A VLSI-FRIENDLY SYNAPSE MODEL AND ITS APPLICATION FOR FAST AND CHEAP SIGNAL CLASSIFICATION PROBLEMS WITH NEURO-FUZZY NETWORKS

by

R. Dogaru and L. O. Chua

Memorandum No. UCB/ERL M97/5

16 January 1997

A VLSI-FRIENDLY SYNAPSE MODEL AND ITS APPLICATION FOR FAST AND CHEAP SIGNAL CLASSIFICATION PROBLEMS WITH NEURO-FUZZY NETWORKS

by

R. Dogaru and L. O. Chua

Memorandum No. UCB/ERL M97/5

16 January 1997

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A VLSI-friendly Synapse Model and its Application for Fast and Cheap Signal Classification Problems with Neuro-fuzzy Networks

R. DOGARU[1], L. O. CHUA,
University of California Berkeley, 258 M Cory Hall,
Berkeley, CA-94720 1770
E-mail: radu_d@fred.eecs.berkeley.edu

Abstract

A new model for both synapse and its associated learning rule is proposed. Simulation results obtained by replacing the multiplier-based synapse with the novel one in an easy to train neuro-fuzzy network proved that for signal classification problems there are no major differences in the overall system performances, while the new, *comparative* synapse, is much attractive in terms of VLSI or software implementation and it also offers the advantage of a simple implementation for on-chip learning.

1. Introduction

Artificial neural networks may be efficiently used in signal classification tasks when there is very little or no knowledge about the signal source model. Such problems arise often in speech recognition, bio-medical data classification, telecommunications and many other fields. In image processing, there is by now a well established theory and design techniques for Cellular Neural Networks [Chua93] which allow very fast image processing at convenient costs when such systems are implemented in VLSI technology. In many cases, processing speed and implementation costs are important issues and thus, one may select from the numerous artificial neural or neuro-fuzzy systems proposed in literature ([Hasn95],[Hayk94],[Lin96]) only those which may be easily "on-line" trained (i.e. without local minima on the error surface which implies restarting the learning process) in a reasonable small number of epochs (number of pattern presentations) and having a low cost implementation.

When dealing with digital hardware, cost is related with area occupied on silicon die or with the number of basic logical gates while when dealing with software implementation one may think about minimising the computation time and developing such models that may not require additional math-processors for low cost designs. On the other hand, it is well known that synapses are the most important parts of neural systems, minimising the area or computation time associated with them being the main goal of the neural network engineers [Sheu95]. With only few exceptions (e.g. [Clou94],[Lont92] or [Whit92]), the multiplier-based synapse model is mostly used. For this synaptic model, the state v_j of the "target" neuron "j" is computed as: $v_j = \sum_i w_{j,i} \cdot y_i$ where y_i denotes the output of a "source" neuron "i" in the network, and $w_{j,i}$ denotes a numerical value associated with the synapse "i-j" often called weight and which

[1] On leave from University "Politehnica" of Bucharest with a Fulbright grant (permanent e-mail address: radu_d@atm.neuro.pub.ro and http://atm.neuro.pub.ro/~radu_d)

is the basis for distributed learning and information processing in neural systems. A very common mathematical operator, multiplication is however expensive to implement, particularly in digital systems. Different solutions to this problem were proposed in both signal processing [Jiang89] and neural networks literature [White92] which are mainly exploiting the idea of approximating multiplication with simple operators based on shift registers. Despite their computational efficiency, these methods have the disadvantage that they are not universal and for each particular signal processing system a new design procedure must be developed and sometimes there may be a loss in the quality performances of the overall system.

Neural or neuro-fuzzy systems are adaptive systems, and it is sure that biological synapses are not implementing a specific mathematical operator as the multiplier one, but such subsystems as a whole are perfectly functional for different tasks. Thus, we consider that for this class of systems, adaptation is an important feature that may be exploited not only for learning but also for giving us more freedom to chose such models for parts (neurons, synapses) that are most convenient to implement in our given technologies. Some particular results of this way of thinking were described in [Dgr96a] and [Dgr96b], the main feature of the systems designed according with this principle being a more efficient implementation without loosing the basic functionality which is measured as performance indicators related to the specific problem. In this paper, we extended this principle to the problem of signal classification with feed-forward networks by proposing a new synapse model which we called "comparative" synapse. Here, we considered supervised classification problems [Dud73] where the input patterns are analog feature vectors (which may be signal samples or outputs from adequate pre-processing systems) and the desired outputs are binary values representing bits in label words associated with classes to be recognised. The system to be trained was considered a neuro-fuzzy architecture proposed in [Yama92] which is very convenient in terms of training speed and convergence since the adaptation take places in a simple linear perceptron layer. Adaptation of this architecture for classification task is presented in section 2. The novel synapse model and a simple learning rule will be introduced in section 3, while in section 4 simulation results are presented for a set of 6 benchmark classification problems when the standard multiplier-based synapse (linear perceptron) was replaced by the comparative one in the output perceptron layer of the feed-forward neuro-fuzzy network. Conclusions and further research perspectives are presented in section 5.

2. A neuro-fuzzy architecture for classification tasks

Different neural or neuro-fuzzy systems for classification purposes were proposed in literature in the last decade. Among these, the Multi-layer Perceptron (MLP) trained with backpropagation gained a high popularity. However, it has a series of drawbacks which makes it non-attractive for fast and easy to implement signal processing tasks. These drawbacks are mainly related with the need to propagate errors trough the network, with very complex error surfaces which often implies convergence problems and with difficulties in selecting an optimal architecture (i.e. number of layers, neurons in layers) for a specific problem. While our purpose was mainly to test the performances of a new synapse against the classical multiplier-based one, we selected for this study a much convenient and easy to train neuro-fuzzy structure which was proposed in [Yama92] for signal prediction and system identification tasks. This network (see Fig. 1), for which fast learning capabilities and good performances were reported for chaotic signal prediction, may be considered as a particular case of the more general ANFIS [Jang95] model which also includes radial basis function (RBF) networks as a particular case.

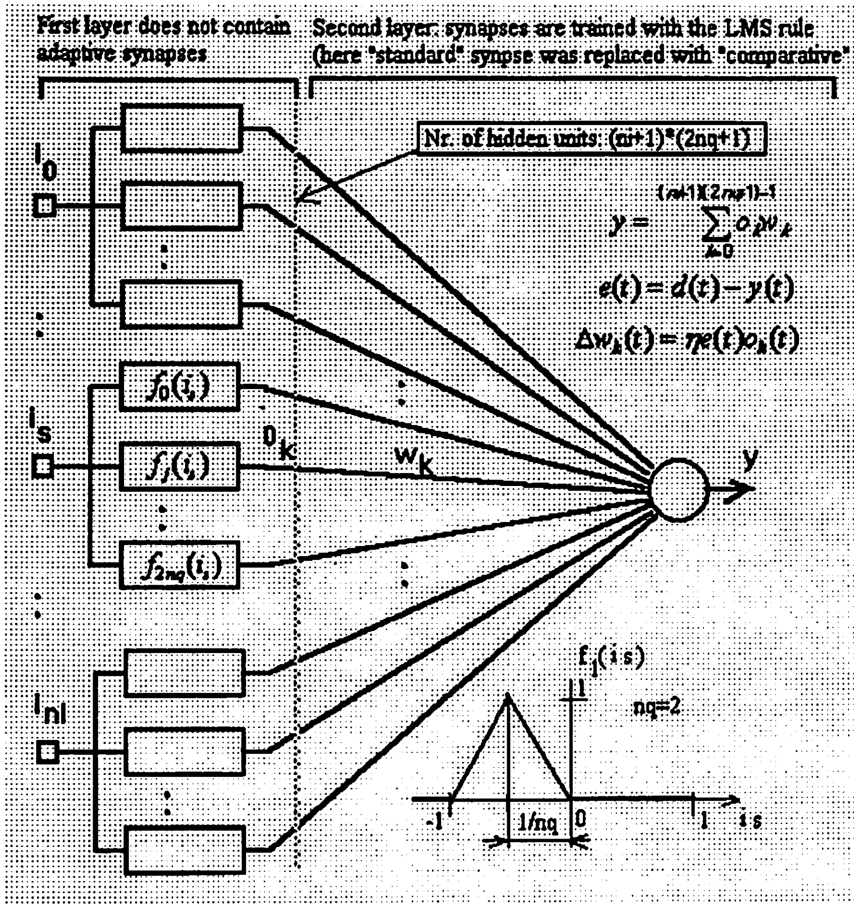

Figure 1: The fast learning neuro-fuzzy structure; multiplier-based synapses in the output layer may be replaced with the novel comparative synaptic model.

The main and most attractive feature comparing with other adaptive non-linear systems is that it has only one adaptive layer (the output one) which collects outputs o_k from a non-trainable "hidden" layer. This is composed by non-monotone non-linear units which are related with overlapping fuzzy-membership functions $f_j(I_S)$ used to quantify in a soft way with *nq* subdomains any particular input I_S which may vary within [-1,1] domain. Instead of RBF networks, there is no need to train the input layer while the shapes and the centres of each fuzzy-membership function are previously defined and are only dependent on the specific number *nq*. Thus, the only one parameter that may be varied in order to find an optimal performance is *nq*.

While in the original paper [Yama92] this architecture and its learning algorithm were derived using fuzzy-logic theory and it is used only for signal prediction task, we can also consider it as a hybrid system based on a non-linear and non-trainable pre-processor corresponding with the fuzzy-membership functions which is followed by a simple adaptive system. When standard synapses are used, the output layer may be considered as a simple linear perceptron [Widr60] which is easy to train with guaranteed convergence to the optimal solution. When the input vectors are samples belonging to different non-linear separable classes (a classification problem), Cover's theorem [Cov65] states that a non-linear preprocessing system may be found which extends the dimensionally of the input vector but makes the output image linear-separable. The theorem is not constructive i.e. does not give a method to build this non-linear preprocessing system. Thus, we may consider the fuzzy-preprocessing proposed in [Yama92] as a particular solution where for an optimal *nq* value, the output vectors of the "hidden" layer become linear separable and thus the following linear perceptron is able to find an optimal solution for the classification problem. Of course, the value of *nq* will depend on the particular problem being thus related with the shape of the non-linear border between classes.

We may go further and extend the linear perceptron to a more general (non-linear) one where the multiplier-based synapses will be replaced by a more general two-variable function and thus, the perceptron output will be described by: $y = \sum_k \text{synapse}(o_k, w_k)$. In what follows, we will introduce a particular model for this synapse and a learning (update) rule showing by computer simulations that overall system performances are not strongly dependent on the particular synapse model and thus we have freedom to choose the most convenient to implement one.

3. Comparative synapse model and the associated learning rule

3.1. The comparative synapse model

A general synapse model may be written as $y_k = synapse(o_k, w_k)$ where o_k denotes a synaptic input coming from another neuron output in the network, w_k is the weight value associated with the distributed memory and y_k denote the synaptic output. The ***multiplier-based*** synapse used in linear perceptrons is described by:

$$y_k = o_k \cdot w_k \tag{1}$$

while the novel model that we propose (comparative synapse) is described by:

$$y_k = \text{sign}(o_k) \cdot \text{sign}(w_k) \cdot \min(|o_k|, |w_k|) \tag{2}$$

This model is related with fuzzy-AND operators [Lin96] or T-norms and a slightly different version is used in Fuzzy-ART [Carp91] neural networks for computing the activation of the output layer neurons. While the main operation is a comparison between input and weight values we will call it a *comparative* synapse. Such synapse is used to replace the multiplier-based one used by the output linear perceptron of the neuro-fuzzy network (Fig. 1). In order to train the network, we have to know the output error $e = d - y$ for each pattern presentation, where d denotes the desired output. For classification problems it is convenient to consider each desired output as having only the values +1 or -1. During the retrieval phase, the linear output neuron is replaced with a hard-limiter one ($y = \text{sign}\left(\sum_k \text{synapse}(o_k, w_k)\right)$) preserving the weight values obtained during learning [Widr60].

3.2. Implementation issues

In what follows we consider that a multiplication device is needed to find the product $p = x \cdot y$ only if both x and y magnitudes are expected to vary during a learning or retrieval process within a specified domain $\Omega = [\varepsilon, 1]$ where $\varepsilon = 1/2^{res}$, this domain corresponding with a fixed point arithmetic using *res* bits. For a fast speed combinational implementation such a digital multiplier requires $O(res^2)$ logical gates. However, when at least one of x *or* y is a power of 2, the multiplier device can be replaced with a shift register having only $O(res)$ complexity. When one of the two operands is 1 or -1, the multiplier device becomes a simple gate for changing the sign and thus having the lowest $O(1)$ complexity. Other functions used in connection with the novel synapse model are the "sign", "step", "min" and absolute value. Here, $\mathrm{sign}(x) = -1$ if $x < 0$ and $+1$ if $x > 0$ and $\mathrm{step}(x) = (1 + \mathrm{sign}(x))/2$. Both these functions and the absolute value function may be implemented in digital technology with $O(1)$ complexity while the "min" function require $O(res)$ complexity.

For the comparative synapse implementation we will also assume that both weight and synapse input magnitudes are restricted to vary within Ω in order to ensure a simple VLSI implementation. A graphical comparison of the two synaptic functions (multiplier-based and comparative) is shown in Fig. 2. for this restricted case.

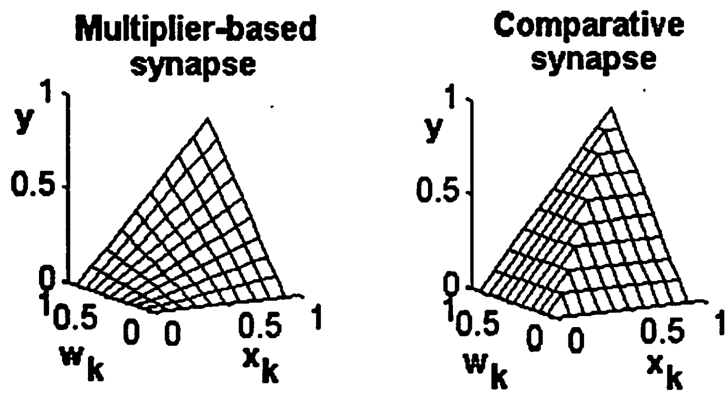

Figure 2: A graphical comparison between the two synaptic models ; synapse output is represented on the vertical axis.

One may also consider the comparative synapse as a two variable piece-wise linear approximation of the multiplier-based synapse. Some useful properties of such approximations were derived in [Chua77].

3.3. A learning rule proposal for the comparative synapse

In order to get fast learning in non-linear adaptive systems incremental learning is considered [Hayk94]. In this type of learning, for each moment t when a new pattern is applied at the input, a weight update $\Delta w_k(t) = w_k(t+1) - w_k(t)$ is computed based on the actual error $e(t)$ obtained in respect with the desired output $d(t)$. In what follows we will consider only the moment t and for sake of simplicity we will not explicitly write the discrete-time variable. The goal is to minimise a pre-defined goal function which is often an error average over all possible patterns. While is impractical to use such a goal function, in incremental learning one seeks to minimise only an instantaneous goal function E. Very often, this function is selected as being the SE (squared error) function $E = 0.5e^2$. Different methods may be chosen, such as random update, reinforcement learning or gradient learning [Hasn95]. The first two, have the advantage that they are totally independent on the non-linear system model and thus may be applied for any synaptic or neuron models. However, they are much slower than methods based on gradient learning. In an incremental gradient learning one use the steepest descent to find a minimum of the goal function

E and thus: $\Delta w_k = -\eta \dfrac{\partial E}{\partial w_k}$. As drawbacks, this method cannot guarantee convergence towards a global minimum while the gradient may also become zero in local minima of the goal function. It guarantees also minimising the error function only when the adaptive system can be described as a composition of differentiable functions. Using this method for the general perceptron model, the update rule may be written as:

$$\Delta w_k = -\eta \frac{\partial E}{\partial w_k} = \eta \cdot e \cdot \frac{\partial synapse(o_k, w_k)}{\partial w_k} \qquad (3)$$

For the linear perceptron (multiplier-based) synapse this rule becomes:

$$\Delta w_k = -\eta \frac{\partial E}{\partial w_k} = \eta \cdot e \cdot o_k \qquad (4)$$

and was first derived in [Widr60] being cited in literature as the $\mu - LMS$ learning rule [Hasn95]. For this particular case, convergence to a global minimum is mathematically provable.

The comparative synapse model (2) is non-differentiable in origin in respect with weights. However, we may restrict weights for implementation reasons to $|w_k| \in \Omega$ and thus they will never reach the 0 value. Thus, we have to derive a learning rule which is theoretically valid except a domain $\Gamma = (-\infty,-1) \cup (-\varepsilon,\varepsilon) \cup (1,\infty)$ where we actually do not accept weights to vary. So, in addition to the learning rule we have to specify a weight restriction rule. Using fixed point implementation with saturation in both negative and positive values this restriction rule is straightforward implemented. In this circumstances, we can derive the following approximate update rule:

$$\Delta w_k = \eta \cdot e \cdot \mathrm{sign}(o_k) \cdot \mathrm{step}(|o_k| - |w_k|) \qquad (5)$$

While the surface error is now complex we cannot extend here the principle which guarantees global minimum convergence of the gradient learning for linear perceptrons. However, like in the case of back-propagation networks, numerical simulations indicate convergent behaviour when enough small adaptation rate η was used.

The main advantage of both comparative synapse model (2) and its associated learning rule (5) is that only $O(res)$ complexity is needed for implementation instead of $O(res^2)$ complexity required when multipliers have to be implemented. For the learning rule (5) we assumed the product $\eta \cdot e$ being computed with a shift register having $O(res)$ complexity. This is easy to do while accepting a power of 2 for the learning rate which is not a critical parameter. For comparison, using a typical resolution res=16 bits, this imply at least 16 times area reduction for the comparative synapse in respect with the multiplier based one. However, we expect this reduction to be greater, while the number of gates per bit in a multiplier unit is clearly larger than the number of gates per bit in a comparison unit.

In the next section we will investigate how this new model affects the overall performances and we will see that there is no major influence on both optimal structure and classification performance of the overall system, this property being problem independent.

4. Simulation results for classification problems

In order to test the influence of the novel synapse on the overall network performance we have selected an Internet available database which is extensively described in [Ele95]. The problems included in this database are classification problems where input patterns are analog as is the case in most of the signal processing tasks. It includes both artificial (synthetic) and real-world patterns which were used to test and compare the performances of many different neural architectures, extensive results of this research being also published in [Ele95]. For each problem we used separate and non-overlapping training and test sets of patterns. The qualitative performance of the network which was considered was the generalisation misclassification error *Err_gen* defined as the percent of incorrectly classified patterns in the test set. For all problems, the neuro-fuzzy network was trained according with the cross-validation principle, i.e. after each training epoch the test set was presented to the network and the learning was stopped when a minimum in *Err_gen* occurred. For each problem experiments were carried out using both multiplier-based and comparative synapse and for different *nq* values in order to find the optimal neuro-fuzzy network (i.e. which minimise *Err_gen* in respect with *nq*). The same learning rate $\eta = 0.01$ was used for all experiments. Results are presented in Fig. 3. for 3 synthetic (a. b. and c.) and 3 real-world (d. e. and f.) of the problems in the above mentioned data-base.

The "GAUSS" problem is the most difficult and it consists in 2 dimensional patterns belonging to 2 classes (100 in the train set and 1000 in the test one) with heavy overlapped distribution and non-linear separability. The best theoretical error was computed as 26.37% for this data-set. Our results, presented in Fig. 3.a. show that best performance is the same (30%) with a slight difference in the optimal structure (*nq=2* for multiplier-based and *nq=3* for comparative). The training speed was identical (80 epochs) for both synaptic models.

The "CLOUDS" problem is also an artificial one, with 2 dimensional patterns having high overlapping distributions and belonging to 2 non-linear separable classes. The best theoretical error obtained when using a Bayesian classifier was computed as 9.66%. The results presented in Fig. 3.b. indicate a slight difference (-0.6% in favour of multiplier-based) in both system performance and optimal structure while the number epochs needed to reach optimal performance was almost the same, i.e. 160 for comparative and 140 for multiplier-based.

For the "CONCENTRIC" problem, synthetic patterns belonging to 2 classes are distributed inside and outside of a well defined circular (non-linear) border. The same optimal performance (2%) was obtained (Fig. 3.c.) using both synaptic models and for slightly different optimal structures (comparative is now favoured with *nq=2* instead *nq=3* for the other model).

The "PHONEME" problem is a real world one, where each pattern is a 5 dimensional feature vector obtained from voice signals after pre-processing (see [Ele95] for details) and it may belong to one of the classes "nasal" or "oral". This is considered as a difficult problem, the best error (14.5%) reported in [Ele95] was obtained by employing a k_NN classifier. As it may be seen from Fig. 3.d. the neuro-fuzzy system performs better for the comparative synapse (*Err gen*=18%) than for the multiplier-based one but with the same optimal structure (*nq=3*) .

In the "IRIS" problem, each pattern is a four dimensional feature vector which is a numerical description of an iris flower which may belong to one of three possible classes (species). This is a common benchmark problem in pattern recognition while it has both class overlapping and non-linear border between classes. Results presented in Fig. 3.e. show better performance for comparative synapse (2% error comparing with 4% for multiplier-based) while the optimal structure is only slightly different (*nq=1* in favour of multiplier-based). For this problem, 4.5% error was reported in [Ele95] as the best result when back-propagation network was employed.

Finally, the "WINE" problem consists of 13 dimensional patterns where each pattern is a feature vector obtained from some measurable qualities of a specific wine sort which may belong to one of three classes. This problem is a nearly separable one and thus, a simple linear perceptron (using multiplier-based synapse) is able according with linear separability theory to discriminate between classes after adequate training.

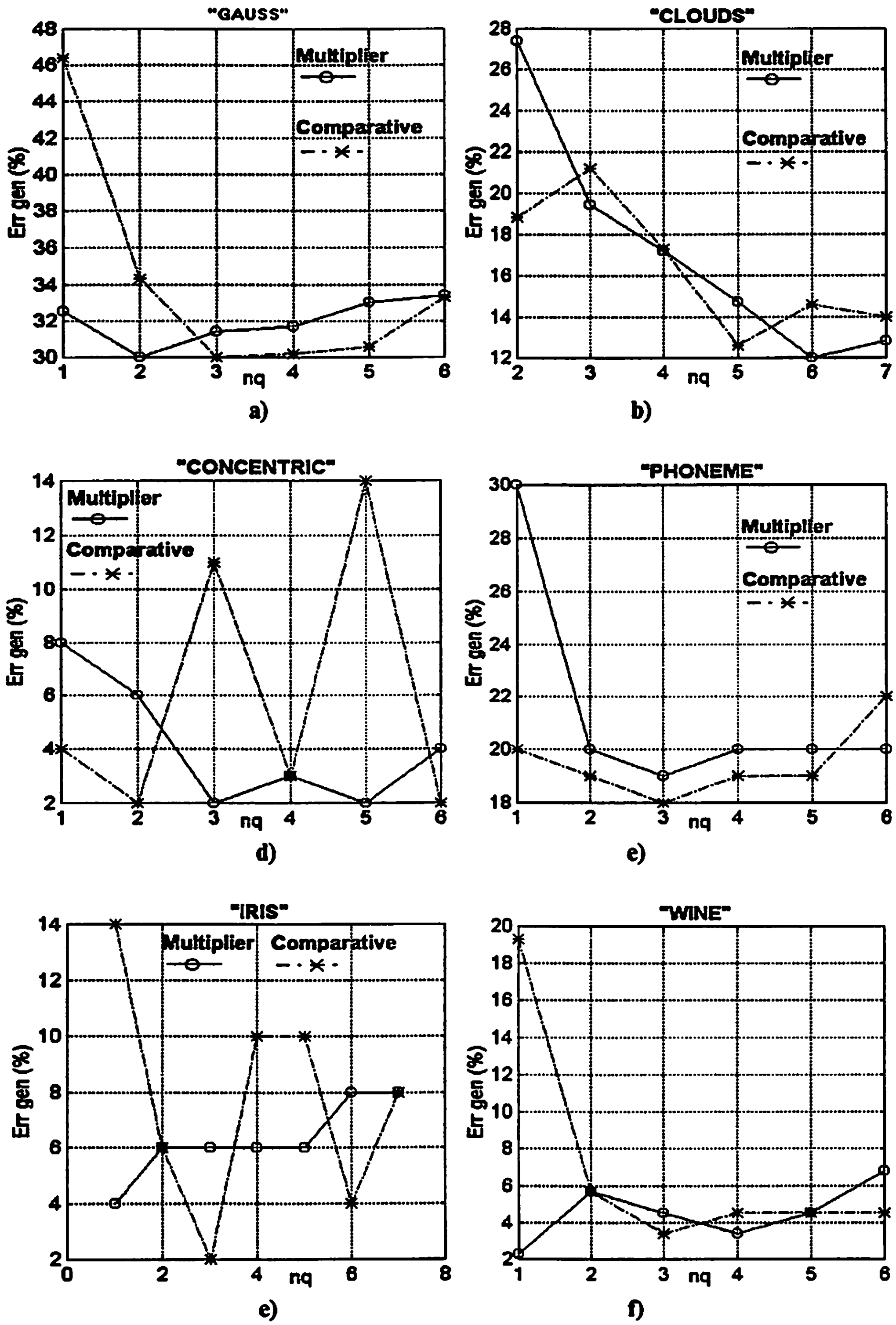

Figure 3: Simulation results for different classification problems; a) GAUSS; b) CLOUDS c) CONCENTRIC; d) PHONEME; e) IRIS; f) WINE. The misclassification error *Err_gen* is represented versus the structure parameter *nq* for both multiplier-based ('o') and comparative ('x') synapse models for each problem.

This may explain why the best performance was obtained as 2% with *nq=1* for multiplier-based synapse while the optimal structure was *nq=3* for the comparative one (with 3% error performance). In the last case, the linear separability theory is not applicable and thus the hidden layer formed by fuzzy membership functions is needed to compensate the non-linear separation performed by the comparative-based synapses perceptron in the output layer.

When patterns from other problems were used, the same features were observed, i.e. that there is a very slight dependence of both optimal structure and error performance on the synaptic model we have chosen. This observation confirm our hypothesis that adaptation is able to compensate effects of choosing different synaptic models and thus one may efficiently exploit the advantages of the comparative synapse for signal classification problems.

5. Conclusions

A new model for synaptic processing of information in neural networks was proposed based on the observation that adaptation may be used not only for learning but also to accommodate different synaptic models. From the point of view of fast and cheap VLSI or/and software implementation the "comparative" model was selected as being the most convenient. This model has also the advantage of a very simple to implement learning rule derived as an approximation of incremental gradient learning for the proposed model.

When this model was used to replace multiplier-based synapses in a fast-learning and simple to learn neuro-fuzzy system, both overall quality performance and optimal structure were maintained almost similar for a wide range of classification problems. The difference between optimal performance obtained with the neuro-fuzzy network for both synaptic models and the one reported in the extensive study [Ele95] for the same classification problem is not a consequence of the synaptic model changing but one of the particular neuro-fuzzy structure used. Thus, we may conclude that the novel model is well suited to replace the classical multiplier-based synapse in order to build fast and easy to implement neuro-fuzzy classifiers for signal processing purposes.

Numerical simulations show good convergence and no major changes in the overall system performance for classification problems. However, when tested it for signal processing tasks, where the output is not constrained to have only two values, the performances obtained with the proposed learning rule applied to the novel synapse are worst than those obtained using the standard linear perceptron. Thus, deriving a good and simple to implement learning algorithm for such kind of problems remains still an open problem.

Other applications of the comparative synapse, in adaptive neural systems where outputs are constrained to take only binary values, or in systems where other procedures than gradient learning are employed to find the synaptic values (e.g. the CNN template design or the weight space exploration [Dgr96c]) should be also considered in order to take advantage of the efficient implementation of the proposed model. For example, in order to obtain a more compact device, such synapse may replace the multiplier-based one in a CNN designed for image halftoning [Crou93] or in RBF networks used for classification.

6. Acknowledgements

This work was carried out at the University of California at Berkeley in the framework of a Fulbright research grant awarded to the first author. It was also partially supported by the Office of Naval Research under grant N00014-96-1-0753. We wish also to thank to Dr. Johan Suykens and Dr. Tao Yang for many stimulating discussions and valuable suggestions.

References

[Carp91] Carpenter, G.A, Grossberg, S. and Rosen, D.B., "Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system", in *Neural Networks, Vol. 4*, pp. 759-771, 1991

[Chua77] L. O. Chua, S. M. Kang, "Section-wise piecewise-linear functions: canonical representation, properties, and applications", in *Proceedings of the IEEE*, Vol. 65, No. 6, June 1977.

[Chua93] L. O. Chua, T. Roska, "The CNN Paradigm", in *IEEE Tr. on Circuits and Systems I*, Vol. 40, No. 3, March 1993, pp. 147- 156.

[Clou94] J. Cloutier, P. Simard, "Hardware implementation of the backpropagation without multiplication", in *Proceedings of the Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems (MicroNeuro '94)*, pp. 46-55, 1994.

[Crou93] K.R. Crounse, T. Roska, L.O. Chua, "Image halftoning with cellular neural networks", in *IEEE Trans. on Circuits and Systems - II*, Vol. 40, No. 4, April 1993.

[Cov65] Cover, T.M., "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition", in *IEEE Trans. on Electronic Computers*, EC-14, pp. 326-334, 1965.

[Dgr96a] R. Dogaru, A.T. Murgan, S. Ortmann, M. Glesner, "A modified RBF neural network for efficient current-mode VLSI implementation", in *Proceedings of the Fifth International Conference on Microelectronics for Neural Networks and Fuzzy Systems (Micro-Neuro '96)*, IEEE Computer-Press, Laussane 12-14 Febr. 1996, pp. 265-270, 1996.

[Dgr96b] Dogaru, R., Murgan, A.T, Glesner, M., Ortmann, S., "A VLSI friendly image compression algorithm based on Fuzzy-ART neural network" in *Proceedings WCNN'96* (World Congress on Neural Networks), San Diego, September 1996, pp. 1327-1330.

[Dgr96c] Dogaru, R., Murgan, A.T, Glesner, M., Ortmann, S., "Searching for robust chaos in discrete time neural networks using weight space exploration", in *Proceedings ICNN'96* (International Conference on Neural Networks, Washington, 1996), pp. 688-693.

[Dud73] Duda, R. O., and P.E. Hart, "Pattern classification and scene analysis", N.Y. John Wiley, 1973

[Ele95] ***, ELENA-95 "Enhanced Learning for Evolutive Neural Architecture", technical report ESPRIT-6891, (Benchmarks and Databases), available from Internet at *http://www.dice.ucl.ac.be/neural-nets/ELENA/ELENA.html.*

[Hasn95] M.H. Hassoun, "Fundamentals of Artificial Neural Networks", 1995, MIT Press

[Hayk94] Haykin, S. , "Neural networks: A Comprehensive Foundation", 1994, New York, Macmillan.

[Jang95] Jang, J-S. R., Sun, C-T., "Neuro-fuzzy modelling and control", in *Proceedings of IEEE*, Vol. 83, No. 3, 1995, pp. 378-406.

[Jian89] Z. Jiang, "FIR filter design and implementation with powers-of-two coefficients", *in Proc. IEEE ICASSP'89*, Glasgow, U.K., May 1989, pp. 1239-1242.

[Lin96] Lin, C-T., George Lee, C.S. , "Neural fuzzy systems - a neuro-fuzzy synergism to intelligent systems", Prentice Hall, 1996.

[Lont92] J. B. Lont, W. Guggenbühl, "Analog CMOS implementation of a multilayer perceptron with nonlinear synapses", in *IEEE Trans. on Neural Networks*, Vol. 3, No. 3, May 1992, pp. 457-465.

[Sheu95] Sheu, B. J., Choi, J., "Neural information processing and VLSI", Kluwer Academic Publishers, 1995.

[Yama92] T. Yamakawa, E. Uchino, T. Miki, H. Kusangi, "A Neo Fuzzy Neuron and its Applications to System Identification and Prediction of the System Behavior", *in Proceedings of the 2'nd International Conference on Fuzzy Logic & Neural Networks* (Iizuka, Japan, July 17-22, 1992), pp. 477-483

[Whit92] B.A. White, M. I. Elmasry, "The Digi-Neocognitron: a digital neocognitron neural network model for VLSI", in *IEEE Tr. on Neural Networks*, Vol. 3, No. 1, January 1992, pp. 73-85.

[Widr60] Widrow, B., and Hoff, M. E., Jr., "Adaptive switching circuits", in *IRE Western Electric Show and Convention Record*, part 4, pp. 96-104, 1960.