Available online at www.sciencedirect.com
ScienceDirect
Applied Soft Computing 7 (2007) 1157–1163
Applied Soft Computing
www.elsevier.com/locate/asoc

Time series prediction with single multiplicative neuron model

R.N. Yadav *, P.K. Kalra, J. John

Department of Electrical Engineering, Indian Institute of Technology, Kanpur, India

Available online 9 March 2006

Abstract

Single neuron models are typical functional replica of the biological neuron that are derived using their individual and group responses in networks. In recent past, a lot of work in this area has produced advanced neuron models for both analog and binary data patterns. Popular among these are the higher-order neurons, fuzzy neurons and other polynomial neurons. In this paper, we propose a new neuron model based on a polynomial architecture. Instead of considering all the higher-order terms, a simple aggregation function is used. The aggregation function is considered as a product of linear functions in different dimensions of the space. The functional mapping capability of the proposed neuron model is demonstrated through some well known time series prediction problems and is compared with the standard multilayer neural network.

Keywords: Multiplicative neuron model; Capacity of single neuron; Time series prediction; Mackey–Glass time series; Financial time series prediction

1. Introduction

Time series prediction and system identifications are key problems of function approximation. Various neural network architectures and learning methodologies have been used in the literature for time series prediction. In this paper, we used a single multiplicative neuron for time series prediction. An artificial neuron is a mathematical model for the biological neuron and approximates its functional capabilities. The major issue in artificial neuron models is the description of single neuron computation and interaction among the neurons with the application of the input signals. The McCulloch–Pitts model initiated the use of summing units as the neuron model, while neglecting all possible nonlinear capabilities of the single neuron and the role of dendrites in information processing in the neural system. In [1], the authors discuss the relevance of multiplicative operations, particularly in computation underlying motion perception and learning. It has been further proved that Weierstrass's theorem ensures that a network composed of one input layer and one hidden layer of product units can represent any continuous function on a finite interval [2]. The detail description of learning methodology in multiplicative neural networks has been provided by the authors in [23,24]. However, with an increasing number of terms in the higher-order expression for the polynomial neuron, it is exceedingly difficult to train a network of such neurons. We consider a simpler model for the polynomial neuron with a well-defined training procedure based on standard back-propagation. The polynomial neuron proposed in this work considers a product of linear terms in each dimension of the space. Section 2 describes the single neuron systems and the related literatures. The description of the proposed multiplicative neuron with its capacity and learning rules is provided in Section 3. Section 4 discusses the detail applications of the proposed model for time series prediction problems. Section 5 provides the concluding remarks of the paper.

2. Single neuron systems

The human nervous system is an extremely complex structure of about 10^{11} neuron units. The neurons are information processing units with three basic components—dendrites, soma and the axon and are arranged in functional constellation or assemblies according to the synaptic contacts they make with one another. The synaptic transmission involves complicated chemical and electrical processes. The sensory or chemical stimuli initiate the change in the synaptic potential. The dendrites are receptive surfaces for input signals to the neuron and generally conduct them passively to the soma. The dendrites are highly branched structures which aid in spreading

* Corresponding author. Present address: Department of Electronics and Communication Engineering, Maulana Azad National Institute of Technology, Bhopal, India. Tel.: +91 755 2670328x467.

E-mail address: rnyadav@gmail.com (R.N. Yadav).

1568-4946/$ – see front matter
doi:10.1016/j.asoc.2006.01.003

the synaptic or receptor potentials through itself to the site of impulse initiation. Axons convert these signals into train of impulses popularly known as spikes [3]. However, these are simplified models and many instances of dendritic processing have been observed which serve as a principle substrate for information processing within the neuron itself. Dendrites provide a back-propagation medium for the neuron to itself and the interfacing between axons and dendrites are remodeled throughout life explaining the formation of long-term memory [4–6]. Multiplication in neurons often occurs in dendritic trees with voltage-dependent membrane conductances [7]. In [8], the authors provide a *linear subspace* based approach to model the computation abilities of a neuron model. Linearity is believed to be sufficient for capturing the passive, or cable, properties of the dendritic membrane where synaptic inputs are currents that add. However, synaptic inputs can interact nonlinearly when the synapses are co-localized on patches of dendritic membrane with specific properties. The spatial grouping of the synapses on the dendritic tree is reflected in the computations performed at local branches. An artificial neuron model should then be capable of including this inherent nonlinearity in the mode of aggregation. Multiplication being the most basic of all nonlinearities has been a natural choice of models trying to include nonlinearity in the artificial neuron model. In [1], the authors explain the relevance of using multiplication as a computationally powerful and biologically realistic possibility of synthesizing high-dimensional Gaussian radial basis function from low dimensionality. The role of multiplication is explained in the computation underlying motion perception and learning in pairs of individual synapses to a small set of neurons. The nonlinear capability of a neuron is usually modeled through a stationary nonlinearity introduced after the aggregation. This however is not sufficient to capture the possible nonlinear associations among the inputs to the single neuron system.

Motivated by the nonlinear characteristics of the neuron and the classic Stone–Weierstrass's theorem, which states that every continuous and bounded function on a finite interval can be approximated by a sequence of polynomials of a degree arbitrarily large, a class of neuron models known as *sigma–pi* [9] and *higher-order* [10–16] neurons have been introduced and successfully used. The details of back-propagation learning algorithm with multiplicative neural networks can be found in [23,24]. These models have been proved to be more efficient as both single-units and also in networks. However, they suffer from the typical curse of dimensionality due to a combinatorial explosion of terms, demanding sparseness in representation. In this work, a new neuron model inspired from the class of higher-order neurons is proposed. The proposed model has a simpler structure without an issue of selecting the relevant monomials or the requirement of sparseness that was necessary to be imposed on the higher-order neurons to keep learning practical [17]. Thus, the model can be used in the same form in networks of similar units or in combination with the traditional McCulloch–Pitts neuron model without considering sparseness of terms. The number of parameters for the unit is twice the dimensionality of the inputs to the neurons.

We represent the single neuron as a learning machine, the problem of which is at the very core of intelligence- both natural and artificial. The capacity of the proposed learning machine using a multiplicative single neuron model is investigated using Vapnik's statistical learning theory [18]. The learning machine in our case is inspired from the model of learning from examples, where the examples are generated from a fixed but unknown probability distribution and are independent and identically distributed. VC-dimension bounds have been an important criteria to estimate the capacity of product unit networks [19] and polynomial surfaces [20]. We use the VC-dimension to evaluate the capacity of the classifier constructed by thresholding the output of the proposed neuron model. The real-counterpart of the VC-dimension and pseudo-dimension, is similarly used to estimate the function approximation ability of the single neuron model. A similar study of multiplicative neural networks is discussed in [21], where the author has provided bounds for the VC-dimension and the pseudo-dimension for a multiplicative neural network.

3. The multiplicative single neuron

The schematic diagram of a generalized single multiplicative neuron is shown in Fig. 1. Here, the operator $\mathbf{\Omega}$ is a multiplicative operation as in Eq. (1) with the weights w_i and biases b_i being the parameter $\boldsymbol{\theta}$ of the operator.

$$\mathbf{\Omega}(\mathbf{x}, \boldsymbol{\theta}) = \prod_{i=i}^{n} (w_i x_i + b_i) \tag{1}$$

Unlike the higher-order neuron, this model is more simpler in terms of its parameters and one does not need to determine the monomial structures prior to training of the neuron model. In [22], the authors have introduced a translated multiplicative neuron, which is a special case of the neuron model proposed in this work. The authors in [22] prove that such a neuron model can solve the N-bit parity problem. In terms of the definition of the higher-order neuron as given in [17], the polynomial neuron model for an *n-dimensional* input, is a 2^{n-1} higher-order neuron and every variable has *degree one* such that the neuron has *individual degree one*. It should be noted that the number of free parameters in the higher-order neuron with the given structure is same as that of the proposed neuron.

In this work, we consider a multiplicative operator and investigate the abilities of such a model to serve as a typical learning machine. In the next section we investigate the capacity of the proposed neuron model and provide a bound on

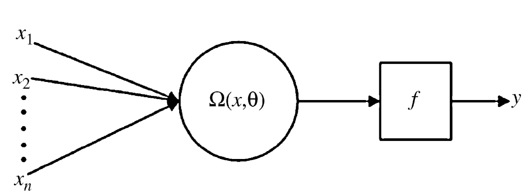

Fig. 1. A generalized single multiplicative neuron.

the VC-dimension and the pseudo-dimension of the multiplicative single neuron systems.

3.1. The capacity of the multiplicative single neuron

We begin by reviewing the classical Sauer's lemma [25–27] that forms the building principle for shattering and pseudo-shattering of a finite set S using a function space F.

Lemma. *For a function class F with VCdim(F) = d (Sauer [25]),*

$$\Pi_F(N) \le \sum_{i=0}^{d} \binom{N}{i} \; \forall \text{ positive integers } N \ge d.$$

Moreover, this bound is tight.

The Sauer's Lemma provides a method to bound the growth function using the VC-dimension and the following result follows immediately.

Corollary. *For a function class F with VCdim(F) = d, the growth function*

$$\Pi_F(N) = 2^N;\ N \le d$$

and

$$\Pi_F(N) < \left(\frac{eN}{d}\right)^d;\ N > d.$$

Definition. The degree of a polynomial in n-variables of the form

$$f(\mathbf{x}) = \sum_i x_1^{r_{1_i}} \cdots x_n^{r_{n_i}}$$

is equal to max $(r_{1_i} + \cdots + r_{n_i})$. Similarly, the degree of a curve is equal to the degree of the polynomial that defines the curve.

Theorem. (*Bezout's Theorem*) *Given two curves of degree r and s, respectively, they will meet in exactly rs points in* $\mathbf{C}^2$*—the space of complex numbers.*

3.2. Learning rule for the multiplicative single neuron

We describe an error back-propagation based learning rule for the proposed neuron model. The simplicity of the learning method makes it convenient for the model to be used in different situations unlike the higher-order neuron model, which is difficult to train and is susceptible to combinatorial explosion of terms. A simple gradient descent rule, using a norm-squared error function, is described by the following set of equations

$$u = \prod_{i=1}^{n} (w_i x_i + b_i) \tag{2}$$

$$y = g(u) = \frac{1}{1 + e^{-u}} \tag{3}$$

$$E = \frac{1}{2N} \sum_{p=1}^{N} (y^p - y^p_{\text{desired}})^2 \tag{4}$$

$$\Delta w_i = -\eta \frac{dE}{dw_i} = -\eta y(y-d)(1-y) \frac{u}{(w_i x_i + b_i)} x_i \tag{5}$$

$$\Delta b_i = -\eta \frac{dE}{db_i} = -\eta y(y-d)(1-y) \frac{u}{(w_i x_i + b_i)} x_i \tag{6}$$

$$w_i^{\text{new}} = w_i^{\text{old}} + \Delta w_i \tag{7}$$

$$b_i^{\text{new}} = b_i^{\text{old}} + \Delta b_i \tag{8}$$

where P is the number of input patterns. The learning rate η can either be adapted with epochs or can be fixed to a small number based on heuristics. This learning method is used to train the single neuron model in the next section to solve some famous benchmark problems relating to both classification and function approximation.

4. Results and discussion

Various neural network architectures and learning methodologies have been used in literatures [28–33] for solving time series prediction problems. We discuss some of the important problems that can be broadly categorized as function approximation. Detailed experiments and comparison with existing multilayer network (MLN) topology suggest that the proposed multiplicative single neuron model is a much more improved, yet simpler form of neuron unit and can be trained easily to achieve better results. In all of the problems we discuss, the dataset has been pre-processed by normalizing them between 0.1 and 0.9. In all the simulations, the results reported are the average of several runs in each case. All multilayer networks reported are trained using the standard gradient descent learning algorithm. The network topology reported is in the form of

$$n \times h_1 \times \cdots \times h_k \times o,$$

where n is the number of input nodes, h_is the number of nodes in the ith hidden layer (for $i = 1, \ldots, k$) and o is the number of output nodes. Only a single neuron model is used to solve the problems in all cases in contrast to a multilayer network.

4.1. Mackey–Glass time series

The Mackey–Glass (MG) time series [34] represents a model for white blood cell production in leukemia patients and has nonlinear oscillations which is widely used for testing the performance of neural network models. The series is a chaotic time series which makes it an universally acceptable representation of nonlinear oscillations of many physiological processes. The MG delay-difference equation is given by Eq. (9).

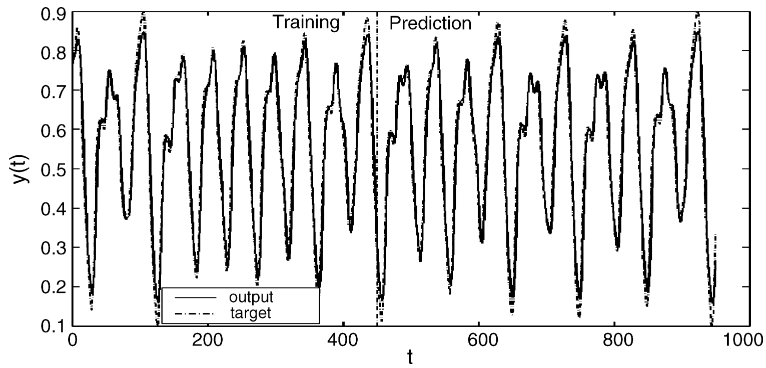

Fig. 2. Long-term prediction results for the Mackey–Glass time series dataset using the proposed neuron model.

Table 1
Comparison of performance for Mackey–Glass time series dataset between multiplicative neuron model and a standard multilayer network

Method	Structure	Parameters	Training MSE	Testing MSE	Epochs
MNM	1	8	0.00099	0.001	3000
MLN	4 × 3 × 1	19	0.0030	0.0038	5000

Table 2
Comparison of performance for the Box–Jenkins gas furnace dataset between multiplicative neuron model and a standard multilayer network

Method	Structure	Parameters	Training MSE	Testing MSE	Epochs
MNM	1	4	0.0016	0.0018	500
MLN	2 × 2 × 1	9	0.0082	0.0226	5000

$$y(t+1) = (1-b)y(t) + a\frac{y(t-\tau)}{1+y^{10}(t-\tau)} \quad (9)$$

where $a = 0.2$, $b = 0.1$ and $\tau = 17$. The time delay τ is a source of complications in the nature of the time series. The objective of the modeling is to predict the value of the time series based on four previous values. Four measurements $y(t)$, $y(t-6)$, $y(t-12)$ and $y(t-18)$ are used to predict $y(t+1)$. The training is performed on 450 samples and the model is tested on 500 time instants post training. A mean square error of 0.00099 was achieved on training the model for 3000 epochs. Fig. 2 shows the training and prediction results. In Table 1, the performance of multiplicative neuron model is compared with a multilayer network with one hidden layer having three nodes and trained using gradient descent. The performance of the multiplicative single neuron is definitely better than the multilayer network in this case, though it has fewer parameters.

4.2. Box–Jenkins gas furnace

The Box–Jenkins gas furnace dataset [35] reports the furnace input as the gas flow rate $u(t)$ and the furnace output $y(t)$ as the CO_2 concentration. In this gas furnace, air and methane were combined in order to obtain a mixture of gases which contained CO_2. We model the furnace output $y(t+1)$ as a function of the previous output $y(t)$ and input $u(t-3)$. The performance is shown in Table 2 and Fig. 3.

4.3. HCL-Infinet internet traffic

Short-term internet traffic data was supplied by HCL-Infinet (a leading Indian ISP). Weekly Internet Traffic Graph with a 30-min average is shown in Fig. 4. The solid-graph in gray shows the incoming traffic while the line-graph in black represents the outgoing traffic. All values are reported in bits per second. We propose a model for predicting the internet traffic using previous values. Three measurements $y(t)$, $y(t-1)$ and $y(t-2)$ are used to predict $y(t+1)$ for both incoming and outgoing traffic. In both the incoming and outgoing cases, 350 training samples were taken and the model was tested for prediction using 274 samples. Figs. 5 and 6 show the prediction results for incoming and

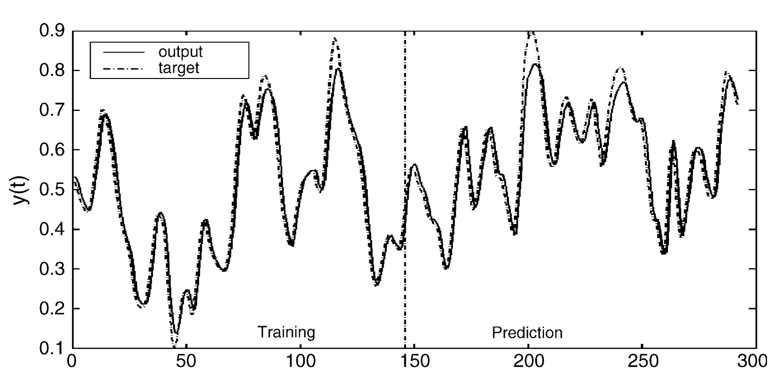

Fig. 3. Comparison of performance for Box–Jenkins time series dataset between multiplicative neuron model and a standard multilayer network.

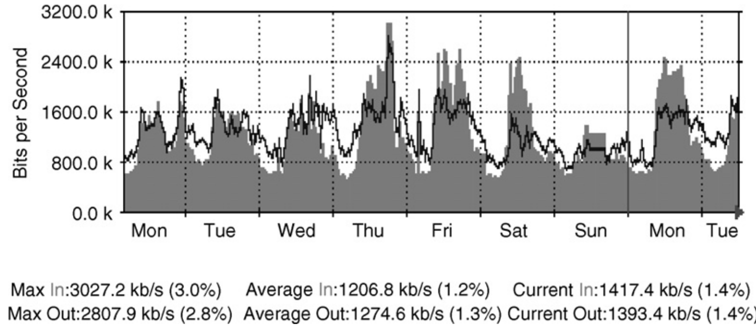

Fig. 4. Weekly graph (30 min average) of the internet traffic for the HCL-Infinet Router at Delhi, India.

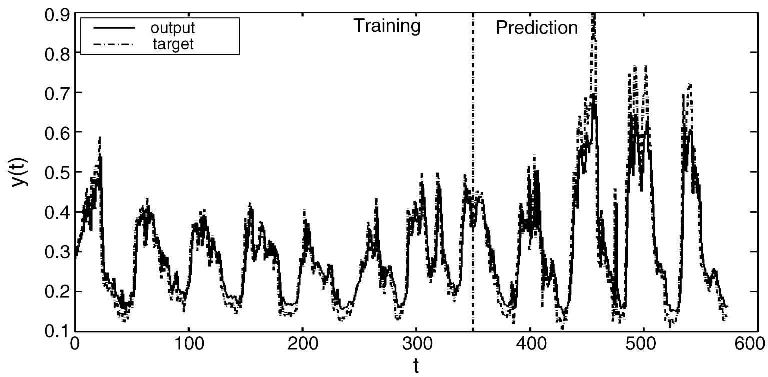

Fig. 5. Testing result on the HCL-Infinet MRTG incoming internet bandwidth usage data.

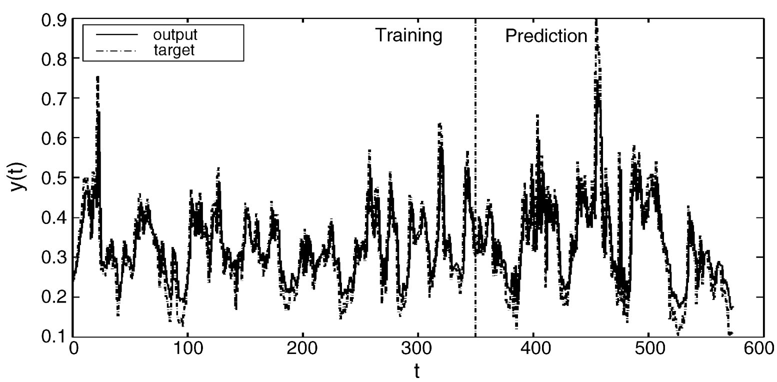

Fig. 6. Testing result on the HCL-Infinet MRTG outgoing internet bandwidth usage data.

outgoing internet traffic data, respectively. The performance is compared with multilayer network for both the cases in Tables 3 and 4 and shows better results with fewer parameters.

4.4. Financial time series

The analysis and prediction of financial time series is of primary importance and a great challenge in the field of economics. In this time series, one forecast the stock exchange index of stock market and is definitely a difficult job. We have used this time series for testing the learning and generalization capabilities of the proposed neuron model. An average of daily currency exchange rate difference between USD and INR from 2002 to 2004 (total 800 samples) were used. Five hundred samples were used for training the neuron model and 300 samples were used for testing. Fig. 7 shows the

Table 3
Comparison of performance for the incoming internet bandwidth usage of the HCL-Infinet Router data between multiplicative neuron model and a standard multilayer network method

Method	Structure	Parameters	Training MSE	Testing MSE	Epochs
MNM	1	6	0.0019	0.0071	1000
MLN	$3 \times 4 \times 1$	21	0.0021	0.0141	5000

Table 4
Comparison of performance for the outgoing internet bandwidth usage of the HCL-Infinet Router data between multiplicative neuron model and a standard multilayer network

Method	Structure	Parameters	Training MSE	Testing MSE	Epochs
MNM	1	6	0.0032	0.0066	300
MLN	$3 \times 4 \times 1$	21	0.0034	0.0134	5000

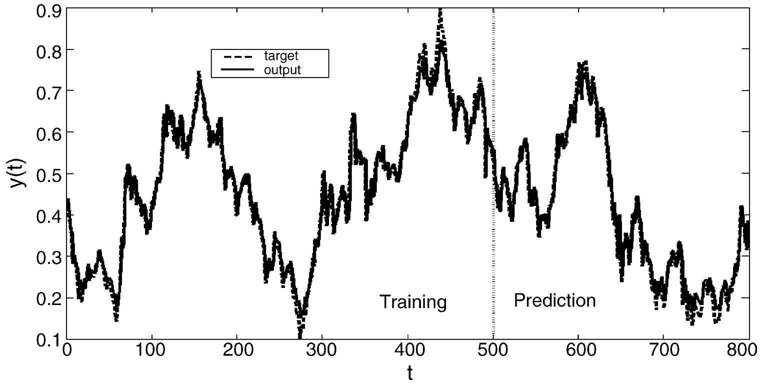

Fig. 7. Testing result on the daily currency conversion rate difference data.

Table 5
Comparison of performance between multiplicative neuron model and a standard multilayer network for daily currency conversion rate difference

Method	Structure	Parameters	Training MSE	Testing MSE	Epochs
MNM	1	8	0.00082	0.00085	2000
MLN	$4 \times 5 \times 1$	31	0.0018	0.0011	10000

target and predicted values of the proposed neuron model. The detail comparison with existing multilayer network is provided in Table 5 which clearly indicates that the proposed neuron models over performs existing multilayer neural networks.

5. Conclusion

A multiplicative neuron model based on polynomial structure has been proposed. This proposed neuron model has been used as a learning machine and its function approximation capabilities have been tested on some famous time series prediction problems. The simulation results show that the proposed neuron model exhibits significantly better performance as compared to the existing multilayer neural networks and reduces the computational complexity by more than 50%. It has also been observed that in case of proposed neuron model the computational time is significantly reduced in all the reported time series prediction problems.

Acknowledgment

We would like to thank Mr. P.V. Ramadas, HCL-Infinet, New Delhi, India for providing the Internet Traffic data.

References

[1] C. Koch, T. Poggio, in: T. McKenna, J. Davis, S.F. Zornetzer (Eds.), Multiplying with Synapses and Neurons in Single Neuron Computation, Academic Press Inc., San Diego, CA, USA, 1992.
[2] T. Poggio, On optimal nonlinear associative recall, Biol. Cybernet. 19 (1975) 201–209.
[3] F. Rieke, D. Warland, R. Steveninck, W. Bialek, Spikes: Exploring the Neural Code, MIT Press, Cambridge, MA, USA, 1997.
[4] B. Mel, Dendritic Learning, in: M.A. Arbib (Ed.), Handbook of Brain Theory and Neural Networks, MIT Press, Cambridge, MA, USA, 2003.
[5] W. Rall, Perspectives on neural model complexity, in: M.A. Arbib (Ed.), Handbook of Brain Theory and Neural Networks, MIT Press, Cambridge, MA, USA, 2003.
[6] I. Segav, M. London, in: M.A. Arbib (Ed.), Dendritic Processing in Handbook of Brain Theory and Neural Networks, MIT Press, Cambridge, MA, USA, 2003.
[7] B. Mel, Information processing in dendritic trees, Neural Comput. 6 (1994) 1031–1085.
[8] B.A. Arcas, A.L. Fairhall, W. Bialek, What can a single neuron compute, in: T. Leen, T. Dietterich, V. Tresp (Eds.), Adv. Neur. Inform. Process. 13 (2001) 75–81.
[9] T.A. Plate, Randomly connected sigma-pi neurons can form associator networks, NETCNS Network Comput. Neural Syst. 11 (2000).
[10] M. Guler, E. Sahin, A new higher-order binary-input neural unit: learning and generalizing effectively via using minimal number of monomials, in: Proceedings of Third Turkish Symposium on Artificial Intelligence and Neural Networks, 1994, pp. 51–60.
[11] M. Guler, E. Sahin, A binary-input supervised neural unit that forms input dependent higher-order synaptic correlations, in: Proceedings of World Congress on Neural Networks, 1994, pp. 730–735.
[12] M. Guler, E. Sahin, Learning higher-order correlations and symmetries using a periodic activation function, in: Proceedings of Fifth Turkish Symposium on Artificial Intelligence and Neural Networks, 1996.
[13] M. Sinha, K. Kumar, P.K. Kalra, Some new neural network architectures with improved learning schemes, Soft Comput. 4 (4) (2000) 214–223.
[14] R.N. Yadav, V. Singh, P.K. Kalra, Classification using single neuron, in: Proceedings of First IEEE International Conference on Industrial Informatics, INDIN'03, Banff, Canada, August 21–24, (2003), pp. 124–129.
[15] R.N. Yadav, N. Kumar, P.K. Kalra, J. Joseph, Multilayer neural networks using generalized mean neuron model, in: Proceedings of the IEEE International Symposium on Communication and Information Technologies, Sapporo, Japan, October 26–29, (2004), pp. 93–97.
[16] D.K. Chaturvedi, M. Mohan, R.K. Singh, P.K. Kalra, Improved generalized neuron model for short-term load forecasting, Soft Comput. 8 (5) (2004) 370–379.
[17] M. Schmitt, VC dimension bounds for higher-order neurons, in: Proceedings of the Ninth International Conference on Artificial Neural Networks, vol. 2, 1999, pp. 563–568.
[18] V. Vapnik, Statistical Learning Theory, John Wiley and Sons, Inc., New York, 1998.
[19] M. Schmitt, VC dimension bounds for product unit networks, in: Proceedings of the International Joint Conference on Neural Networks, vol. 4, 2000, pp. 165–170.
[20] M. Anthony, Classification by polynomial surfaces, Discrete Appl. Maths 61 (1995) 91–103.
[21] M. Schmitt, On the complexity of computing and learning with multiplicative neural networks, Neural Comput. 14 (2001) 241–301.

[22] E.M. Iyoda, H. Nobuhara, K. Hirota, A solution for the *N*-bit parity problem using a single multiplicative neuron, Neural Process. Lett. 18 (2003) 233–238.
[23] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error back propagations, in: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, MIT Press, 1986.
[24] D. Li, K. Hirasawa, J. Hu, J. Murata, Multiplication units in feedforward neural networks and its training, in: Proceedings of Ninth International Conference on Neural Information Processing, ICONIP'02, Singapore, 2002.
[25] N. Sauer, On the density of family of sets, J. Comb. Theory (A) 13 (1972) 145–147.
[26] D. Haussler, P.M. Long, A generalization of Sauer's Lemma, J. Comb. Theory (A) 71 (1995) 219–240.
[27] P. Auer, P.M. Long, W. Maass, G.J. Woeginger, On the complexity of function learning, Mac. Learn. 18 (1995) 187–236.
[28] R. Drossu, Z. Obradovic, Rapid design of neural networks for time series prediction, IEEE Comput. Sci. Eng. 3 (2) (1996) 78–89.
[29] A.B. Geva, ScaleNet-multiscale neural network architecture for time series prediction, IEEE Trans. Neural Networks 9 (6) (1998) 1471–1482.
[30] H. Wakuya, J.M. Zurada, Bidirectional computing architecture for time series prediction, Neural Networks 14 (9) (2001) 1307–1321.
[31] L. Manevitz, A. Birtar, D. Giroli, Neural network time series forecasting of finite element mesh adaptation, Neurocomputing 63 (2005) 447–463.
[32] T.Y. Kim, K.J. Oh, C. Kim, J.D. Do, Artificial neural networks for non-stationary time series, Neurocomputing 61 (2004) 439–447.
[33] S.G. Kong, Time series prediction with evolvable block based neural networks, in: IEEE International Joint Conference on Neural Networks, July 25–29, vol. 2, (2004), pp. 1579–1583.
[34] M. Mackey, L. Glass, Oscillation and chaos in physiological control systems, Science 197 (1997) 287–289.
[35] G.E.P. Box, G.M. Jenkins, G.C. Reinse, Time Series Analysis: Forecasting and Control, Prentice Hall, Englewood Cliffs, NJ, 1994.