15th Triennial World Congress, Barcelona, Spain

A RECURRENT FUZZY NEURON FOR ON LINE MODELLING OF NONLINEAR SYSTEMS

Edmary Altamiranda [*,1] **Eliezer Colina** [**,2]

[*] *Gerencia Técnica, Automatización, PDVSA PEQUIVEN. El Tablazo, Zulia - Venezuela*
[**] *Departamento de Sistemas de Control Escuela de Ingeniería de Sistemas. Universidad de Los Andes, Mérida – Venezuela*

Abstract: This paper presents a recurrent fuzzy neuron (RFN) which facilitates nonlinear mapping from an input space to an output space. The synaptic junctions are characterized by a set of IF-THEN rules and recurrent characteristics provide dynamic properties to the neuron, allowing its application to on line modelling for a variety of nonlinear systems. The effectiveness of this neuron to synthesize complex nonlinear models, is illustrated by simulation results related to on line prediction of chaotic behavior and modelling of time varying nonlinear systems.

Keywords: Fuzzy Systems, Neural Dynamics, Nonlinear Models, Dynamic Modelling

1. INTRODUCTION

Artificial Neural Networks, for modelling of complex dynamic systems have the following advantages: Distributed information processing capabilities, an inherent potential for parallel computation, nonlinear behavior and learning properties. Conventional feedforward multilayered neural networks facilitates nonlinear mapping from an input space to an output space. The network system can be established only by training and its ability for complex mapping may increase with the number of layers and neural elements in each layer. Most of times it is difficult to establish in advance, the number of neural elements necessary and sufficient to achieve an adequate mapping. Since all the initial weights are assigned randomly and the error weight space might have local minima, the learning error may be significant even after a long learning period. In order to deal with such difficulties a fuzzy neuron structure is proposed by (Yamakawa, 1994). The concept associated to this structure modify the conventional neuron model which possesses constant synaptic weights followed by a nonlinear activation function. Contrary to that, the fuzzy neuron has nonlinear synapses characterized by sets of fuzzy IF-THEN rules with singleton weights in consequent. When it comes to system's modelling, a dynamic neural structure, which contains feedback terms may provide more advantages than a purely feedforward neural structure. Supervised learning algorithms for dynamic neural structures have been developed for identification, control and optimization of dynamic systems (Pineda, 1987), (Williams and Zisper, 1989), (Narendra and Parthasarathy, 1990) and (Parlos *et al.*, 1991). For some problems, a small feedback system is equivalent to a large and possible infinite feed-

[1] Contact main author at: Puertos de Altagracia PDVSA PEQUIVEN, Automatización, Gerencia Técnica. e-mail adresses: ALTAMIRANDE@PDVSA.COM, edmary_al@telcel.net.ve
[2] ecolina@ing.ula.ve

forward system (Hush and Horn, 1993). In order to obtain the nonlinear mapping capabilities of the fuzzy neuron presented by (Yamakawa, 1994) and the features of feedback systems, a fuzzy neuron structure which incorporates recurrent connections is proposed. Such recurrent connections provide dynamic characteristic to the fuzzy neuron behavior, making the proposed structure, a good candidate for on line modelling of a variety of nonlinear dynamic systems. The proposed RFN potentials are illustrated by simulation results related to on line prediction of chaotic behavior and, modelling of time varying nonlinear systems.

2. A FUZZY NEURON MODEL

Provided a quadratic learning error, a single ordinary neuron model guarantees to find a global minimum. The neuron output and the learning error are represented by

$$y = f(\sum_{i=1}^{p} w_i x_i) \tag{1}$$

$$E = \frac{1}{2}\sum_{k=1}^{p}(y_k - y_k^d)^2 \tag{2}$$

respectively, where x_i, is the input signal to the *i-th* synapse, w_i the corresponding input weight, y_k the neuron output for *k-th* pattern, y_k^d the *k-th* training pattern and p the number of patterns. Note from 1 and 2, that the error-weight space exhibits a parabolic function. Multilayered neural networks which possesses significant generalization capabilities, have to be constructed with too many neural elements where a large number of parameters are embedded, therefore, the error surface is not parabolic with respect to the weights causing the appearance of local minima (Zurada, 1992) and (Hetch, 1990). In order to avoid the local minima problem, a fuzzy neuron, where many parameters can be embedded and which accomplishes generalization by itself, has been proposed in (Yamakawa, 1994).

2.1 Structure of the Fuzzy Neuron

The structure of the fuzzy neuron presented in (Yamakawa, 1994) is shown in Figure 1

The characteristics of each synapse are represented by a nonlinear function f_i and the soma does not exhibit a sigmoidal function at all. Aggregation of synaptic signals is achieved by an algebraic sum. Thus the output of this fuzzy neuron can be represented by the following equation:

$$y = \sum_{i=1}^{m}(f_i(x_i)) \tag{3}$$

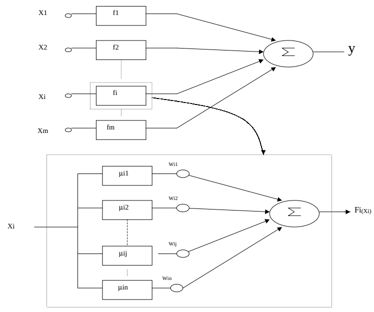

Fig. 1. Structure of Fuzzy Neuron

The input space for x_i is divided into several fuzzy segments which are characterized by membership function $\mu_{i1}, \mu_{i2}, ..., \mu_{ij}, ..., \mu_{in}$ within the range between x_{min} and x_{max}, as shown in Figure 2

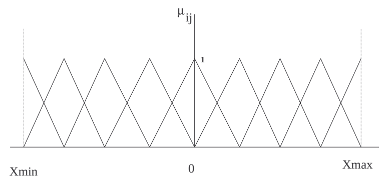

Fig. 2. Membership Function for Nonlinear Synapsis of Fuzzy Neuron

The nonlinearity $f_i(x_i)$ is determined by fuzzy IF-THEN rules and the k-th Fuzzy IF-THEN Rule in i-th synapse output is obtained by fuzzy inference with defuzzification,represented by:

If Input signal x_i is included in the fuzzy segment μ_{ik}, then the synapse output is w_{ik}

Compatibilities of the input signal x_i with the antecedents.2 of these rules are obtained from the membership function to be $\mu_{ik}(x_i)$ and $\mu_{i,k+1}(x_i)$ at which the constants w_{ik} and $w_{i,k+1}$ in consequents should be adopted. The synapse output is inferred by one or two rules activated by the input signal x_i and represented by one or a couple of singletons truncated by $\mu_{ik}(x_i)$ and $\mu_{i,k+1}(x_i)$ as shown in Figure 3

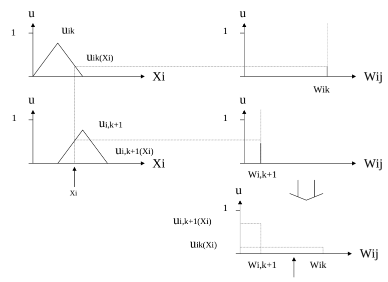

Fig. 3. Synapse Output Inference

A deterministic value of the synaptic output is obtained by the defuzzification, so called *center-of-gravity method*, defined by:

$$f_i(x_i) = \frac{\sum_{j=1}^{n} \mu_{ij}(x_i) w_{ij}}{\sum_{j=1}^{n} \mu_{ij}(x_i)} = \frac{\mu_{ik}(x_i) w_{ik} + \mu_{i,k+1}(x_i) w_{i,k+1}}{\mu_{ik} + \mu_{i,k+1}} \quad (4)$$

Since the membership functions are complementary,(the summation of two neighboring membership function is always unity), an input signal x_i activates only one or two rules simultaneously for each synapse. Thus, the output of the fuzzy neuron may be rewritten as:

$$f_i(x_i) = \mu_{ik}(x_i) w_{ik} + \mu_{i,k+1}(x_i) w_{i,k+1} \quad (5)$$

This equation implies that only activated branches are effective for learning therefore, only one or two weights corresponding to the activated branches are adjusted at the time.

3. RECURRENT FUZZY NEURON MODEL

In this paper, a modified fuzzy neuron is proposed, which presents the capabilities of nonlinear mapping of the fuzzy neuron presented in (Yamakawa, 1994) and the advantages that provide feedback systems. The proposed structure incorporates recurrent connections which provide dynamic characteristic. This properties make the neuron a good candidate for on line modelling of nonlinear MISO and SISO systems.

3.1 Structure of the RFN for Nonlinear Modelling of MISO and SISO Systems

The structure of the RFN is shown in Figure 4.

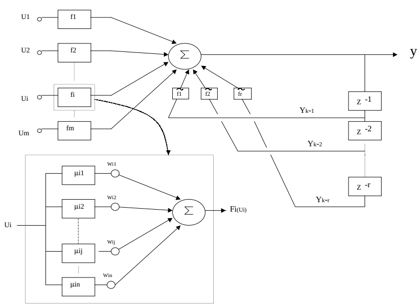

Fig. 4. Structure of RFN

The feedforward and recurrent connection f_i and $\tilde{f}_r$ respectively, possesses nonlinear synaptic weights which are determined by fuzzy IF-THEN rules. The synapse output is obtained by fuzzy inference with defuzzification, therefore, the output of this RFN may be represented by:

$$y(k) = \sum_{i=1}^{m} f_i(u_i(k)) + \sum_{r=1}^{n} \tilde{f}_r(y(k-r)) \quad (6)$$

where $u_i(k)$, $i = 1..m$ correspond to the system inputs at the time k, $y(k-r)$ $r = 1..n$, are recurrent terms associated to the neuron output at the time $k-r$ and n is the estimated system order. Dynamic inputs delays can be embedded in the RFN structure but it is important to consider an estimated of the system relative degree, to define the appropriate number of recurrent terms. In order to explain this fact in a better manner, consider the following SISO system

$$\begin{aligned} y^d(k) = F(y^d(k-1), .., y^d(k-n)) + \\ G(u(k), u(k-1), ..., u(k-m)) \end{aligned} \quad (7)$$

where y^d is the SISO system output and u is the SISO system input

This system may be modelled by an RFN by defining the neuron inputs as the delays associated to the unique system input

$$y(k) = \sum_{i=0}^{m} f_i(u(k-i)) + \sum_{r=1}^{n} \tilde{f}_r(y(k-r)) \quad (8)$$

Assuming that the estimated relative degree is at least zero, which implies causality, the number of recurrent terms and inputs delays must be defined such that $n \geq m$.

An extension of this representation for MISO systems may be defined in the following way

$$\begin{aligned} y^d(k) = F(y^d(k-1), ..., y^d(k-n)) + \\ G_1(u_1(k), u_1(k-1), ..., u_1(k-m_1)) + \\ .... + G_m(u_m(k), u_m(k-1), ..., u_m(k-m_m)) \end{aligned} \quad (9)$$

where y^d is the MISO system output, m is the number of inputs $u_1, ..., u_m$ and $m_1, ..., m_m$ correspond to the input delays respectively.

The before mentioned system may be modelled by a RFN as:

$$\begin{aligned} y(k) = \sum_{i_1=0}^{m_1} f_{i_1}^1(u_1(k-i_1)) + ... + \\ \sum_{i_m=0}^{m_m} f_{i_m}^m(u_m(k-i_m)) + \sum_{r=1}^{n} \tilde{f}_r(y(k-r)) \end{aligned} \quad (10)$$

where $f_{i_m}^m$ correspond to the synapse associated to the $i_m - th$ input delay of the $m - th$ input.

The number of recurrent terms and input delays must be defined such that $n \geq max(m_1, ..., m_m)$ in order to guarantee causality with respect to every inputs.

Each synapse output is inferred by one or two rules activated by an input or a recurrent signal. Similar to equation 5, two expressions may be obtained for calculating the deterministic value for the synaptic connections of the RFN. Without loss of generality the feedforward connections are given by:

$$f_i(u_i(k)) = \mu_{i,j}(u_i(k))w_{i,j}(k) + \mu_{i,j+1}(u_i(k))w_{i,j+1}(k) \quad (11)$$

and the recurrent connections may be expressed as

$$\tilde{f}_r(y(k-r)) = \tilde{\mu}_{r,j}(y(k-r))\tilde{w}_{r,j}(k) + \tilde{\mu}_{r,j+1}(y(k-r))\tilde{w}_{r,j+1}(k) \quad (12)$$

For nonlinear systems, output behavior may be characterized in several zones. The number of membership function may define the number of zones associated to the output nonlinear behavior respect to a specific input variable. Generally, the number of regions or zones that characterize a complex nonlinear system it is not completely known, therefore, it is convenient, to start the RFN design using a reduced number of membership functions and increase progressively this number for each input variable, in order to improve convergence. A reduced number of membership functions may affect the algorithm convergence but too many memberships functions make difficult computational processing for implementation purposes.

It is important to mention that constraint propositions defined above, for m and n are valid for input/output systems. Time series may be modelled using RFN structures, for this case the constraint is $n \geq 1$ to guarantee neuron dynamic properties. m is normally associated to real output delays.

3.2 Learning Algorithm for the Recurrent Fuzzy Neuron

The general learning algorithm is defined in terms of a steepest descent method, where the change of weights is achieved for a set of input patterns p. The error index is given by the average squared error for p patterns in the following way:

$$E(k) = \frac{1}{2P}\sum_{q=1}^{p}(y_q(k) - y_q^d(k))^2 = \frac{1}{2P}\sum_{q=1}^{p}e_q^2(k) \quad (13)$$

where $y_q(k)$ is the RFN output, $y_q^d(k)$ is the desired output, corresponding to the pattern q at time k and $e_q(k)$ is a learning error between the RFN and the desired output at time k.

During the learning sessions, the updating rule for the weights is given by

$$w_{i,j}(k+1) = w_{i,j}(k) + \Delta w_{i,j}(k) \qquad \Delta w_{i,j}(k) = -\alpha\frac{\partial E(k)}{\partial w_{i,j}(k)} \quad (14)$$

The derivatives of the error index E(k) with respect to the weights of the RFN are as follows

For the feedforward connections:

$$\frac{\partial E(k)}{\partial w_{i,j}(k)} = \sum_{q=1}^{p} e_q(k)\frac{\partial y_q(k)}{\partial w_{i,j}(k)} \quad (15)$$

The adjustment for weights $w_{i,j}$ in time k is given by

$$\Delta w_{i,j}(k) = -\alpha\sum_{q=1}^{p} e_q(k)[\frac{\partial y_q(k)}{\partial f_i(u_{q,i}(k))}\frac{\partial f_i(u_{q,i}(k))}{\partial w_{i,j}(k)}] \quad (16)$$

where $u_{q,i}$ corresponds to the $q-th$ pattern of the $i-th$ RFN feedforward input

$$\Delta w_{i,j}(k) = -\alpha\sum_{q=1}^{p}(y_q(k) - y_q^d(k))\mu_{i,j}(u_{q,i}(k)) \quad (17)$$

For the recurrent connections:

$$\frac{\partial E(k)}{\partial \tilde{w}_{i,j}(k)} = \sum_{q=1}^{p} e_q(k)[\frac{\partial y_q(k)}{\partial \tilde{w}_{i,j}(k)} + \frac{\partial y_q(k)}{\partial y_q(k-r)}\frac{\partial y_q(k-r)}{\partial \tilde{w}_{r,j}(k)}] \quad (18)$$

The adjustment for weights $\tilde{w}_{r,j}$ in time k is given by

$$\Delta\tilde{w}_{r,j}(k) = \Delta_1\tilde{w}_{r,j} + \Delta_2\tilde{w}_{r,j} \quad (19)$$

where

$$\Delta_1\tilde{w}_{r,j} = -\alpha\sum_{q=1}^{p} e_q(k)[\frac{\partial y_q(k)}{\partial \tilde{w}_{r,j}(k)}] \quad (20)$$

$$\Delta_2\tilde{w}_{r,j} = -\alpha\sum_{q=1}^{p} e_q(k)[\frac{\partial y_q(k)}{\partial y_q(k-r)}\frac{\partial y_q(k-r)}{\partial \tilde{w}_{r,j}(k)}] \quad (21)$$

It is not difficult to understand that the first term $\Delta_1\tilde{w}_{r,j}$, corresponds to a static partial derivative obtained from applying conventional static steepest descent method to the error index, and the second term $\Delta_2\tilde{w}_{r,j}$, is due to the dynamic behavior of the RFN.

Calculation of $\Delta_1\tilde{w}_{r,j}$

$$\Delta_1\tilde{w}_{r,j} = -\alpha\sum_{q=1}^{p} e_q(k)[\frac{\partial y_q(k)}{\partial \tilde{f}_i(y_q(k-r))}\frac{\partial \tilde{f}_i(y_q(k-r))}{\partial \tilde{w}_{r,j}(k)}] \quad (22)$$

$$\Delta_1\tilde{w}_{r,j} = -\alpha\sum_{q=1}^{p} e_q(k)[\mu_{r,j}(y_q(k-r)] \quad (23)$$

Calculation of $\Delta_2 \tilde{w}_{r,j}$

$$\Delta_2 \tilde{w}_{r,j} = -\alpha \sum_{q=1}^{p} e_q(k)[\frac{\partial y_q(k)}{\partial y_q(k-r)} z_q^r(k)] \quad (24)$$

where

$$\frac{\partial y_q(k)}{\partial y_q(k-r)} = \tilde{w}_{r,j}(k)\dot{\tilde{\mu}}_{r,j}(y_q(k-r)) + \tilde{w}_{r,j+1}(k)\dot{\tilde{\mu}}_{r,j+1}(y_q(k-r)) \quad (25)$$

Let z_q^r be defined as

$$z_q^r(k) = \frac{\partial y_q(k-r)}{\partial \tilde{w}_{r,j}(k)} \quad (26)$$

In order to provide dynamic characteristics to the updating rule for recurrent connections, the term 26, is determined through the successive application of the chain rule. Without loss of generality, only first order terms are considered, therefore $z_q^r(k)$ may be represented by the following first order time-varying linear system:

$$z_q^r(k) = \frac{\partial y_q(k-r)}{\partial y_q(k-r-1)} z_q^r(k-1) + \frac{y_q(k-r)}{\partial \tilde{w}_{r,j}(k)} \quad (27)$$

$$z_q^r(k) = [\tilde{w}_{r,j}(k-1)\dot{\tilde{\mu}}_{r,j}(y_q(k-r-1)) + \tilde{w}_{r,j+1}(k-1)\dot{\tilde{\mu}}_{r,j+1}(y_q(k-r-1))] z_q^r(k-1) + \tilde{\mu}_{r,j}(y_q(k-r)) \quad (28)$$

This definition for z_q^r, is a modification of a dynamic updating rule for recurrent neural networks presented in (Jin *et al.*, 1994).

Finally $\Delta \tilde{w}_{r,j}$ may be expressed as

$$\Delta \tilde{w}_{r,j} = -\alpha \sum_{q=1}^{p} e_q(k)[\mu_{r,j}(y_q(k-r)) + \frac{\partial y_q(k)}{\partial y_q(k-r)} z_q^r(k)] \quad (29)$$

The error- weight space for the proposed RFN exhibits a parabolic function and has no local mnima, because only one neural element is embedded. Additionally, RFN's dynamic characteristics given by recurrent terms, provide higher convergence speed than purely feedforward neural structures.

4. ON LINE MODELLING OF NONLINEAR SYSTEMS USING RFN

In this section, it will be illustrated the potentials of the proposed RFN for modelling of complex dynamic systems, through its application to on line prediction of chaotic behavior and modelling of time variant nonlinear systems.

4.1 Case of Simulation 1:Prediction of Chaotic Behavior

Consider the chaotic time series generated by the following recurrent formula:

$$x(k+1) = \frac{5x(k)}{1+x^2(k)} - 0.5 * x(k) - 0.5x(k-1) + 0.5x(k-2) \quad (30)$$

A RFN based structure used to model this system is defined as:

$$y(k) = \sum_{i=1}^{4} f_i(x(k-i)) + \tilde{f}_1(y(k-1)) \quad (31)$$

The learning rate used, $\alpha = 0.22$, was obtained empirically and the initial conditions for the chaotic series were: $x(0) = 0.7333$, $x(1) = 0.234$ and $x(2) = 0.973$. The number of patterns used for each time instant $p = 2$ and the membership functions for all (RFN) inputs are shown in Figure 5.

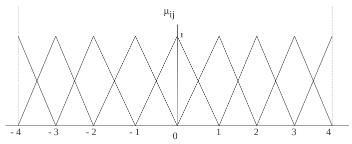

Fig. 5. Membership Function for RFN Inputs and Recurrents Terms. Chaotic Series

Figure 6, illustrates the effectiveness of the RFN to predict on line the chaotic behavior of series represented in equation 30

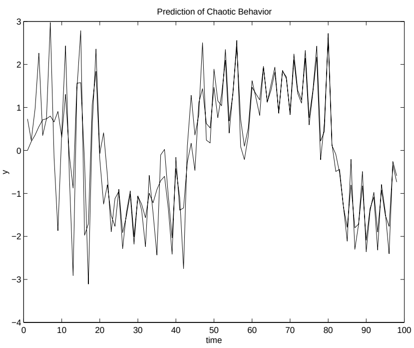

Fig. 6. On line Prediction of Chaotic Behavior by (RFN)

4.2 Case of Simulation 2:Modelling of a Time Varying Nonlinear System

Consider the following time varying nonlinear SISO system

$$\begin{aligned} x_1(k+1) &= u(k) \\ x_2(k+1) &= x_3(k) \\ x_3(k+1) &= x_4(k) \\ x_4(k+1) &= g(x) + \Delta g(k) + u(k) \\ y^d(k) &= x_4(k) \\ u(k) &= sin(k) \end{aligned} \quad (32)$$

where

$$g(k) = \frac{x_1(k)x_2(k)x_3(k)}{1 + x_2^2(k) + x_3^2(k)} + 0.5x_4(k) \quad (33)$$

The varying term structure is given by

$$\Delta g(k) = 0.5 * x_1(k)x_2(k)x_3(k) \quad (34)$$

This term is defined only on the interval $[50, 150]$, else $\Delta g(k)$ is equal to zero

The (RFN) based structure used to model this SISO system is defined as

$$y(k) = \sum_{i=1}^{2} f_i(u(k-i)) + \sum_{r=1}^{3} \tilde{f}_r(y(k-r)) (35)$$

The learning rate used, $\alpha = 0.17$, was obtained empirically. The number of patterns used for each time instant $p = 1$.

The membership functions for RFN inputs and recurrent terms are shown in Figures 7 and 8 respectively

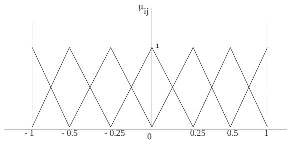

Fig. 7. Membership Function for RFN Inputs. Time-Varying Nonlinear System

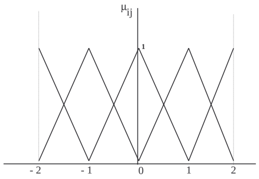

Fig. 8. Membership Function for RFN Recurrent Terms. Time-Varying Nonlinear System

Figure 9 illustrate the potentials of the RFN to model on line the time varying nonlinear SISO system

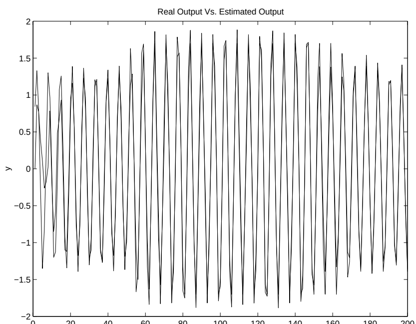

Fig. 9. On line Modelling of time varying nonlinear SISO System by (RFN)

5. CONCLUSION

In this paper a new recurrent fuzzy neuron is proposed. The RFN incorporates recurrent terms which provide dynamic characteristics to the structure, making it a good candidate for on line modelling of MISO and SISO systems. The effectiveness of this neuron to represent complex nonlinear processes, is illustrated by computer simulation results related to on line prediction of chaotic behavior and modelling of a time varying nonlinear system.

REFERENCES

Hetch, R. (1990). *Neurocomputing*. Addison-Wesley Publishing Company.

Hush, D.R. and B.G. Horn (1993). Progr in supervised neural networks. *IEEE Signal Procesing Magazine* (1), 8–39.

Jin, L., P. Nikiforuk and M. Gupta (1994). Dynamic recurrent neural networks for control of unknown nonlinear systems. *Journal of Dynamic Systems Measurement and Control* **116**, 567–576.

Narendra, K.S and K. Parthasarathy (1990). Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Networks* **2**(2), 4–27.

Parlos, A., A. Atiya and K. Chong (1991). Recurrent multilayer perceptron for nonlinear systems identification. *Proc. IJCNN* **II**, 537–540.

Pineda, F.J. (1987). Generalization of back-propagation to recurrent neural networks. *Physical Review Letters* **59**(19), 2229–2232.

Williams, R. and D. Zisper (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation* **2**, 4–27.

Yamakawa, T. (1994). *A Neo Fuzzy Neuron and its Application to System Identification and Prediction of Chaotic Behavior*. pp. 383–405. IEEE Press. New York.

Zurada, J.M. (1992). *Introduction to Artificial Neural Systems*. West Publishing Company.