A Simple MOSFET Model of an Artificial Synapse

MOMCHIL MIHAYLOV MILEV
Faculty of Electronic Engineering and Technology[1]
Technical University of Sofia
8 St. Kliment Ohridsky Blvd, Sofia
BULGARIA

Abstract: A simple analog-signal synapse model is developed and afterward implemented on a standard 0.35μm CMOS process to provide for large scale of integration, high processing speed and manufacturability of a multi-layer artificial neural network. Synapse non-linearity with respect to synapse weight is studied. Demonstrated is the capability of the circuit to operate in both feed-forward and learning (training) mode. The effect of the synapse's inherent quadratic nonlinearity on learning convergence and on the optimization of weight vector update direction is analyzed and found to be beneficial. The suitability of the proposed implementation for very large-scale artificial neural networks is confirmed.

Key-Words: Artificial neural networks (ANNs), analog implementation, Very Large Scale of Integration (VLSI), nonlinear synapse, synapse with nonlinearity.

1 Introduction

The signal processing speed, scale of integration, low power consumption and manufacturability of nowadays ANNs determine their feasibility and usage in real-life applications. Due to conflicting requirements in lowering the supply voltages and increasing clock speeds of the digital circuits, many researchers consider analog implementations of neural networks as a way to carry over signal processing functions with fewer numbers of active semi-conductor devices. The integration of large numbers of neurons in a single chip is beneficial since it increases the VC-dimension[1][2]. It requires the minimization of the synapse area and a more efficient way of data exchange between neurons to be devised. In this respect, analog implementations offer certain benefits making them good contenders for real-time applications. First, they offer high processing speed since the analog signal processing is carried out through summation and multiplication of continuous current or voltage signals with virtually no delay. Second, analog implementations, typically, can have larger scale of integration since they avoid data-path organization which often requires data multiplexing, bus sharing, and data-flow control logic, further limiting the effective rate at which digital neural circuits can process input signals. The main disadvantages of the analog-based designs of ANNs are considered to be their lower accuracy and the difficulties with linearity in the computations. These two factors are challenged in this article. First, it is demonstrated that the term "absolute accuracy" is often of lower significance with respect to the ability of a neural network to function in many practical applications. Second, it is demonstrated that ideal linearity in the multiplication computations is not necessarily desirable or even required In most cases, nonlinearity in the synapse transfer function is, in fact, beneficial[11][12][13]. This article is limited to the discussion of the quadratic nonlinearity in the synapse multiplication function of a specific analog implementation.

[1] Momchil Milev is currently employed by Texas Instruments Inc, Tucson, AZ. The information contained in this paper is not subject to intellectual property, trademark or nondisclosure agreements, should not be affiliated with Texas Instruments Inc nor it does describe a product by TI or its subsidiaries.

The paper is structured as follows. Section 2 describes the proposed analog, nonlinear, one-transistor synapse model, explains the motivation behind avoiding use of floating-gate devices. Section 3 examines the inherent nonlinearity of the synapse with respect to its weight. Synapse model functional verification results follow brief extracts from our analytical research on the effects of the quadratic nonlinearity on the feed-forward and LMS training. Results of our circuit simulations and system-level MatLab™ verification of an artificial neuron acting as linear classifier are presented next. Summary and conclusions wrap up the paper.

2 Model overview

By using the physics of analog devices, analog implementations of ANNs offer the advantage to carry out synaptic function with only a small number of transistors. In order to benefit fully from the simple current summing law and avoid parasitic capacitive load delays, we chose pre- and post-synaptic activity signal to be represented by analog current. To simplify synapse design and minimize synapse silicon layout area, as well as to allow for ANN on-chip integration with other standard CMOS products, it is decided to implement synapse multiplication function by a simple single semiconductor device—a MOSFET.

From the first-order DC, large-signal low frequency approximation[2] model (1) of a MOSFET's drain current in non-saturated region of operation $v_{DS} \le (v_{GS} - V_T)$, after simplification[3][3][4],

$$i_D = \beta\left[(v_{GS} - V_T)v_{DS} - \tfrac{1}{2}v_{DS}^2\right](1 + \lambda v_{DS}) \qquad (1)$$

we use the product of the gate-source and drain-source voltage to produce one of the components of the synaptic activity value defined by[5]:

$$\nu = \sum_{k=1}^{N} w_k x_k \qquad (2)$$

Summing the currents of those "partial products", we get the complete "sum of the weighted products". To express this, we consider a single synapse, *k*, and define:

$$\nu_k \triangleq \tfrac{i_D}{\beta}; x_k \triangleq (v_{GS} - V_T); w_k \triangleq v_{DS};$$

Next, from (1), we derive a generic form of the quadratic nonlinearity of the synapse's internal activity field with respect to its weight:

$$\nu_k = x_k w_k - \zeta w_k^2 \qquad (3)$$

where ζ is a constant ($\zeta = 0.5$). We chose the above definitions due to practical considerations- to provide for signal values that are of the same or close order of magnitude. Nevertheless, the results in this text are more generic and can be applied to other, similar to expression (3), non-linear relationships, provided that the relationship can be approximated linearly within a certain operational range.

For N -number of synapses, the overall synaptic activity is:

$$v = \sum_{k=1}^{N} \nu_k = \sum_{k=1}^{N} x_k w_k - \zeta \sum_{k=1}^{N} w_k^2 \qquad (4)$$

Based on these considerations, a single MOSFET device offers a simple way of constructing a "linear combiner" in hardware. Its main advantage over single-transistor synapses, implemented in analog-floating-gate capable technologies, is that it does not require any special fabrication technology, and thus it is easily integrated with other standard CMOS applications to build a complete system-on-a-chip (SoC). Floating-gate technology is available in most "standard" CMOS processes; however, it is most often used for binary information storage. In order to reach a 9-bit or better analog storage resolution more specialized and expensive floating-gate fabrication technology is required. Additionally, analog floating-gate control circuits are complicated and small weight updates are difficult [14].

The proposed synapse model is inherently nonlinear but simple enough in its implementation to occupy a very small silicon area, making it very useful in VLSI systems. Further, we show that this nonlinearity is not detrimental to the qualities of the proposed synapse but, in fact, could be beneficial. We also include circuit simulation and system-level behavioral simulation

[2] For $V_{GS} \ge 1.0$ V, $V_{DS} \le 100$mV and $V_{SB} = 0$V, second-order effects, including channel-length modulation, short-channel and temperature effects are estimated to contribute an average error of -5.1%. This error, however, is considered included in the overall nonlinearity of i_D and does not change the applicability of the considerations given.

[3] For typical operating drain-source voltage ($v_{DS} < 100mV$) in non-saturated mode of operation, channel-length modulation contributes error of no more than 0.02% which is ignored in further consideration.

results that support the feasibility of using such nonlinear synapses as building blocks of ANNs.

3 Effects of the nonlinearity

3.1 Effect of synapse quadratic nonlinearity in feed-forward mode

To show the effect of the quadratic nonlinearity with respect to synapse weight, due to the described implementation, we evaluate the error defined by:

$$\varepsilon_{lin} = \frac{(\nu - \nu_{ideal})}{\nu_{ideal}} 100 \quad ,[\%] \tag{5}$$

Expressed in terms of synapse transistor quantities:

$$\varepsilon_{lin} = -\zeta\left(\frac{w}{x}\right) = -\frac{1}{2}\frac{v_{DS}}{v_{GS} - V_T} 100 \quad ,[\%] \tag{6}$$

From (6) we note that the linearity error does not depend on transistor transconductance parameters i.e., on process or geometrical parameters. For a typical signal range ($v_{DS} = 100mV,\ v_{GS} = 1.0V,\ \mathrm{V_T} = 0.65V$), we estimate this nonlinearity "error" to be less than 15% (14.29% worst-case). We could apply an input bias

$$x_0 = -\theta\left(\zeta\sum_{k=1}^{N} w_k^2\right)^{-1} \tag{7}$$

to an extra synapse (theta-synapse) to eliminate this "offset" error[4] in feed-forward mode if needed. In feed-forward mode this bias term is a known constant, thus we could eliminate this term after network training is complete and weights are known. Such correction, however, is not applied in the experiments shown since it is our belief that this inherent offset term is accounted for by the Back-Propagation algorithm during training and, thus, it can be treated by the adaptive process as "constant input noise".

3.2 Effect of synapse quadratic nonlinearity in least-mean-square (LMS) training

To study the effects of the "offset" term in (4), we use the instantaneous estimate of the gradient and the method of steepest descent in LMS training:

$$\varepsilon(\mathrm{W_N}) \triangleq \tfrac{1}{2} e_{\mathrm{N}}^2 \tag{8}$$

$$\mathbf{W_{N+1}} = \mathbf{W_N} - \eta.\nabla\varepsilon_{\mathbf{W}}(\mathbf{X_N}, \mathbf{W_N}) \tag{9}$$

Using equation (4) for the method of steepest descent we obtain weight update rule in vector format:

$$\nabla\varepsilon_{\mathbf{W}}(\mathbf{X_N}, \mathbf{W_N}) = (-\mathbf{X_N} + 2\zeta\mathbf{W_N})e_N \tag{10}$$

$$\mathbf{W_{N+1}} = \mathbf{W_N} + \eta\mathbf{X_N}e_{\mathbf{N}} - 2\eta\zeta\mathbf{W_N}e_{\mathbf{N}} \tag{11}$$

A corresponding weight-update vector diagram is shown in Figure 1. We define the difference between the update vector in the case of an ideally linear synapse output and the case of a nonlinear synapse with quadratic weight-nonlinearity as a 'residual weight gradient vector':

$$\mathbf{W}_R \triangleq 2\zeta\mathbf{W_N}e_{\mathbf{N}} \tag{12}$$

We then define a 'modified'[5] instantaneous error gradient vector estimate:

$$\widetilde{\nabla}\varepsilon_{\mathrm{W}}\big|_{\mathrm{N}} = -\mathbf{X_N}e_{\mathrm{N}} + \mathrm{W}_R \tag{13}$$

and then re-write the weight update rule(11):

$$\mathbf{W_{N+1}} = \mathbf{W_N} + \eta(\mathbf{X_N}e_{\mathbf{N}} - \mathbf{W_R}) \tag{14}$$

We have analyzed the effect of the modified gradient vector in two ways: effect on the direction of the weight-vector update, and effect on the magnitude (norm) of the update. We have concluded that:

1. The modified weight update vector due to (12) is always rotated in the direction of the input vector

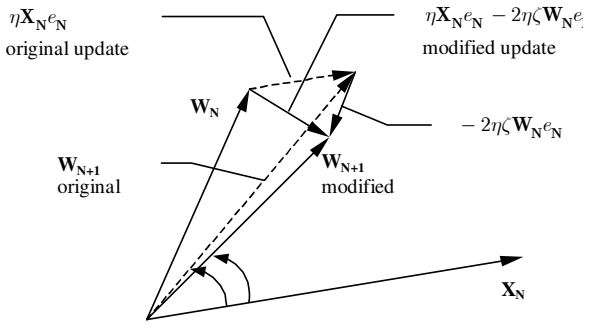

Figure 1 Modified weight vector update diagram due to 'residual weight vector' term

[4] in several applications, this nonlinearity in feed-forward mode proved not relevant to the success of the network for correct classification due to flexibility in the output space definition and, therefore, correction was not necessary

[5] Compared to original LMS steepest descent method

regardless of the input or weight vectors relative position or size. Thus, in both angle and Euclidean-distance sense the new weight-update vector is closer to the input vector when compared to traditional LMS with gradient descent.

a) An angle exists between the input and weight vectors

$$\alpha_{critical} = arc(\zeta \frac{\|W\|}{\|X\|}); \alpha_{critical} \in [0; \pi] \qquad (15)$$

for which:

- magnitude of the modified update is larger than the norm of the update in the original LMS steepest descent method[6] if $\measuredangle(X_N, W_N) > \alpha_{critical}$
- magnitude of the modified update vector is smaller compared to the norm of the original method if $\measuredangle(X_N, W_N) < \alpha_{critical}$

Therefore, the effect of the 'residual weight gradient vector' on the adaptation is considered beneficial – increasing the amount of update, hence, speeding up the convergence of the weight vector when it is 'far' from the steepest descent direction and decreasing the amount of update for weight vectors close to the direction of steepest descent [6][10]. The latter is considered helpful in avoiding weight-vector oscillations around the optimum solution for increased learning-rates, thus, again providing faster convergence conditions.

Additionally, by expanding the error cost function in a Taylor series around the weight vector at any given time, it has been proven that the error is minimized with every step of the iterative descent regardless of the modification due to the residual weight gradient vector i.e. synapse quadratic nonlinearity with respect to its weight. A comparative analysis was also conducted between the modified update(11) and the generalized 'delta rule' including the 'momentum term' as it is known by Rumelhart et al [7]. It was concluded that, while the use of the momentum term can decrease the stable range of the learning rate parameter and lead to instability [8][9], the effect of the residual weight vector, in contrast, does not decrease the learning rate range and is stabilizing inside the $\alpha_{critical}$-determined spatial cone. The details of this research, however, are outside of the scope of the present article and are not included here. More information on training ANNs with non-linear synapses can be found in [15][16].

[6] For same learning rate and instantaneous error amount

3.3 Experimental data

To verify and support the theoretical findings, a number of circuit-level and system-level simulations were carried out. Circuit level simulation results and plots for nonlinear synapse operation, weight-charging, input-signal conversion and others are exhaustive and available from the author upon request[17][19]. System-level simulations were conducted using MatLab™ software to train and test a neuron using synapses with quadratic nonlinear synapses as modeled by (3) to perform a linear classifier function. Sets of 2D linearly separable clusters of random vectors were generated and then LMS steepest descent training was performed over the same data twice – once for a neuron having ideally linear synapses and again for the described model of a neuron with nonlinear synapses. More than 200 simulation runs over clusters of 100 vectors with varying cluster size and dispersion were evaluated. The results showed [10] that the classification success of the neuron using nonlinear synapses modeled by (4) was, generally, not lower than the success rate of the correct classification of the neuron with linear synapses, and in many instances was better. Additionally, in most cases, convergence during the training of the neuron using nonlinear synapses was reached in fewer epochs than for the case of the neuron with ideally linear synapses. The results for the original neuron with ideally linear synapse are depicted by an 'o'-symbol and the results of the neuron with quadratic nonlinearity in the synapses are shown with an 'x'-symbol. Selected plots showing the final MSE, the number of epochs in which convergence was reached, final learning rate parameters and rate of successful classification for the training and test runs in the two cases are shown in Figure 2 through Figure 6.

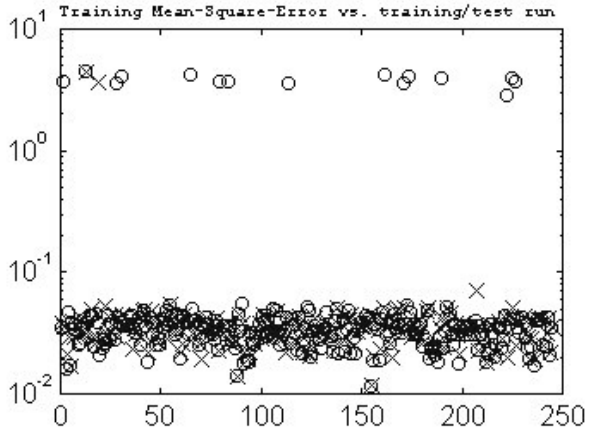

Figure 2 Final MSE vs. training/test run number

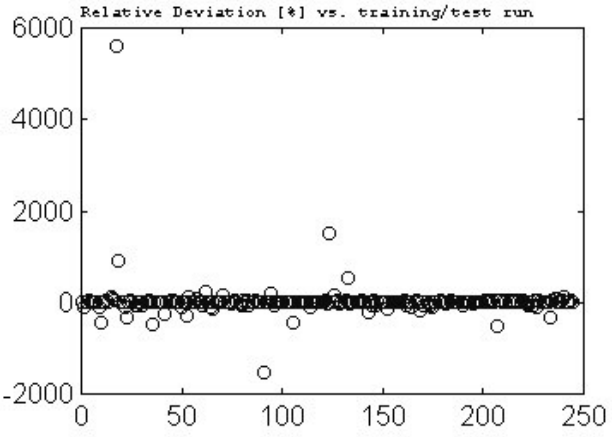

Figure 3 Relative deviation in percent of the output response of a neuron using nonlinear synapses with quadratic nonlinearity vs the response of a neuron with linear synapses

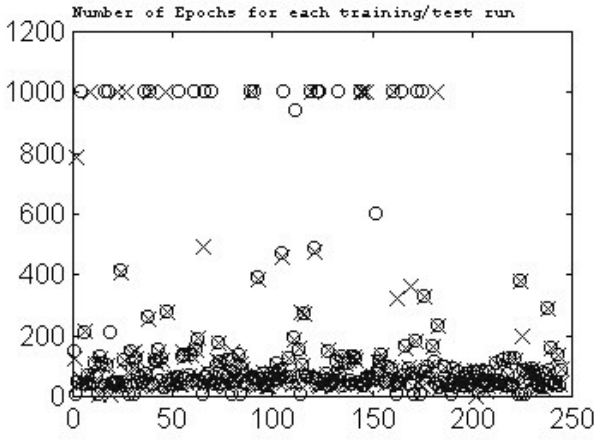

Figure 4 Number of epochs in which convergence was reached in the case of the original linear synapse neuron model and neuron using nonlinear synapses

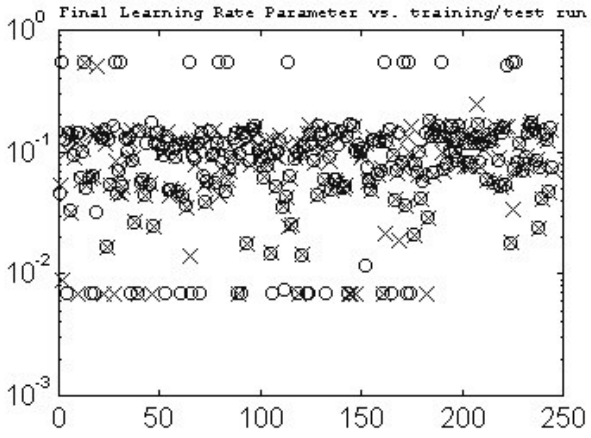

Figure 5 Final learning rate parameter for each training run

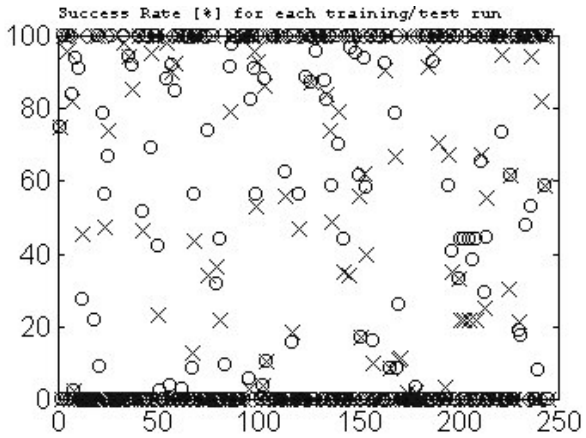

Figure 6 Rate of successful classification of the neuron with linear synapses -'o' and with non-linear synapses -'x'

4 Summary and Conclusion

We show that it is feasible to implement an analog synapse using only basic properties of MOSFETs in a standard CMOS fabrication process. We describe the model and investigate its operation in feed-forward and learning modes of operation. Due to limited size of this presentation, we show the complete design of the synapse circuit, a neuron and ANN based on this model in [18] and [19]. We demonstrate that not only is the inherent quadratic non-linearity with respect to synapse weight not detrimental to the ability of the synapse to function in LMS training mode, but also that the latter can offer distinct advantages in learning convergence. We do so by offering results from both theoretical and experimental research we have conducted. We suggest that a simple synapse circuit, based on this synapse model can prove useful in VLSI systems-on-a-chip and we further exploit this topic in [19] to demonstrate its feasibility for on-chip integration with other CMOS products.

5 Acknowledgments

Most of all, I would like to thank Cynthia Roedig, Texas Instruments Inc, for proofreading the first draft and for her help in improving phrasal expressions. I very much thank Ivaylo Milev for his contribution in improving clarity of the original manuscript and overall presentation structure. I thank my wife, Diana for her support and encouragement. Also, I thank Vadim Ivanov, Texas Instruments, for his suggestion to present this paper at WSEAS-2004 conference.

REFERENCES

[1] A.A. Blumer, Ehrenfeucht, D. Haussler and M.K. Warmuth, "Learnability and the Vapnik-Chervonenkis Dimension", *Journal of the Association for Computing Machinery*, vol.36, pp.929-965, 1989.

[2] R.W. Newcomb, N. El-Leithy, "Perspectives on realizations of neural networks", *CAS,IEEE International Symposium on Neural Networks,* pp.818-819 vol.2, 1989.

[3] Phillip E. Allen, Douglas R. Holberg, *CMOS Analog Circuit Design*, Oxford University Press, 1987; pp.98-101, pp.198-207.

[4] Roubik Gregorian, Gabor C. Temes, *Analog MOS Integrated Circuits For Signal Processing*, A Wiley-Interscience Publication, John Wiley & Sons, New York, 1986; pp.462-483.

[5] Simon Haykin, "*Neural Networks – A Comprehensive Foundation*", second edition, 1999 Prentice Hall, New Jersey; pp. 3-18, pp.51-53, pp.121-122, pp.128-132.

[6] M. Milev, "Hardware model of a neural synapse with nonlinearity", *Tenth National Application-Scientific Conference on Electronic Technology with International Participation ET-2001*, Sozopol, Bulgaria, 2001.

[7] D.E. Rumelhart, G.E. Hinton, R.J. Williams, "Learning representations of back-propagation errors", *Nature (London)*, vol.323, pp.533-536.

[8] M. Hagiwara, "Theoretical derivation of momentum term in back-propagation", *International Joint Conference on Neural Networks*, vol. I, pp.682-686, Baltimore 1992.

[9] Simon Haykin, "*Neural Networks – A Comprehensive Foundation*", second edition, 1999 Prentice Hall, New Jersey; pp.170-171, p.249 note 2.

[10] M. Milev, "LMS training of a PE with non-linearity with respect to weights as a linear classifier", *Ninth National Conference on Electronic Technology with International Participation- ET2000*, Sozopol, Bulgaria, 2000.

[11] Liang, P.; Jamali, N. , "Artificial neural networks with nonlinear synapses and nonlinear synaptic contacts", *Systems, Man and Cybernetics, 1992., IEEE International Conference on , 1992* pp.1043 -1048 vol.2

[12] Lont, J.B.; Guggenbuhl, W, "Analog CMOS implementation of a multilayer perceptron with nonlinear synapses", *IEEE Transactions on Neural Networks ,* vol.3 issue 3 , May 1992 pp. 457 -465

[13] Lont, J.B., "A large scale neural network with nonlinear synapses", *1994 IEEE World Congress on Computational Intelligence., 1994 IEEE International Conference on Neural Networks*, Volume: 1 , 1994 pp. 350 -353 vol.1

[14] Horio, Y.; Ymamamoto, M.; Nakamura, S, "Active analog memories for neuro-computing", *IEEE International Symposium on Circuits and Systems, 1990*, 1-3 May 1990 pp. 2986 -2989 vol.4.

[15] Nakayama, K.; Hirano, A.; Fusakawa, M., "A selective learning algorithm for nonlinear synapses in multilayer neural networks", *Proceedings. IJCNN '01. International Joint Conference on Neural Networks, 2001*, Volume: 3, 2001, pp. 1704 -1709 vol.3

[16] Myung-Ryul Choi; Jin-Sung Park , "Implementation of MEBP learning circuitry with simple nonlinear synapse circuits", *Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE '99. 1999 IEEE International* , Volume: 1 , 1999 pp. 315 -320 vol.1

[17] Milev, Momtchil Mihaylov , "Method and apparatus for modeling a neural synapse function by utilizing a single conventional MOSFET", United States Patent Application No.09/968,263. Pub. No. US2002/0057606 A1 May 16, 2002, http://www.uspto.gov.

[18] M. Milev, M. Hristov, "Analog Implementation of ANN with Nonlinearity in Synapses", *NNS,IEEE Transactions on Neural Networks, Special issue on hardware implementations* pp.1187-1200 vol.14, 2003.

[19] M. Milev, "VLSI Implementation of an Artificial Neural Matrix with Analog Nonlinear Synapses", *Third WSEAS International Conference in Applications of Electrical Engineering*, AEE'04, May 12-15, 2004.