuits. This implementation computes the Radial Basis Function (RBF) and Multilayer Perceptron (MLP) propagation rules with unified framework on a single hardware. Open and closed decision boundaries and intermediate types of decision boundaries in between those two cases are realized with only one CSFNN neuron circuitry to show the functionality of CSFNN neuron. The implemented circuitry has been applied to highly nonlinear pattern recognition benchmark to ensure performance of designed circuitry. Training process realized chip-in-the-loop learning technique. Successful classification performance is obtained for this problem. In further work, CSFNN network will be applied an image recognition problem.

ACKNOWLEDGMENT

This research has been supported by TUBITAK- The Scientific and Technological Research Council of Turkey. Project Number : 104E133**.**

REFERENCES

1. B.J. Sheu, J. Choi, Neural Information Processing and VLSI, Kluwer Academic Publishers, USA, ISBN: 0792395476, 1995
2. N. Chaiyaratana, A. M. S. Zalzala, Evolving Hybrid RBF-MLP Networks Using Combined Genetic/Unsupervised/Supervised Learning, UKACC Int. Conf. on Control'98, Swansea, UK, IEE Publication 455, vol. 1, pp. 330-335, 1998
3. G. Dorffner, Unified frameworks for MLP and RBFNs: Introducing Conic Section Function Networks, Cybernetics and Systems, Vol.25. pp. 511-554, 1994
4. T. Yıldırım, J.S.Marsland, An RBF/MLP Hybrid Neural Network Implemented In VLSI Hardware, Conf. Proc. of NEURAP'95 Neural Networks and Their Applications, Marseilles, France, pp. 156-160, 1996.
5. H. Esmaelzadeh, H. Farshbaf, C. Lucas, S.M. Fakhraie, Digital Implementation For Conic Section Function Networks, Microelectronics, 2004. ICM 2004 Proceedings. The 16th International Conference on, pp. 564 – 567, 2004
6. S. M Fakhraie, K.C. Smith, VLSI Compatible Implementations for Artificial Neural Networks. Kluwer Academic Publisher, pp:14-20, 1996
7. G. Cauwenberghs, M.A. Bayoumi, Learning on Silicon Adaptive VLSI Neural Systems, Kluwer Academic Publishers, USA, ISBN:0792385551, 1999.
8. A.J. Montalvo, R.S. Gyurcsik, J.J Paulos, Toward a General-Purpose Analog VLSI Neural Network With On-Chip Learning, Neural Networks, IEEE Transactions on, Vol 8, 2, pp. 413 – 423, 1997
9. G. Cairns, L. Tarassenko, Learning with Analogue VLSI MLPs, Microelectronics for Neural Networks and Fuzzy Systems, 1994., Proceedings of the Fourth International Conference on, pp. 67 – 76, 1994.