**Online Elman Neural Network Training by Genetic Algorithm**

**Ali Hussein Hasan and Wathiq Hayawi Laith**

**University of Sumer, Thi-Qar, Al-Rifai, Iraq**

**Abstract**

Although most offline and online training algorithms based on gradient search techniques like backpropagation algorithm and its modifications or on Kalman filter approaches, it has been shown that these techniques are severely limited in their ability to find global solutions, they converge slowly, get local minimization too easily and courses oscillation. Global search techniques have been identified as a potential solution to this problem, but they are limited to offline training because the long time of convergence. In this paper we submit an online fast global search technique based on modifications of the basic genetic algorithm, we apply the new approach especially on the Elman network which is generally suffer from very long training time.

**Introduction**

Artificial Neural Networks (ANN’s), also known as connectionist models, are rapidly evolving facet of artificial intelligence (AI) [1]. In particular, the main idea is to reproduce the intelligence and the capability to learn from examples, simulating the brain neuronal structure on a calculator. ANN’s are interconnected networks of simple elements which interact with the objects of the real world in the same way as biological nervous system does. Because of the biological basis for the artificial neural networks, it is not surprising that many of the terms used in their study are borrowed from neurophysiology [1]. The processing units are neurons, nodes or processors; while the connections between these units are known as interconnected, synapses or weights. The pattern of the connections between the units determines the architecture of the network, which in the extremes, can be fully interconnected (recurrent neural networks) or connected in one direction only (feed forward neural networks).

Multi layers feedfarward neural network based on series connection of neuron layers, each one composed by a set of neurons connected in parallel. The signals flow from the input layer through the hidden layer(s) to the output layer via uni-directional connections. There are one or more layers between the input and output layers.

Feedforward networks can be only used for dynamic relationship between input and output variable by including lagged values of input and output variables in the input layer. Recurrent Neural Network (RNN) allows for an internal feedback in the system. Internal feedback is a more successful way to account for dynamics in the model. It contains the entire history of inputs as well as outputs [2]. Recurrent neural networks are fundamentally different from feedforward architectures in the sense that they not only operate on an input space but also on an internal state [3].

ANN’s must be learned before its using. There are many learning algorithms that can be used to train ANN’s; back-propagation (with its modifications) is currently the mainstay of artificial neural networks learning. BP algorithm has high accuracy, but it has some disadvantages: it converges slowly, gets local minimization too easily and courses oscillation. Local minimization can be solved by adjusting the initial weights, while slow convergence and the oscillation are usually coursed by getting into local minimization in the later period of the network training [4]. Miao et al[8] indicated that the BP solutions are usually forced to the local minimum due to the gradient descent algorithm used to get weights of connections. Engoziner et al. [9] presented that BP use some variation of the gradient technique, which is essentially a local optimizing method and thus has some inevitable drawbacks, such as easily trapping into the local optimal and dissatisfying generalization capability.

Genetic algorithms (or simply GAs) are powerful and widely applicable stochastic search and optimization methods based on the concepts of natural selection and natural evaluation. Genetic algorithms were first invented by John Holland in 1960s and were developed by Holland and his students and colleagues at the university of Michigan in the 1960s and the 1970s [5]. GA shows great promise in complex domains because it operates in an iterative improvement fashion. The search performed by it is probabilistically concentrated towards regions of the given data set that have been found to produce a good classification behavior. An overview about GAs and their implementation in various ﬁelds was given by Goldberg [6] or Michalewicz [7]. GA may be used to do one or more of the following when it's combined with neural networks: - 1. Wight training for supervised learning and reinforcement learning applications. 2. Select training data and 3. Finding neural network architecture.

Consequently, newly research tends to hybridize several artiﬁcial intelligence (AI) techniques to improve the performance. Topchy et al [10] proposed two algorithms to learn ANN weights; in the first proposed algorithm, learning process can be considered as evolutionary adaptation of network parameters to optimal internal representation of the information being processed. The second algorithm mainly focuses on direct combination of genetic algorithm and delta- rule for training two subsets of multilayered perceptron's parameters, i.e. hidden and output layer weights respectively. Sexton et al [11] made a comparison between the backpropagation and genetic algorithms. Sexton et al [12] examined two well known global search techniques, Simulated Annealing and the Genetic Algorithm, and compare their performance in neural network training. Schiffmann et al [13] wrote a technical report about using genetic algorithms in neural networks fields. Topchy et al [14] presented a RBFN training algorithm based on evolutionary programming and cooperative evolution. Gruau [15] combined neural network architecture optimization and weights learning in single algorithm. Schaffer et al [16] made a good survey of various combinations of genetic algorithms and neural networks.Fogel [17] used evolutionary programming to create neural networks that are capable of playing Tic-Tac-Toe. Wies[18] focused on the intersection of neural networks and evolutionary learning and showed the basic aspects of combination of these learning paradigm. Gupta and Sexton [19] showed that the use of GA can provide better results for training feedforward NN than the traditional techniques of backpropagation. Other interesting papers deal with NN optimized by GA can be founded in [20-29].

**Elman Recurrent Neural Networks**

The block diagram of Elman network is shown in figure 1; in addition to the basic layers, inputs, hidden and output layers, there is also context layer. The context layer contains the previous output of the hidden layer. The input vector () with the output of the context layer propagate through a weight layer to the hidden layer. The output of the network is the linear combination sum of the hidden layer outputs.

Let the input signal is ();

Where:-

is the net sum of the hidden layer neurons.

is the output of the hidden layer neurons.

is the net sum of the output layer neurons.

is the outputs of the network.

is the connection weight between the input and hidden layer.

is the connection weight between the hidden layer and context layer.

is the connection weight between the hidden and output layer.

activation function of the hidden layer neurons.

*yk+1*

*uk*

***-i-***

*-1-*

Figure 1:- Elman neural network structure.

**Proposed GA Learning**

GAs have been shown to be an effective strategy in the off-line design of many fields. The GAs are not too demanding, as could be natural to expect, in terms of their needs of computational power, being applied so far to a wide range of problems, going from the fields of Combinatorial or Numeric Optimization to Image Processing or even Machine Learning. GAs operate on a population of potential solutions applying the principle of survival of the ﬁttest to produce (hopefully) better and better approximations to a solution. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of ﬁtness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.

A genetic algorithm has three major components. The first component is related with the creation of an initial population of *m* randomly selected individuals. The initial population shapes the first generation. The second component inputs *m* individuals and gives as output an evaluation for each of them based on an objective function known as fitness function. This evaluation describes how close to our demands each one of these *m* individuals is. Finally the third component is responsible for the formulation of the next generation. A new generation is formed based on the fittest individuals of the previous one. This procedure of evaluation of generation *N* and production of generation *N+1* (based on *N*) is iterated until a performance criterion is met. The creation of offspring based on the fittest individuals of the previous generation is known as breeding. The breeding procedure includes three basic genetic operations: reproduction, crossover and mutation.

Reproduction selects probabilistically one of the fittest individuals of generation *N* and passes it to generation *N+1* without applying any changes to it. Crossover selects probabilistically two of fittest individuals of generation *N*; then in a random way chooses a number of their characteristics and exchanges them in a way that the chosen characteristics of the first individual would be obtained by the second and vice versa. Following this procedure creates two new offspring that both belong to the new generation. Finally the mutation selects probabilistically one of the fittest individuals and changes a number of its characteristics in a random way. The offspring that comes out of this transformation is passed to the next generation.

The basic idea of this work is by inserting the best chromosomes from previous generations in the elite matrix; then we check the performance of the problem, now if the elite matrix matches the requirements then exist the GA, else perform the GA and update the elite matrix. Figure 2 summarize the proposed algorithm:

General Online GA:

Initialization:-

Get the first input and output measurements.

Perform GA initialization by create random Population; i=1.

Calculate the best performance chromosome.

Elite Matrix[i]= best performance chromosome.

For each input/ output measurements, do:

Checks the Elite Matrix == required performance.

Test the stopping conditions: if they matching the requirement then Designate the results and get another input/output measurement. Else:

i=i+1;

Perform GA.

Elite Matrix[i] = best performance chromosome.

Start

Create initial random Population

i=1

Enter

For l=1:i Evaluate NN by Elite Matrix

yes

Criterion is Satisfied

Designate Results

No

Perform GA: Do until reach max generation

* rank individuals
* Select individuals
* recombine individuals
* Apply mutation
* Evaluate offspring



No

yes

Elite Matrix[i]=Best Chromosome

i=i+1

Figure 2: Proposed Genetic Algorithm

**Results:**

This section shows some simulation results, the testing signal of 100 input samples used is:-

...........................................7

The real number encoding is employed. The population size is set to 20, and the initial generation number is set to 1000 then reduced to 50 after one cycle.

The simulation is made on the following three plants that differ in their degree of nonlinearity [30].

Plant Number 1:- Which is a linear in both input and output behaviour, as describe in equation (8).

.........…..….....8

Where A1=1.752821, A2= -0.818731, B1=0.011698, B2=0.010942

The response is shown in figure 3.

Plant Number 2:- Which is a linear in the input and nonlinear in the output behaviour, as describe in equation (9).

......................9

Where A1=1.04, A2= -0.824, A3=0.130667, B1= -0.16

The response is shown in figure 4.

Plant Number 3:- Which is a strong nonlinear in both input and in the output behaviours, as describe in equation (10).

.........................................................10

The response is shown in figure 5.

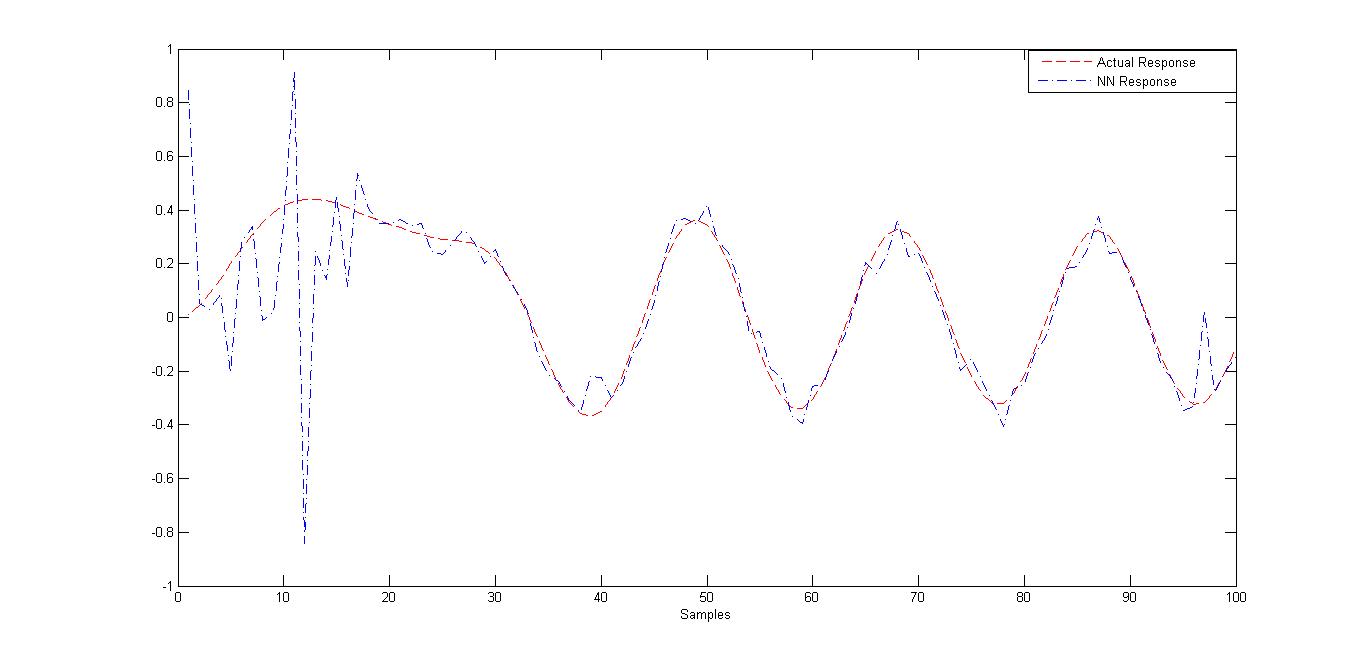
****

Figure 3: First System output response

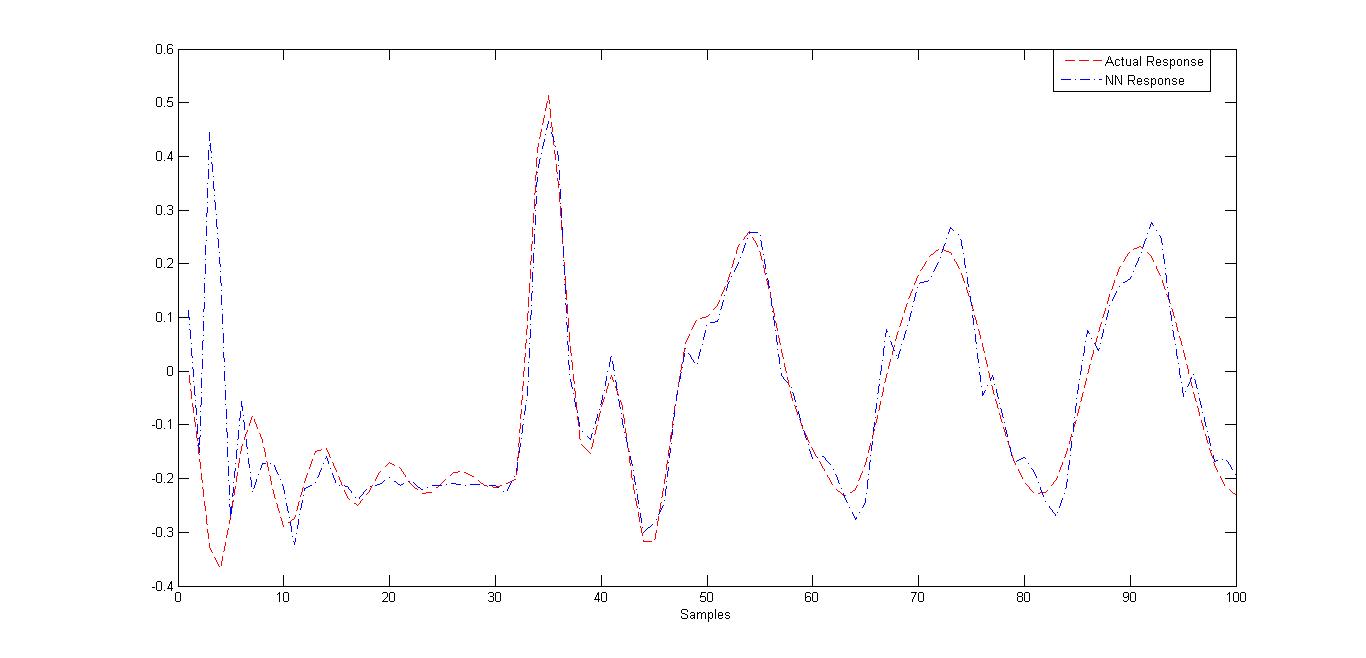
****

Figure 4: Second System output response

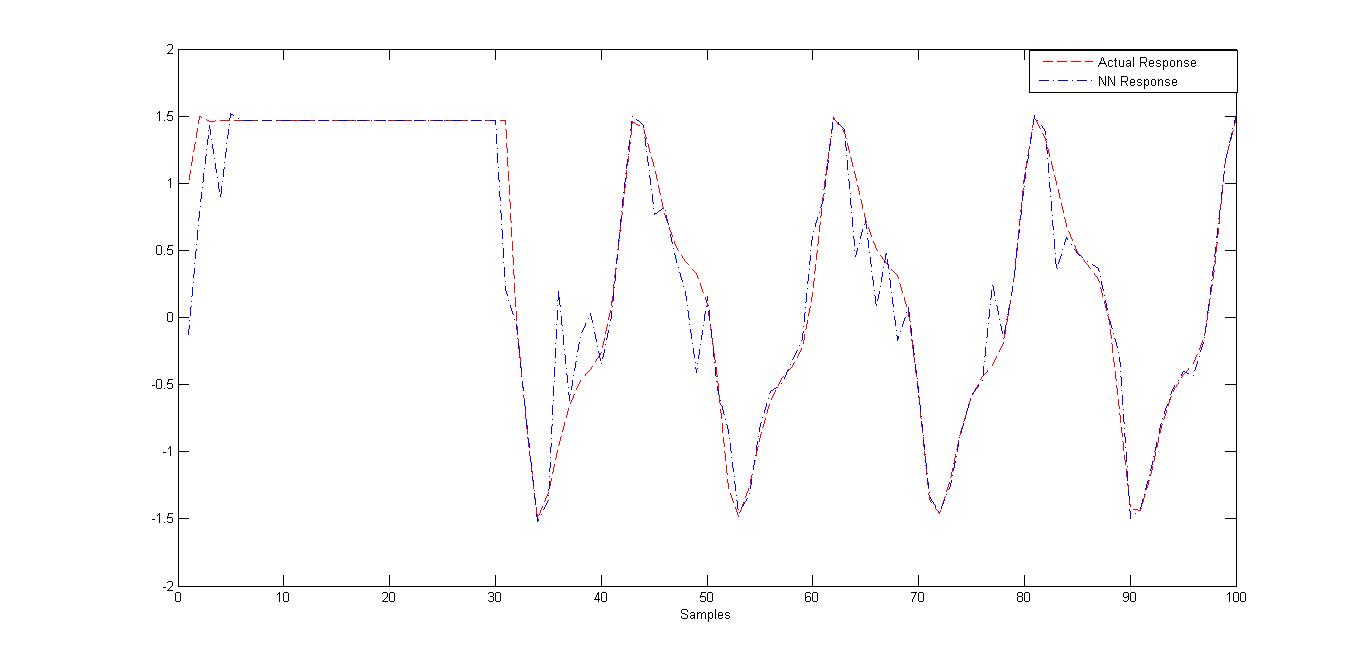
****

Figure 5: Third System output response

**Conclusions**

Its known that most NN especially Elman NN suffer from long time convergence problem as well as falling in local minima points when its optimized by either backpropagation algorithms or real time Extended Kalman filter. In this work, we proposed an online promising optimization algorithm based on GA, which is fast convergence and global minima points searching. We simulate the algorithm with three dynamical systems with different nonlinearities, we show that the proposed approach suitable for real time NN application.

**References**

Gary M. Scott and W. Harmon Ray, “Experiences with model-based controllers based on neural network process models”, Dept of Chemical Engineering, University of Wisconsin, Madison, USA, http//www.esf.edu,1993.

Haykin, S.: "Neural Network: A Comprehensive Foundation," Macmillan Publishing Company, NJ, Lehigh Press, 1994.

Mikael Bo d´en, "A guide to recurrent neural networks and backpropagation", School of Information, Science, Computer and Electrical Engineering, Halmstad University. November 13, 2001.

Yongli Wang, "Optimizing of Artificial Neural Network Based on Immune Genetic Algorithm in Power Load Forecasting", In Proceedings of ACIIDS Posters. 2011, 329-338.

Mitchell, M. “An Introduction to Genetic Algorithms”, Massachusetts Institute of Technology, England, first MIT press paperback edition, 1998.

J. H. Holland, "Adaptation in Natural and Artiﬁcial Systems", University of Michigan Press, Ann Arb or, MI, 1975.

Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, Berlin Heidelb erg New York, 1996.

Miao, K., F. Chen, and Z.G. Zhao, “Stock price forecast based on bacterial colony RBF neural network,” Journal of Qing Dao University, Vol.20, 50–54, 2007.

Engoziner, S. and E.Tomesn, “An accelerated learning algorithm for multiplayer perception: Optimization layer by layer,” IEEE Trans.onNeuralNetwork, Vol.6, 31–42, 1995.

Topchy, P. Alexander, Lebedko, A. Oleg and Miagkikh, V. Vector, "FAST LEARNING IN MULTILAYERED NEURAL NETWORKS BY MEANS OF HYBRID EVOLUTIONARY AND GRADIENT ALGORITHMS", in Proc. Of the first Int. Conf. on Evolutionary Computations and its Applications, ed. E.D. Goodman et al, (RAN, Moscow), pp 390-399, 1996.

Sexton, R. S., Dorsey, R. E., & Johnson, J. D., " Toward Global Optimization of Neural Networks: A Comparison of Backpropagation with Genetic Algorithms", Decision Support Systems, 22(2), pp 171-186, 1998.

Sexton, R. S., Dorsey, R. E., & Johnson, J. D.," Optimization of Neural Networks: A Comparative Analysis of the Genetic Algorithm and Simulated Annealing", European Journal of Operational Research, 1999.

W. Schiffmann, M. Joost, and R. Werner. Synthesis and performance analysis of multilayer neural network architectures. Technical Report 16, University of Koblenz, Institute of Physics, 1992.

Topchy, A., Lebedko, O., Miagkikh, V., Kasabov, N.: Adaptive training of radial basis function networks based on co-operative evolution and evolutionary programming. In: Kasabov, N., et al. (eds.) Prog. in connectionist-based information syst. pp. 253-258. Springer, Heidelberg (1998).

GRUAU, Frederic, "Genetic synthesis of Boolean neural networks with a cell rewriting developmental process", [International Workshop on](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=893) [Combinations of Genetic Algorithms and Neural Networks, COGANN-92](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=893), pp 55-74, 6 June 1992.

Schaffer, J. David, Whitley, Darrell and Eshelman, J. Larry, " Combinations of Genetic Algorithms and Neural Networks: A Survey of the State of the Art ", [International Workshop on](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=893) [Combinations of Genetic Algorithms and Neural Networks, COGANN-92](http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=893), pp 1-37, 6 June 1992.

Fogel, D.B, "Using Evolutionary Programming to Create Neural Networks that are Capable of Playing Tic-Tac-Toe", IEEE International Conference on Neural Networks, Orincon Corp., San Diego, CA, pp 875 - 880 vol.2, 1993.

Wies, Gerhard, "Towards the Synthesis of Neural Networks and Evolutionary Learning", Progress in Neural Networks: Architecture, edited by Omid M. Omidvar, Charles L. Wilson, pp 145-174, 1997.

Gupta, N.D. Jatinder and Sexton, S. Randall, "Comparing backpropagation with a genetic algorithm for neural network training", Omega. v27. 679-684, 1999.

David J. Montana and Lawrence Davis, "Training Feedforward Neural Networks Using Genetic Algorithms", Proceedings of the International Joint Conference on Artificial Intelligence, Vol. 1, pp 762-767, 1989.

Jingbo Xu, Huimin Lu and Qiang Li, " OPTIMIZATION OF PROCESS PARAMETERS OF PREPARING FOAMED A1-Si ALLOY BASED ON GA-BASED BP NEURAL NETWORK ", International Conference on Advances in Materials and Materials Processing ICAMMP 2011, Department of Metallurgical and Materials Engineering Indian Institute of Technology, Kharagpur December 9-11, 2011.

Chien-Yu Huang and Chieh Huang, " A Robust Design Approach for Genetic Algorithm-based Back Propagation Neural Networks", Proceedings of International Conference on Business and Information (BAI2012), Sapporo, Japan, 3-5 July 2012.

Byoung-Tak Zhang and Heinz Mühlenbein, "Evolving Optimal Neural Networks Using Genetic Algorithms with Occam's Razor", Complex Systems, Vol. 7 No.3, pp: 199-220, 1993.

Ioan Ileană, Corina Rotar, Arpad Incze، " THE OPTIMIZATION OF FEED FORWARD NEURAL NETWORKS STRUCTURE USING GENETIC ALGORITHMS ", Proceedings of the International Conference on Theory and Applications of Mathematics and Informatics - ICTAMI 2004, Thessaloniki, Greece.

Milos Manic and Bogdan Wilamowski, "Robust Algorithm for Neural Network Training", Proceedings of International Joint Conference on Neural Networks, 2002. IJCNN '02, Vol. 2, pp: 1528 – 1533, 2002.

Sexton, R. S., Dorsey, R. E., & Johnson, J. D., " BEYOND BACKPROPAGATION: USING SIMULATED ANNEALING FOR TRAINING NEURAL NETWORKS", Journal of Organizational and End User Computing (JOEUC), Volume 11, Issue 3, 1999.

Sexton, R. S., Alidaee, Bahram, Dorsey, R. E., & Johnson, J. D.," Global Optimization for Artificial Neural Networks: A Tabu Search Application", European Journal of Operational Research. v106. 570-584, 1998.

Xin Yao، " Evolving artificial neural networks", Proceedings of the IEEE, Vol.87,، Issue: 9, pp: 1423 – 1447, Sep 1999.

Weiss, G.," Neural networks and evolutionary computation. I. Hybrid approaches in artificial intelligence ", Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, Vol. 1, pp: 268 – 272, 27-29 Jun 1994.

DUC Truohg Pham and Liu Xing, “Neural Networks for Identification, Prediction and Control”, Springer-Verlag, 2nd, 1995.