ISCAS 2000 - IEEE International Symposium on Circuits and Systems, May 28-31, 2000, Geneva, Switzerland

VLSI IMPLEMENTATION OF GRBF (Gaussian Radial Basis Function) NETWORKS

İ.C. Çevikbaş[1], A.S. Öğrenci[2], G. Dündar[3] and S. Balkır[3,4]
[1] YITAL, Marmara Research Center, TUBITAK (The Scientific and Technical Research Council of Turkey)
[2] Electronics Engineering Department, Kadir Has University, Istanbul, Turkey
[3] Electrical and Electronic Engineering Department, Bogazici University, Bebek 80815, Istanbul, Turkey
[4] Dept. of Elect. Eng., 209N WSEC 0511, University of Nebraska-Lincoln, Lincoln, NE 68588, USA
E-mail: sbalkir@unlserve.unl.edu

Abstract: A GRBF network is designed for VLSI implementation. Building blocks of the network consist mainly of analog circuits: op-amp, multiplier, multiplying DAC (digital to analog converter), floating resistor, summer and exponentiator. Parameters of the network (center, width of the Gaussian function and output layer weights) are represented digitally for convenient interfacing. It is shown that individual GRBF units allow independent tuning of center, width and amplitude. Several network structures are simulated as function approximation examples, and the performance is verified to be satisfactory.

1. Introduction

GRBF neural networks can be used effectively for both supervised and unsupervised types of learning. They are used for function approximation, where the highly nonlinear Gaussian basis functions provide good locality for incremental learning [1]-[2]. GRBF networks are also used where a similarity measure has to be computed such as in classification and self organization problems. Lastly, GRBF networks are functionally equivalent to Takagi-Sugeno model of fuzzy inference systems [3].

The main advantages of GRBF networks can be summarized as follows: Training time for such networks is generally "order of magnitude" less than the time required for comparable MLP (Multi Layer Perceptron) structures [4]. They also exhibit a higher degree of fault tolerance in comparison to MLP networks [2]. Furthermore, Gaussian function can also be used as the nonlinearity (instead of the sigmoid function) in MLP networks trained by backpropagation. This is reported to perform again "order of magnitude" speed-up in training of pattern recognition problems [5]. GRBF networks are an attractive solution for many neural network applications; however, special hardware implementations of GRBF circuitry are needed for fast and independent operation. The essential mathematical computations to be performed in a GRBF network are as follows:

- computation of cross correlation terms and a weighted summation $r=\Sigma k_{ij}[(x_i-m_i)(x_j-m_j)]$, where x are the inputs, m are the centers and k are related to the widths of GRBF units,
- exponentiation, $exp(-r)$,
- weighted summation of Gaussian outputs.

Several researchers have contributed to analog hardware implementation of GRBF networks by work on building blocks. A pulsed VLSI radial basis function chip is reported in [6]. Churcher et al. [4] proposed circuits for programmable Gaussian operation, that is, center and width of the Gaussian function can be tuned. This is essential for learning. In fact, the learning process adjusts centers and widths of Gaussian functions and weights for their outputs. Choi et al. presented Gaussian synapse cells employing differential pairs for resembling the exponential nonlinearity by piecewise approximation [7]. Exponential characteristic of MOS devices in weak inversion is exploited for gaussian function in [8]. Recently, Lin et al. [9] used current correlator [10] based circuits for realizing Gaussian "bump" type operation where the center, width and amplitude of the bump can be tuned independently. The output is essentially Gaussian-like in weak-inversion operation and square function in strong-inversion. However, there is still a lack of well defined methodology for hardware realization of complete GRBF networks. The aim of this work is to design circuit building blocks and combine them to a GRBF network for satisfactory operation. The results of simulation experiments for three approximation problems demonstrate the validity of the designs.

2. GRBF Neural Network Building Blocks

A multi-input, single-output GRBF neural network consists of a layer of GRBF units with their current outputs summed. Inputs to the GRBF units are x_i (external inputs), and parameters of the unit: m_i (centers of gaussian functions), k_i (cross correlation terms, widths of gaussians) and output weights. Parameters are stored in registers as 8-bit signed numbers. Center values are converted to analog voltage by means of a DAC with an output range of -1V to 1V. The main components in a GRBF unit are the product calculator, summer & exponentiator and weight multiplying DAC. The block diagram of the GRBF unit can be seen in Figure 1 for the case when there are two inputs *x1* and *x2*. *m1* and *m2* are actually voltage outputs of the DAC. In each product calculator (Figure 2), a standard four quadrant Gilbert multiplier computes the product $(x_i-m_i)*(x_j-m_j)$ and produces a differential current which is further multiplied by the width parameter in a multiplying DAC shown in Figure 3. The output of the multiplying DAC is the current $k*(x_i-m_i)*(x_j-m_j)$ where k resembles the width

0-7803-5482-6/99/$10.00 ©2000 IEEE

of the Gaussian circuit. According to the sign bit of the width parameter, the current will come out through the MP or the MN terminal contributing to the LP and LN currents of the product calculator block.

LP and LN nodes of all product calculators are connected together and applied to the summer & exponentiator block (Figure 4). The differential PI and NI currents are mirrored and converted to a single ended current which is then converted to voltage by employing an opamp and a floating resistor. The floating resistor is realized using 6 MOS transistors as described in [11]. Finally, the output voltage of the opamp drives base of an npn transistor for obtaining the exponential current through the transistor which builds a voltage at the collector node (EI in Figure 4). This exponential voltage output will be multiplied by the output layer weight (linear weight) using a multiplying DAC similar to the one described above.

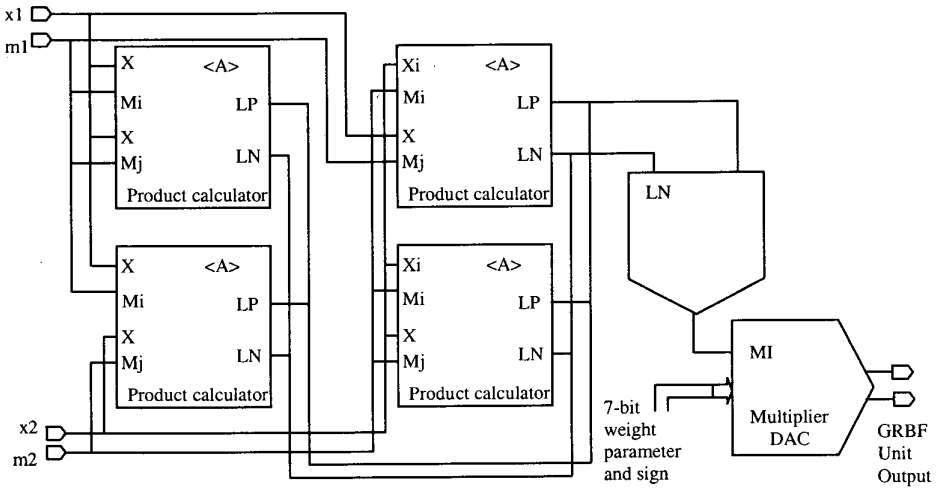

Figure 1. Block diagram of the GRBF (Gaussian Radial Basis Function) unit.

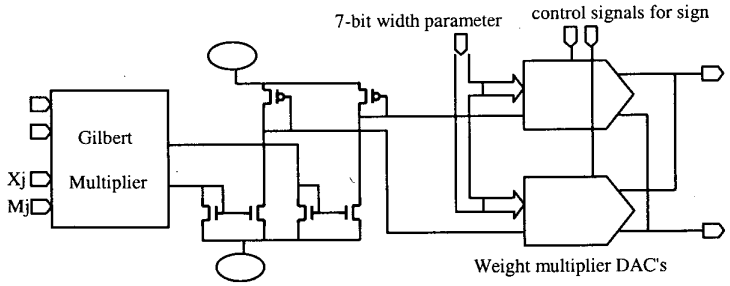

Figure 2. Block diagram of the product calculator unit.

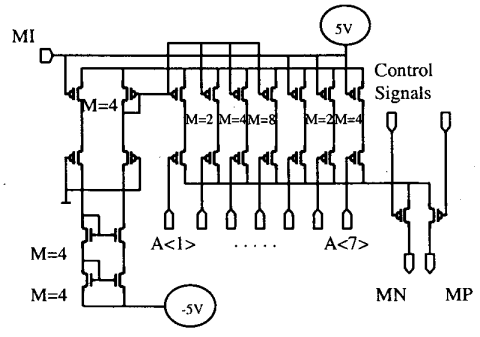

Figure 3. Weight multiplying DAC

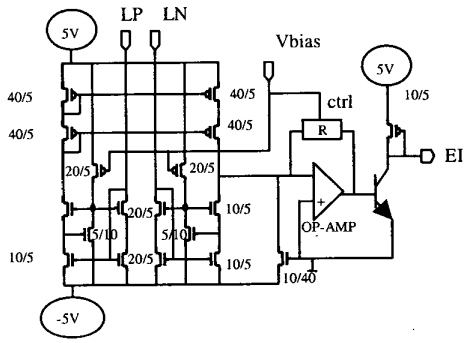

Figure 4. Summer and exponentiator block.

The differential output current of the multiplying DAC in the output stage of the GRBF unit is proportional to the value

$$e^{-\sum k(x_i-m_i)(x_j-m_j)}.$$

A current summer and I-V conversion circuitry similar to the summer of Figure 4 is used to combine GRBF unit outputs. Hence, our GRBF network accepts analog inputs (range between –2V to 2V) and digital parameters, employs current output stages along with multiplying DAC's and gives voltage output.

3. Characteristics of the GRBF Unit

The building blocks described in Section 2 are connected together (with extra digital circuitry for registers) to form a GRBF unit with one input and one output for characterization purposes. For the following simulations +/- 5V supplies are used along with HBIMOS Spice model parameters of the Alcatel Micro-electronics 2μm technology. The tuning performance of the GRBF unit is displayed in Figures 5-7 where it is shown that center, width and peak amplitude of the GRBF unit can be altered independently.

Figure 5 displays tuning of the center of the GRBF unit for maximum peak value. In Figure 6, the width of the Gaussian is adjusted for a center value of 0.5V and for half of the maximum peak value. Last, in Figure 7, peak of the GRBF unit is varied. As can be seen from the figures, the results obtained are quite satisfactory: the parameters of the GRBF units can be tuned independently. The outputs also reveal that symmetrical operation and proper scaling of outputs with respect to weights are achieved. Further simulations also suggest that the maximum values can be altered by simple design modifications in transistor sizing.

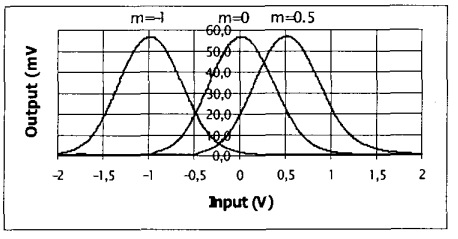

Figure 5. Variation of the center.

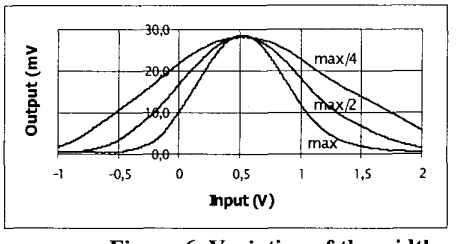

Figure 6. Variation of the width.

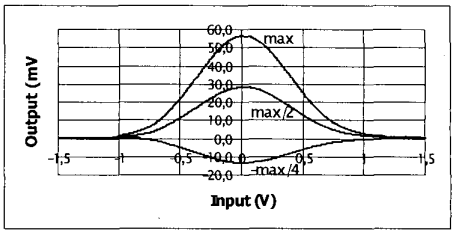

Figure 7. Variation of the peak.

4. Simulation Experiments

In order to verify the circuits described and characterized above, three different single-input, single-output functions are approximated:

- one period of sine function
- square wave
- $f(x)=-35(x+1.1)(x+0.1)(x-0.7)(x-2)$

Parameters for the GRBF network are initially obtained through training of an ideal GRBF network by software. Then, the parameters (center, width and output weight values) are "scaled" so that they are compatible with our circuitry. Center values are just converted to an 8-bit digital code so as to give the same value of analog voltage. Width and output weights are scaled among themselves according to the maximum magnitude obtained. A single fine tuning iteration of these two set of parameters was necessary for the third example, $f(x)$. The issue of proper training will also be addressed in the conclusion.

Parameters for each GRBF unit used in the approximation of the functions are given in Tables 1,3. Simulation results obtained from HSPICE are depicted in Figures 8-10 along with desired (ideal) outputs. As can be seen, the approximation can be regarded as satisfactory. Maximum, average and rms deviations between the desired and simulated values are given in Table 4. Note that the approximation to $f(x)$ is only valid for the range –1V to 1V.

Table 1. Parameters for the sine function.

unit	1	2	3	4
center	-1	-0.5	0.5	1
width	max	max	max	max
peak	0.37max	-max	max	-0.37max

Table 2. Parameters for the square wave.

unit	1	2	3	4	5
center	-1	-0.5	0	0.5	1
width	max	max	max/4	max	max
peak	-max	-max	-max	-max	-max

Table 3. Parameters for $f(x)$.

unit	1	2	3
center	-0.8	0.4	0.8
width	max	0.81max	max/4
peak	max/2	-0.81max	max/4

Table 4. Deviation between simulation and ideal case: % difference with respect to maximum swing.

% deviation	max	average	rms
sine	9.8	3.2	4.2
square	24.0	2.9	4.2
$f(x)$	24.5	10.4	5.6

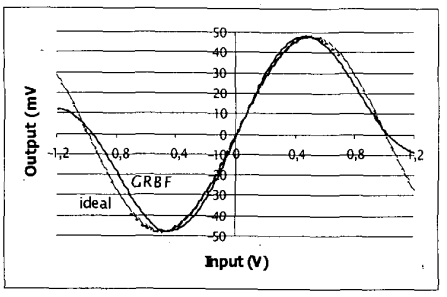

Figure 8. Approximation of sine function.

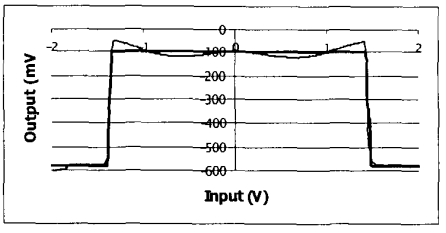

Figure 9. Approximation of square wave

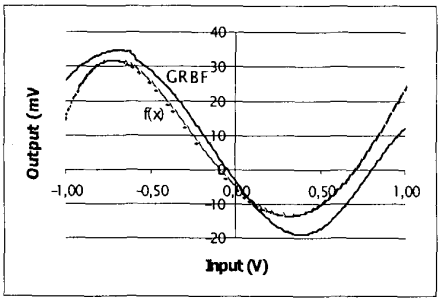

Figure 10. Approximation of $f(x)$.

5. Conclusion

Building blocks for a GRBF neural network unit are designed and combined together with digital registers to incorporate parameters such as center, width and peak amplitude. The GRBF unit exhibits efficient tuning performance with regard to the parameters mentioned above. It has been shown that these units can be used to implement GRBF networks. The essential requirement for proper training of GRBF networks of larger size is that the main blocks of the GRBF unit, namely the product calculator, summer & exponentiator and multiplying DAC's, have to be modeled. Necessary scaling of parameters will then be possible. Modeling based on regression of simulation results of blocks has been applied for backpropagation training of MLP networks successfully [12]; whereas the extension of this approach to GRBF networks with nonideal components needs to be done.

References

[1] S. Aiguo and L. Jiren, "Evolving Gaussian RBF network for nonlinear time series modelling and prediction," *Electronics Letters*, V. 34, No. 12, pp. 1241-1243, June 1998.

[2] S. Lee and R. Kil, "A Gaussian potential function network with hierarchically self organizing learning," *Neural Networks*, V. 4, pp. 207-224, 1991.

[3] K. J. Hunt, R. Haas, and R. Murray-Smith, "Extending the Functional Equivalence of Radial Basis Function Networks and Fuzzy Inference Systems," *IEEE Transactions on Neural Networks*, V. 7, No. 3, pp. 776-781, May 1996.

[4] S. Churcher, A. F. Murray, and H. M. Reekie, "Programmable analogue VLSI for radial basis function networks," *Electronics Letters*, V. 29, No. 18, pp. 1603-1605, September 1993.

[5] J. Platt, "A resource allocating neural network for function interpolation ," *Neural Computation*, V. 3, pp. 213-225, 1991.

[6] D. J. Mayes, A. F. Murray, and H. M. Reekie, "Pulsed VLSI for RBF Neural Networks," in *Proceedings of Int. Conf. On Microelectronics for Neural Networks, Evolutionary and Fuzzy systems*, pp. 174-184, Lausanne, 1996.

[7] J. Choi, B. J. Sheu, and J. C.-F. Chang, "A Gaussian Synapse Circuit for Analog VLSI Neural Networks," *IEEE Transactions on VLSI Systems*, V. 2, No. 1, pp. 129-133, March 1994.

[8] J. Madrenas, M. Verleysen, P. Thissen, and J. L. Voz, "A CMOS Analog Circuit for Gaussian Functions," IEEE Transactions on Circuits and Systems – II, V. 43, No. 1, pp. 70-74, Jan. 1996.

[9] S.-Y. Lin, R.-J. Huang, and T.D. Chiueh, "A Tunable Gaussian/Square Function Computation Circuit for Analog Neural Networks," *IEEE Transactions on Circuits Systems-II*, V. 45, No. 3, pp. 441-446, March 1998.

[10] T. Delbuck, "'Bump' circuits for computing similarity and dissimilarity of analog voltage," in *Proceedings of Int. Neural Networks Society*, Seattle, 1991.

[11] M. Ismail and T. Fiez, *Analog VLSI Signal and Information Processing*, NY, McGraw Hill, 1994.

[12] İ. Bayraktaroglu, A. S. Ogrenci, G. Dundar, S. Balkır, and E. Alpaydın, "ANNSyS: An Analog Neural Network Synthesis System," *Neural Networks*, Vol. 12, pp. 325-338, 1999.