First Joint Congress on Fuzzy and Intelligent Systems
Ferdowsi University of Mashhad, Iran
29-31 Aug 2007

Intelligent Systems Scientific Society of Iran

A Novel Unsupervised Neuro-Fuzzy System Applied to Circuit Analysis

Hadi Sadoghi Yazdi
Seyed Ebrahim Hosseini
Engineering Department, Tarbiat Moallem University of Sabzevar, Sabzevar, Iran

E-mail: {sadoghi, ehosseini}@sttu.ac.ir

Abstract: In this paper, for the first time, unsupervised neuro-fuzzy system is presented and is applied in circuit analysis. Usually Neuro-fuzzy systems have a learning phase in which the system is trained with input data. But if the training set is unavailable, conventional procedures encounter serious problem. Due to unsupervised character, no learning data is needed. To investigate the method, linear circuit is analyzed. Results are compared with the exact solution.

Keywords: Unsupervised neuro-fuzzy system, Circuit analysis.

1. Introduction

Artificial neural networks (ANN), as intelligent computational tools have been used for modeling and optimization of analog circuit design [1]. For example optimization of high-speed very large scale integration (VLSI) interconnects [2], global modeling [3, 4] and circuit synthesis [5].

Numerous problems in science and engineering can be converted to a set of differential equations. Basic numerical methods can be used to solve differential equations such as the finite difference method, the finite element method, the finite volume method and the boundary element method.

Neural networks (NNs) are used to approximate stochastically unknown functions and relations. This approximation is a kind of implicit model of the unknown dependencies. In contrast, differential equations are used to model explicitly all relations. Solving ordinary and partial differential equations can be learned to a good degree by an artificial neural network. The basic ideas were presented by Lagaris, Likas, and Fotiadis [6]. Let the differential equation to be solved be given by:

$$G(x,\psi(x),\nabla\psi(x),\nabla^2\psi(x),...)=0 \quad x\in\overline{D}\subseteq R^n, \qquad (1)$$

Where $\psi(x)$ denotes the solution, G is the functional defining the structure of the differential equation, and ∇ is some differential operator. The basic idea, called collocation method, is to discretize the domain $\overline{D}$ over a finite set of points D. Thus (1) becomes a system of equations. Suppose an approximation of the solution ψ(x) is given by the trail solution ψ_t(x). As a measure for the degree of fulfillment of the original differential equation (1), an error function similar to the mean squared error is defined:

$$E=\frac{1}{|D|}\sum_{x_i\in D}\left[G(x_i,\psi_t,\nabla\psi_t,\nabla^2\psi_t,...)\right]^2. \qquad (2)$$

Therefore, finding an approximation of the solution of (1) is equal to finding a function that minimizes the error E. It is well known that a multilayer feed forward neural network is a universal approximator [7], therefore the trail solution ψ_t(x) can be represented by such an artificial neural network. In case of a given network architecture the problem is reduced to finding a configuration of weights that minimizes (2). As E is differentiable with respect to the weights for most differential equations, efficient gradient-based learning algorithms for artificial neural networks can be employed for minimizing (2).

In [8] Hüsken, and Goerick have been focused on the choice of a set of initial weights using an evolutionary solving a differential equation with variable boundary conditions. Smaoui, Al-Enezi [9] analyzed dynamics of two nonlinear partial differential equations known as the Kuramoto–Sivashinsky (K-S) equation and the two-dimensional Navier-Stokes (N-S) equations using Karhunen-Loeve (K-L) decomposition and artificial neural networks. In [10] Brause used differential equations for modeling of biochemical pathways and these equations were solved using neural networks. In [11] He, Reif, and Unbehauen used a feed forward neural network with the extended back propagation algorithm to solve a class of first-order partial differential equations for input-to-state linearizable or approximate linearizable systems.

Also Manevitz, Bitar, and Givoli [12] presented basic learning algorithms and the neural network model to the problem of mesh adaptation for the finite-element method to solve time-dependent partial differential equations. Time series prediction via the neural network methodology was used to predict the areas of ‘‘interest’’ in order to obtain an effective mesh refinement at the appropriate times. Leephakpreeda [13] presented fuzzy linguistic model in neural network to solve differential equations and applied as universal approximators for any nonlinear continuous functions.

In [14] Malek, and Beidokhti presented a hybrid method based on optimization techniques and neural networks methods for the solution of high order ordinary differential equations. They proposed a new solution method for the approximated solution of high order ordinary differential equations using innovative mathematical tools and neural-like systems of computation. Hybrid method could result in improved numerical methods for solving initial/boundary value problems, without using pre assigned discretization points.

In another work, Mai-Duy, and Tran-Cong [15] presented mesh-free procedures for solving linear ordinary and elliptic partial differential equations based on multi quadric radial basis function networks [15]. Also Jianyu, Siwei, Yingjian, and Yaping in [16] described a neural network for solving partial differential equations in which activation functions of the hidden nodes were the radial basis functions (RBF) whose parameters are learnt by a two-stage gradient descent strategy. Also solving differential equation using neural network was applied to real problems [17].

Artificial neural networks are favorable techniques to solve optimization problems because they can simulate operations of human brain and uses parallel processing to save computational time [18]. Neural networks are demonstrated to have powerful capability of expressing relationship between input-output variables. In fact it is always possible to develop a structure that approximates a function with a given precision. However, there is still distrust about the neural networks identification capability in some applications [19].

Fuzzy logic approach is another intelligent computing tool that is competent for applying to wide variety of problems. Primary aim of Lotfi Zadeh which introduced the notion of a "fuzzy set" in 1965 was to set up a formal framework for the representation and management of vague and uncertain knowledge. Fuzzy set theory plays an important role in dealing with uncertainty in plant modeling applications.

Recently, there has been a growing interest in combining both these approaches, and as a result, neuro-fuzzy computing techniques have been evolved. Neuro-fuzzy systems are fuzzy systems, which use neural networks theory in order to determine their properties (fuzzy sets and fuzzy rules) by processing data samples [20]. Neuro-fuzzy integrates to synthesize the merits of both neural networks and fuzzy systems in a complementary way to overcome their disadvantage. The fusion of neural network and fuzzy logic in neuro-fuzzy models possess both low-level learning and computational power of neural networks and advantages of high-level human like thinking of fuzzy systems. Adaptive neural network based fuzzy inference system (ANFIS) presented by Jang [21]) combines the neural network adaptive capabilities and the fuzzy logic qualitative approach.

ANFIS has attained its popularity due to a broad range of useful applications in such diverse areas in recent years as optimization of fishing predictions [22], vehicular navigation [23], identify the turbine speed dynamics [24], radio frequency power amplifier linearization [25], microwave application [26], image denoising [27, 28], prediction in cleaning with high pressure water [29], sensor calibration [30], fetal electrocardiogram extraction from ECG signal captured from mother [31], identification of normal and glaucomatous eyes [32].

All above applications show that ANFIS is a good universal approximator, predictor, interpolator and estimator. They demonstrate that each non-linear function of many inputs and outputs can be easily constructed with ANFIS. Dynamic behavior of ANFIS motivates us in this study to use it in solving differential equations.

This paper aims at introducing a novel unsupervised ANFIS for numerical solution of differential equations of electric circuits. The paper is organized as follows. The architecture of ANFIS is explained in Section 2. Section 3 is devoted to solve differential equations using unsupervised ANFIS algorithm. Experimental results are discussed in section 4 and in final section conclusions are presented.

2. ANFIS architecture

ANFIS has been proved to have significant results in modeling nonlinear functions. In an ANFIS, the membership functions (MFs) are extracted from a data set that describes the system behavior. The ANFIS learns features in the data set and adjusts the system parameters according to given error criterion. In a fused architecture, NN learning algorithms are used to determine the parameters of fuzzy inference system. Below, we have summarized the advantages of the ANFIS technique.

- Real-time processing of instantaneous system input and output data's. This property helps using this technique for many operational researches problems.
- Offline adaptation instead of online system-error minimization, thus easier to manage and no iterative algorithms are involved.
- System performance is not limited by the order of the function since it is not represented in polynomial format.
- Fast learning time.
- System performance tuning is flexible as the number of membership functions and training epochs can be altered easily.
- The simple if–then rules declaration and the ANFIS structure are easy to understand and implement.

A typical architecture of ANFIS is shown in Fig 1, in which a circle indicates a fixed node, and a square indicates an adaptive node. For simplicity, we consider two inputs x, y and one output z in the fuzzy inference system (FIS). The ANFIS used in this paper implements a first-order Sugeno fuzzy model. Among many fuzzy inference systems, the Sugeno fuzzy model is the most widely used for its high interpretability and computational efficiency, and built-in optimal and adaptive techniques.

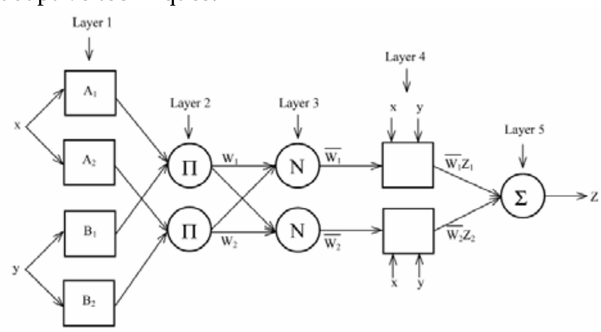

Fig.1: ANFIS architecture. $\Pi, \mathrm{N}, \sum$ are defined in (7), (8), (10) respectively.

For a first-order Sugeno fuzzy model, a common rule set with two fuzzy if-then rules can be expressed as:

Rule 1: If x is A1 and y is B1, then

$$z_1 = p_1 x + q_1 y + r_1 \tag{3}$$

Rule 2: If x is A2 and y is B2, then

$$z_2 = p_2 x + q_2 y + r_2 \tag{4}$$

where $A_i, B_i\,(i=1,2)$ Ai and Bi are fuzzy sets in the antecedent, and $p_i, q_i, r_i\,(i=1,2)$ are the design parameters that are to be determined during the training process. As in Fig 1, the ANFIS consists of five layers:

Layer 1, every node i in this layer is an adaptive node with a node function:

$$O_i^1 = \mu_{A_i}(x), \quad i=1,2$$
$$O_i^1 = \mu_{B_i}(y), \quad i=3,4 \tag{5}$$

Where x, y are inputs of node i, and $\mu_{A_i}(x)$ and $\mu_{B_i}(y)$ can adopt any fuzzy membership function. In this paper, Gaussian MFs are used:

$$gaussian(x,c,\sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2}. \tag{6}$$

where c is center of Gaussian membership function and σ is standard deviation of this cluster.

Layer 2, every node in the second layer represents the ring strength of a rule by multiplying the incoming signals and forwarding the product as:

$$O_i^2 = \omega_i = \mu_{A_i}(x)\mu_{B_i}(y), \quad i=1,2. \tag{7}$$

Layer 3, the ith node in this layer calculates the ratio of the ith rule's ring strength to the sum of all rules' ring strengths:

$$O_i^3 = \varpi_i = \frac{\omega_i}{\omega_1+\omega_2}, \quad i=1,2 \tag{8}$$

where ϖ_i is referred to as the normalized ring strengths.

Layer 4, the node function in this layer is represented by:

$$O_i^4 = \varpi_i z_i = \varpi_i(p_i x + q_i y + r_i), \qquad i=1,2 \tag{9}$$

where ϖ_i is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set. Parameters in this layer are referred to as the consequent parameters.

Layer 5, the single node in this layer computes the overall output as the summation of all incoming signals:

$$O_1^5 = \sum_{i-1}^{2} \varpi_i z_i = \frac{\omega_1 z_1 + \omega_2 z_2}{\omega_1+\omega_2} \tag{10}$$

It is seen from the ANFIS architecture that when the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters:

$$z = (\varpi_1 x)p_1 + (\varpi_1 y)q_1 + (\varpi_1)r_1 + (\varpi_2 x)p_2 + (\varpi_2 y)q_2 + (\varpi_2)r_2 \qquad (11)$$

The hybrid learning algorithm [21, 33] combines the least square method and the back propagation (BP) algorithm to solve the problem. This algorithm converges much faster since it reduces the dimension of the search space of the BP algorithm. During the learning process, the premise parameters in layer 1 and the consequent parameters in layer 4 are tuned until the desired response of the FIS is achieved. The hybrid learning algorithm has a two-step process. First, while holding the premise parameters fixed, the functional signals are propagated forward to layer 4, where the consequent parameters are identified by the least square method. Second, the consequent parameters are held fixed while the error signals, the derivative of the error measure with respect to each output node, are propagated from the output end to the input end, and premise parameters updated by the standard BP algorithm.

3. Unsupervised ANFIS for solving circuit differential equations

A linear time invariant electric circuit has a differential equation (DE) with constant coefficients expressed in the following form,

$$a_n \frac{d^n y(t)}{dt^n} + a_{n-1} \frac{d^{n-1} y(t)}{dt^{n-1}} + \ldots + a_0 y(t) = v_o(t). \qquad (12)$$

$$y(t) \in [a,b]$$

where $a_n, \ldots, a_0$ are constant coefficients and [a, b] is the interval in which the response must be obtained. Necessary initial or boundary conditions for solving this DE are,

$$y(0) = y_0^0,\ y^{(1)}(0) = y_0^{(1)},\ \ldots,$$

$$y^{(n-1)}(0) = y_0^{(n-1)}, \text{ or} \qquad (13)$$

$$y(t_0) = y_{t0},\ y(t_n) = y_{tn}$$

As have been pointed out in [14], a solution for above DE is:

$$\begin{aligned} y_p(t) &= f_i(x, y_0^{(0)}, y_0^{(1)}, \ldots, y_0^{(n-1)}) + g(h_i, u) \\ &= f_i(x, C) + g(h_i, u) \end{aligned} \qquad (14)$$

where $f_i(x,C)$ is a function for satisfaction of initial conditions, C is the initial conditions vector and $g(h_i,u)$ is a function which is zero in initial/boundary points, and u in other pints. u is an unsupervised adaptive neuro-fuzzy inference system. u has an important role, principally u is the main answer without contemplate of initial/boundary points. An interesting note is that the function u is expressed in the form of a fuzzy system and tuned using hybrid learning algorithm. h_i causes $g(h_i,u)$ to vanish at initial/boundary points. An easy and suitable form for $g(h_i,u)$ is $h_i u$. Therefore after substitution of (14) in (12) we can write:

$$a_n \frac{d^n y_p(t)}{dt^n} + a_{n-1} \frac{d^{n-1} y_p(t)}{dt^{n-1}} + \ldots + a_0 y_p(t) = v_o(t). \qquad (15)$$

then,

$$\hat{f}_i(x,C) + b_n \frac{d^n u_p(y_p, W, B)}{dt^n} + b_{n-1} \frac{d^{n-1} u_p(y_p, W, B)}{dt^{n-1}} + \ldots + b_0 u_p(y_p, W, B) = v_o(t). \qquad (16)$$

Where

$$\hat{f}_i(x,C) = a_n \frac{d^n f_i(x,C)}{dt^n} + a_{n-1} \frac{d^{n-1} f_i(x,C)}{dt^{n-1}} + \ldots + a_0 f_i(x,C)$$

W are weights of ANFIS as shown in Fig 1, B includes parameters of $\{p_i, q_i, r_i\}$ and input membership parameters. Also b_i are related to effect of h_i. Finally,

$$u_p(y_p, W, B) = -\frac{1}{b_0}\left(v_o(t) - \left(\hat{f}_i(x,C) + b_n \frac{d^n u_p(y_p, W, B)}{dt^n} + b_{n-1} \frac{d^{n-1} u_p(y_p, W, B)}{dt^{n-1}} + \ldots \right)\right). \qquad (17)$$

we can obtain desired output of $u_p(y_p, W, B)$ from (16) for learning ANFIS as follows,

Now we complete the proposed unsupervised ANFIS for solving differential equations with following algorithm:

Comment: Unsupervised ANFIS to solve differential equations

Initialize:

N: Resolution

dt: Sampling step size

t: Time vector including analysis points

Upi: Generation of fuzzy inference system using Monte-Carlo simulation

Up: Tuning of Upi using hybrid learning

Up_Old=Up;

Criteria= *Abs(Up-Up_old);*

While *Iteration <Max iteration* **or** *Criteria<* threshold

1:[b, f_hat]=**Call** *DF_Equ_Parameter*;

Comment: calculation of b_i, $\hat{f}$

2:[dUp]= **Call** *Ndifference*; **Comment:** calculation of $u_p^{(n)},\ldots,u_p^{(1)}$

3:Desired_Output=*Calculate of (17);*

4:Upi=**Call** *Generate FIS (t, Desired_Output)*

5:Up=**Call** *Tuning_Hybrid Learning (Upi)*

6:Criteria= *Abs(Up-Up_old);*

7:Up_Old=Up;

End While

8:**Display** (14)

In the above algorithm, firstly, initialization is performed including determination of number of point for analysis (*N*), sampling period (*dt*). In the first time step, we cannot calculate difference of u_p in line 2 so, before **While** we have generated an initial u_p. For this purpose, we can use Monte-Carlo simulation and one FIS is generated then we tune it using a hybrid learning algorithm.

A loop is necessary to receive to the convergence condition that includes maximum iteration or small changes in u_p in two consecutive iterations. Hands down we see if u_p is not changed considerably, then it satisfy (16) and it can be used in (14) for finding the answer (this work is done in line 8).

The rest of the algorithm includes line 1 for obtaining b_i, $\hat{f}$, line 2 for computation of $u_p^{(n)},\ldots,u_p^{(1)}$ for substitution in (17), line 3 for calculating (17), line 4 for generating FIS, line 5 for tuning FIS using hybrid learning scheme, and line 6 is measurement of stop criteria.

4. Experimental results

This section appropriates to solve several electric circuits using the proposed approach.

Example 1: First order differential equation with constant excitation. Equ. (18) is the differential equation of the circuit of fig. (2).

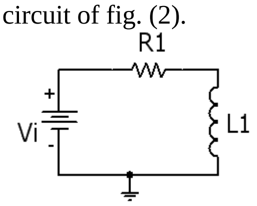

Fig. (2), L=1H, R=2Ω, y(t)=i_L(t)

$$\frac{d}{dt}y(t)+2y(t)=1,\ y(0)=1 \qquad (18)$$

The trial function will be in the form,

$$y_p(t)=1+(t-0)u_p(y_p,W,B) \qquad (19)$$

This solution satisfies the initial condition. After substitution of (19) in (18), the desired response of unsupervised ANFIS (u_p) is found.

$$u_p(y_p,W,B)=\frac{-1-y(t)\frac{d}{dt}u_p(y_p,W,B)}{1+2t} \qquad (20)$$

Figure 3 compares the analytical response with the response from the proposed method. Fig 4 shows *u*, the response of ANFIS model after convergence. As it is shown in Fig 5, in the first iteration ANFIS output is random because input and desired output is random. But after sixth iteration, ANFIS output is close to the exact solution.

Example 2: First order differential equation with sinusoidal excitation.

Suppose the circuit of fig. (2) with a sinusoidal excitation and initial condition y(0)=1.

$$\frac{d}{dt}y(t)+2y(t)=\sin(t),\ y(0)=1 \qquad (21)$$

The desired response of unsupervised ANFIS (u_p) is found as follows,

$$u_p(y_p,W,B)=\frac{\sin(t)-y(t)\frac{d}{dt}u_p(y_p,W,B)-2}{1+2t} \qquad (22)$$

The analytical response and the response from the proposed method are compared in Fig 6.

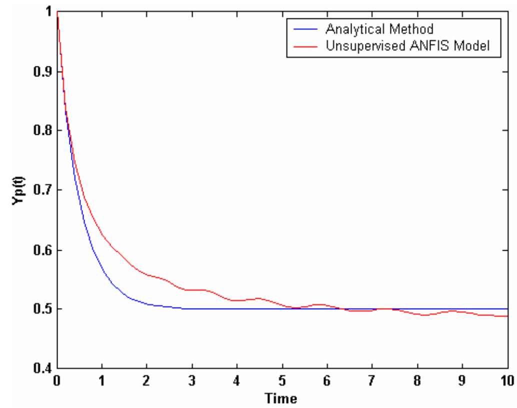

Fig.3: Responses collation about first order differential equation with constant excitation in example 1.

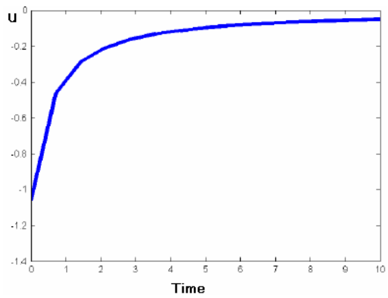

Fig.4: ANFIS output after convergence for example 1.

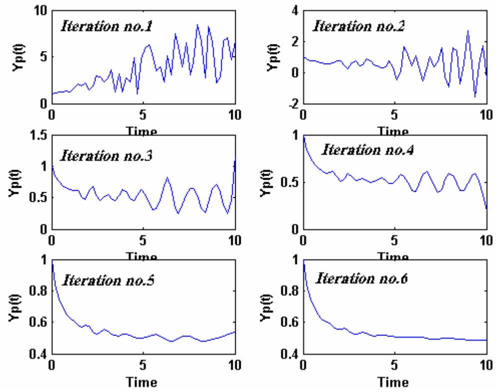

Fig.5: Response in different iterations in example 1.

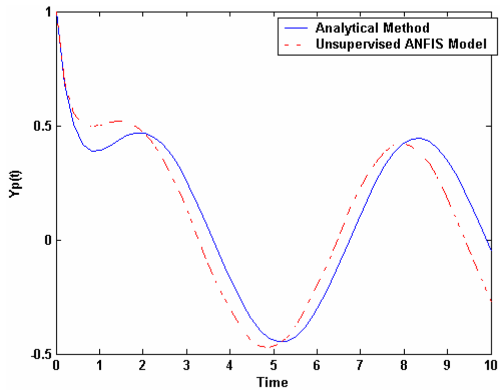

Fig.6: Responses collation in example 2 between analytical and the proposed methods.

5. Conclusion

An unsupervised neuro-fuzzy method is developed for solving differential equations of electric circuits. The method is evaluated with various examples, first and second, linear and nonlinear electric circuits. The comparison of unsupervised neuro-fuzzy solution with analytic solution shows versatility and accuracy of the proposed method.

References:

1. J. Xu, M. C. E. Yagoub, R. Ding, and Q. Zhang, "Neural-Based Dynamic Modeling of Nonlinear Microwave Circuits," IEEE Trans. On Microwave Theory and Techniques, V. 50, N. 12, 2002.
2. A.Veluswami, M. S. Nakhla, and Q. J. Zhang, "The application of neural networks to EM-based simulation and optimization of interconnects in high-speed VLSI circuits," IEEE Trans. Microwave Theory Tech., vol. 45, pp. 712–723, May 1997.
3. S. Goasguen, S. M. Hammadi, and S. M. El-Ghazaly, "A global modeling approach using artificial neural network," IEEE MTT-S Int. Microwave Symp. Dig., pp. 153–156, 1999.
4. S. Goasguen and S. M. El-Ghazaly, "A Practical Large-Signal Global modeling Simulation of a Microwave Amplifier Using Artificial Neural Network," IEEE Microwave and Guide Wave Letters, V.10, N.7, pp. 273–275, 2000.
5. M. Vai and S. Prasad, "Neural networks in microwave circuit design –Beyond black box models," Int. J. RF Microwave Computer-Aided Eng., vol. 9, pp. 187–197, 1999.
6. I. E. Lagaris, A. Likas, and D. I. Fotiadis. "Artificial Neural Network for Solving Ordinary and Partial Differential Equations," IEEE Transactions on Neural Networks, vol.9, no.5, pp.987–1000, 1998.
7. K. Hornik, "Multilayer Feedforward Networks are Universal Approximators." Neural Networks, vol.2, pp.359–366, 1989.
8. M. Hüsken, and C. Goerick, "Fast Learning for Problem Classes Using Knowledge Based Network Initialization," Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00), 2000.
9. N. Smaoui, S. Al-Enezi, "Modeling the Dynamics of Nonlinear Partial Differential Equations Using Neural Networks," Journal of Computational and Applied Mathematics, vol.170, pp.27–58, .
10. R. Brause, "Adaptive Modeling of Biochemical Pathways," Proceedings of the 15th IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI'03), 2003.
11. S. Hea, K. Reif, R. Unbehauen, "Multilayer Neural Networks for Solving a Class of Partial Differential Equations," Neural Networks, vol.13, pp.385–396, 2000.
12. L. Manevitz, A. Bitar, D. Givoli, "Neural Network Time Series Forecasting of Finite-Element Mesh Adaptation," Neurocomputing, vol.63, pp.447–463, 2005.
13. T. Leephakpreeda, "Novel Determination of Differential-Equation Solutions: Universal Approximation Method," Journal of Computational and Applied Mathematics, vol.146, pp.443–457, 2002.
14. A. Malek , R. Shekari Beidokhti, "Numerical Solution for High Order Differential Equations Using a Hybrid Neural Network—Optimization Method," Applied Mathematics and Computation, vol.183, pp.260–271.
15. N. Mai-Duy, T. Tran-Cong, "Numerical Solution of Differential Equations Using Multi Quadric Radial Basis Function Networks," Neural Networks, vol.14, pp.185-199, 2001.
16. L. Jianyu, L. Siwei, Q. Yingjian, H. Yaping, "Numerical Solution of Elliptic Partial Differential Equation Using Radial Basis Function Neural Networks," Neural Networks, vol.16, pp.729–734, 2003.
17. D. R. Parisi, M. C. Mariani, M. A. Laborde, "Solving Differential Equations with Unsupervised Neural Networks," Chemical Engineering and Processing, vol.42, pp.715-721, 2003.
18. A. Cichocki, R. Unbehauen, K. Weinzierl and R. Holzel, A new neural network for solving linear programming

problems, European J. of Operational Research 93, 244-256, (1996).
19. Castro, A., Miranda, V., 2002. Mapping neural networks into rule sets and making their hidden knowledge explicit application to spatial load forecasting. In: Proceedings of the 14th Power System Computation Conference.
20. Mitra, S., Hayashi, Y., 2000. Neuro-fuzzy rule generation: survey in soft computing framework. IEEE Transactions on Neural Networks 3, 748–768.
21. Jang, J.-S.R. (1993).ANFIS: adaptive-network-based fuzzy inference system. IEEE Transaction on System Man and Cybernet, 23(5), 665–685.
22. A. I. Nuno, B. Arcay, J.M. Cotos, J. Varela, "Optimization of Fishing Predictions by Means of Artificial Neural Networks, ANFIS, Functional Networks and Remote Sensing Images," Expert Systems with Applications, vol.29, pp.356–363, 2005.
23. A. Noureldin, A. El-Shafie, M. R. Tahab, "Optimizing Neuro-Fuzzy Modules for Data Fusion of Vehicular Navigation Systems Using Temporal Cross-Validation," Engineering Applications of Arti.cial Intelligence, vol.20, pp.49–61, 2007.
24. N. Kishor, S.P. Singh, A.S. Raghuvanshic, "Adaptive Intelligent Hydro Turbine Speed Identification with Water and Random Load Disturbances," Engineering Applications of Arti.cial Intelligence (2007), doi:10.1016/j.engappai.2006.11.014.
25. K. C. Lee, and P. Gardner, "Adaptive Neuro-Fuzzy Inference System (ANFIS) Digital Predistorter for RF Power Amplifier Linearization," IEEE Trans. on Vehicular Technology, vol..55, no.1, pp.43-51, Jan. 2006.
26. E. D. Ubeyli, I. Guler, "Adaptive Neuro-Fuzzy Inference System to Compute Quasi-TEM Characteristic Parameters of Micro Shield Lines with Practical Cavity Sidewall Profiles, Neurocomputing, vol.70,pp.296–304, 2006.
27. H. Qin, S. X. Yang, "Adaptive Neuro-Fuzzy Inference Systems Based Approach to Nonlinear Noise Cancellation for Images, Fuzzy Sets and Systems (2007), doi: 10.1016/j.fss.2006.10.028
28. P. Çivicioglu, "Using Uncorrupted Neighborhoods of the Pixels for Impulsive Noise Suppression with ANFIS," IEEE Trans. On Image Processing, (2007), Digital Object Identifier 0.1109/TIP.2007.891067.
29. G. Daoming, C. Jie, "ANFIS for High-Pressure Water jet Cleaning Prediction," Surface & Coatings Technology, vol.201, pp.1629–1634, 2006.
30. A. Depari, A. Flammini, D. Marioli, and Andrea Taroni, "Application of an ANFIS Algorithm to Sensor Data Processing," IEEE Trans. On Instrumentation and Meaturemnet, vol.56, no.1, pp.75-79, Feb. 2007.
31. K. Assaleh, "Extraction of Fetal Electrocardiogram Using Adaptive Neuro-Fuzzy Inference Systems," IEEE Trans. On Biomedical Engineering, vol.54, no.1, pp.59-68, Jan. 2007.
32. M-L. Huang, H-Y. Chen, J-J. Huang, "Glaucoma Detection Using Adaptive Neuro-Fuzzy Inference System," Expert Systems with Applications, vol.32, pp.458–468, 2007.
33. Jang J-SR, Sun CT, Mizutani E, "Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence. Prentice-Hall, Englewood Cli.s, NJ, 1997.
34. M. Kosal, S. Herderm, "Analysis of Nonlinear Circuits by Using Differential Taylor Transform," Computers and Electrical Engineering, vol.28, pp.513-525, 2002.
35. D. E. Amos, "A subroutine package for Bessel functions of a complex argument and nonnegative order", Sandia National Laboratory Report, SAND85-1018, May, 1985.
36. D. E. Amos, "A portable package for Bessel functions of a complex argument and nonnegative order", Trans. Math. Software, 1986.