Comparison of Four Neural Network Learning Methods Based on Genetic Algorithm for Non-linear Dynamic Systems Identification

Dr. Rafid Ahmed Khalil, Assist. Prof.,
Mechatronics Eng. Dept., Engineering College,
Mosul University
rafidamori@ymail.com

Abstract

Non-linear dynamical systems are difficult to control due to the model uncertainties and external disturbances that may occur in these systems. This paper addresses the problem of identification using dynamic neural networks (DNNs) based on genetic algorithm (GA) for nonlinear dynamic systems. Four different dynamic neural networks are used for identification of the same nonlinear dynamic system, using the genetic algorithm (GA) to train the Layer-Recurrent Network (LRN), Focused Time-Delay Neural Network (FTDNN), the Elman Network, and Nonlinear Autoregressive Network with Exogenous inputs (NARX). The simulation results show the generalization ability of the four dynamic neural networks which provide the high precision of model of the nonlinear dynamic system. Also this paper illustrates the advantages and disadvantages of the different dynamic neural networks trained by GA.

Keywords: Dynamic Neural Networks, Nonlinear system identification, genetic algorithm.

مقارنة تدريب اربعة شبكات عصبية مبنية على GA في مطابقة الانظمة الديناميكية غير الخطية

د. رافد أحمد خليل

قسم هندسة الميكاترونك / كلية الهندسة

ملخص

من الصعب التحكم في الانظمة الديناميكية الغير خطية بسبب عدم وضوح النموذج (**model**) والتاثيرات السلبية الخارجية التي تؤثر على هذه الانظمة. هذا البحث يناقش مسالة مطابقة (**Identification**) للانظمة الغير خطية الديناميكية باستخدام الشبكات العصبية الديناميكية **DNNs** والمبنية على الخوارزمية الجينية (**GA**). استخدمت اربعة شبكات عصبية هي شبكة **LRN**,شبكة **FTDNN**, شبكة **Elman**, وشبكة **NARX**. نتائج المحاكات وضحت التعميم التي تمتلكها الشبكات العصبية الديناميكية المستخدمة.بالاضافة الى الدقة العالية في نمذجة او تطابق النظام الديناميكي الغير خطي.ايضا وضح هذا البحث مزايا ومساوئ الشبكات العصبية الديناميكية الاربعة باستخدام طريقة التدريب المبنية على الخوارزمية الجينية (**GA**).

Received: 8 – 12 - 2009 **Accepted: 8 – 9 - 2011**

1. Introduction

Artificial neural networks (ANNs) are the computing architecture that consists of massively parallel interconnections of simple computing elements. ANNs have been implemented in various fields such as, image processing, speech recognition, and system identification etc. The fundamental and core property of ANNs for their superiority over other approximation methods are based on the fact that ANNs are able to universally approximate smooth but otherwise arbitrary nonlinear functions on a wide range of complex nonlinear functions on a compact set, using fewer parameters and requiring less computation time [1].

The ANNs are commonly used as a data-based technique in performing non-linear system identification of complex processes. For this purpose, models with adequate memory are required. Therefore, the ANNs have to be provided with dynamic elements and appropriate learning methods [2]. A first approach refers to neural networks with external dynamics, e.g. static ANNs equipped with tapped delay lines, the latter increasing the dimensions of the ANN’s input space. A better approach is achieved by providing the ANN with internal dynamics. This kind of network processes multi-inputs and does not require past values of process measurements as current inputs [3]. There are several ways to deal with nonlinear control design for plants subject to uncertainty and disturbances; among them are robust and adaptive control methods. While neural adaptive control is being intensively developed, so as the applications of neural networks to nonlinear control [2]. For example, Narendra et. al. [4], introduced multilayer neural networks for identification and adaptive control of nonlinear systems. A number of studies such as [5-10], presented an adaptive control of unknown feedback linearizable system for achieving guaranteed performance of the neural network. It has excellent capabilities of nonlinear mapping, learning ability, and parallel computations.

This paper first provides background information on Dynamic Neural Networks architectures, learning algorithm based on genetic algorithm (GA), and a theoretical basis for their use. Then it presents simulations for these architectures for identifying nonlinear dynamic systems or plant. Finally, it shows some new results and discussion for these results.

2. Dynamic Neural Networks (DNNs) for Systems Identification

Artificial neural networks (ANNs) can be classified into dynamic and static categories. Static feed-forward networks have no feedback elements and contain no delays; the output is calculated directly from the input through feed-forward connections. In dynamic networks, the output depends not only on the current input to the network, but also on the current or previous inputs, outputs, or states of the network [11]. Dynamic networks can also be divided into two categories: those that have only feed forward connections, and those that have feedback, or recurrent, connections. Dynamic networks are generally more powerful than static networks (although slightly more difficult to train). Because dynamic networks have memory, they can be trained to learn sequential or time-varying patterns. This has applications in such different areas as channel equalization in communication systems, phase detection in power systems, fault detection, speech recognition [3]. ANNs are used to model the dynamics of a system or plant. Usually the ANN and the plant are feed with the same input signal (or input signals for multi-input systems) for training, then neural network used to model the plant behavior [12]. The network output is compared with the output from the plant and the error is used to update the weights of the synapses (see Fig.1)..

A simple feed-forward neural network can be used, if the ANN is feed with the current input, the previous inputs and the previous outputs. Using this configuration an input-output model of the plant is obtained, instead of a state space model. The advantage of this model is that a series-parallel model can be used while training the network. In a series-parallel model (Fig. 2) the previous outputs from the plant are used as inputs to the ANN instead of the outputs from the proper neural network. Once the model is obtained then the input scan is taken from the previous outputs of the ANN obtaining and autonomous system.

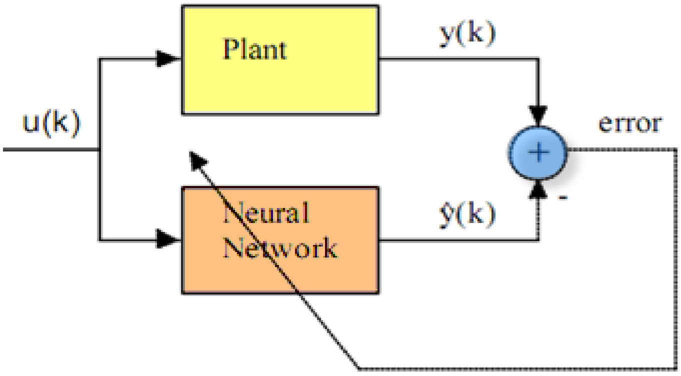


Fig. 1. Artificial neural structure for plant (system) identification

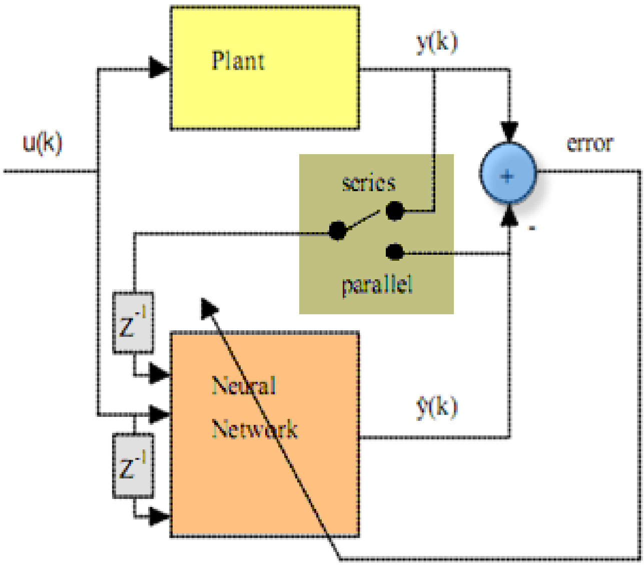


Fig. 2. Serial-parallel identification training model.

This work demonstrates how to create, train ANN based on GA, and apply certain dynamic neural networks in identification nonlinear dynamic system. Some of the networks require dynamic back propagation for computing the gradients and others do not. This section, presents the following four dynamic neural networks.

a. Layer-Recurrent Network (LRN)

The dynamic network to be introduced is the Layer-Recurrent Network (LRN) [5][9]. An earlier simplified version of this network was introduced by Elman [6]. In the LRN, there is a feedback loop, with a single delay, around each layer of the network except for the last layer. The original Elman network had only two layers, and used a tansig transfer function for the hidden layer and a purelin transfer function for the output layer. The network is trained using an approximation to the BP gradient-based algorithm. The network has an arbitrary number of layers and to have arbitrary transfer functions in each layer. Fig. (3) Illustrates a two-layer LRN (according to Matlab abbreviated notation [5]).

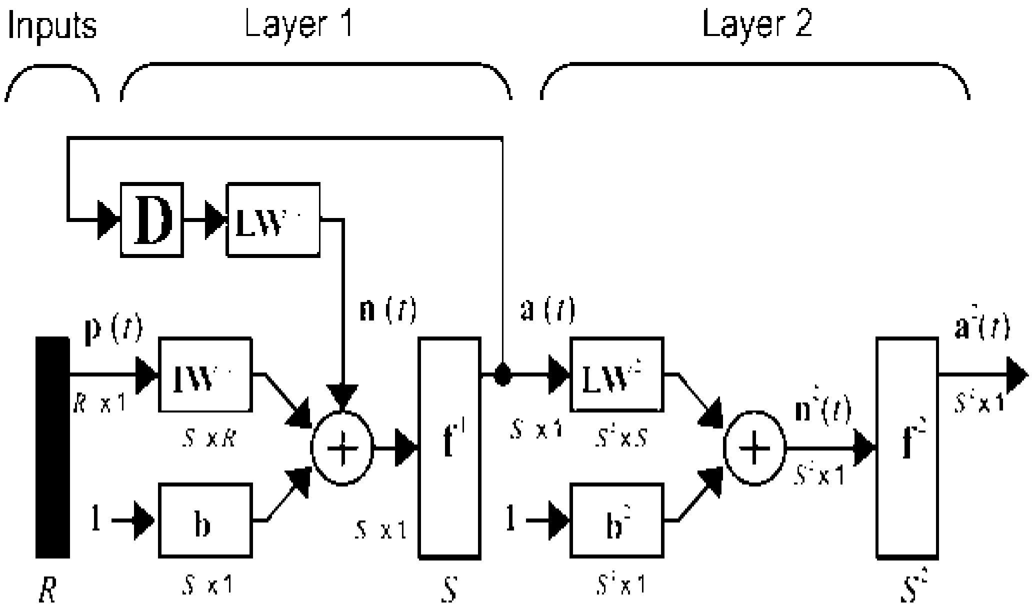


Fig. 3. Layer-Recurrent Network (LRN) architecture.

b. Focused Time-Delay Neural Network (FTDNN)

The next network is the most straightforward dynamic network, which consists of a feed-forward network with a tapped delay line at the input. This is called the focused time-delay neural network (FTDNN) [5]. This is part of a general class of dynamic networks, called focused networks, in which the dynamics appear only at the input layer of a static multilayer feed-forward network. The following Fig. (4), illustrates a two-layer FTDNN.

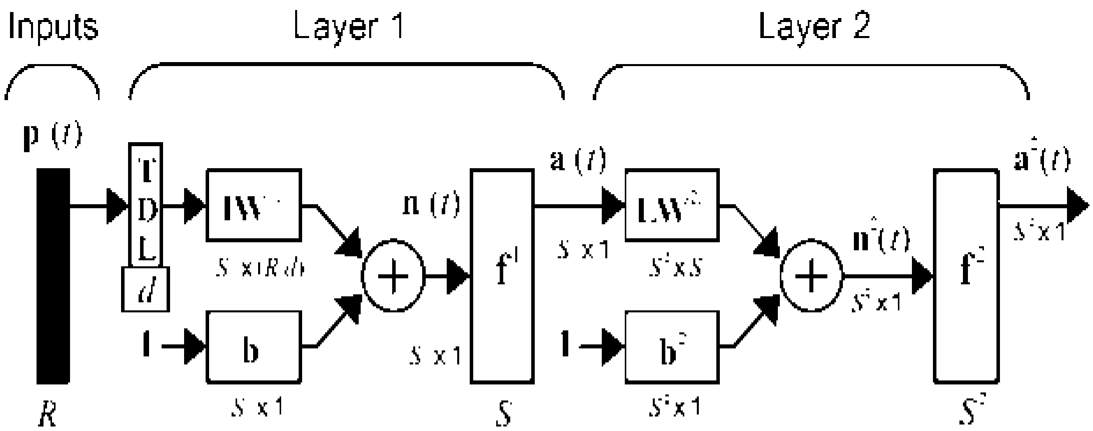


Fig. (4) Focused time-delay neural network (FTDNN) architecture.

c. Elman Network

The Elman network commonly is a two-layer network with feedback from the first-layer output to the first-layer input [5-6]. This recurrent connection allows the Elman network to both detect and generate time-varying patterns. A two-layer Elman network is shown Fig. (5). The Elman network has tansig neurons in its hidden (recurrent) layer, and purelin neurons in its output layer. This combination is special in that two-layer networks with these transfer functions can approximate any function (with a finite number of discontinuities) with arbitrary accuracy. The only requirement is that the hidden layer must have enough neurons. More hidden neurons are needed as the function being fitted increases in complexity.

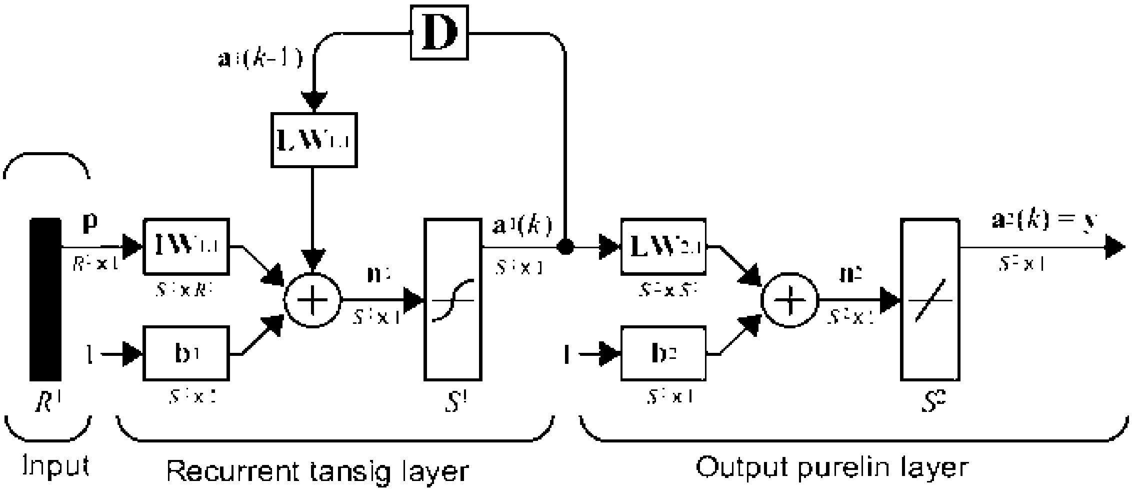


Fig. (5) Elman network architecture.

Note that the Elman network differs from conventional two-layer networks in that the first layer has a recurrent connection. The delay in this connection stores values from the previous time step, which can be used in the current time step. Thus, even if two Elman networks, with the same weights and biases, are given identical inputs at a given time step, their outputs can be different because of different feedback states. Because the network can store information for future reference, it is able to learn temporal patterns as well as spatial patterns. The Elman network can be trained to respond to, and to generate, both kinds of patterns. The default BP training algorithm is used. You might use BP, but it tends to proceed so rapidly that it does not necessarily do well in the Elman network. The backpropagation weight/bias learning function default is, and the default performance function is mean squared error (mse). When the network is created, each layer's weights and biases are initialized with the Nguyen-Widrow layer initialization method.

d. NARX Network

The nonlinear autoregressive network with exogenous inputs (NARX) is a recurrent dynamic network, with feedback connections enclosing several layers of the network [5,13]. The NARX model is based on the linear ARX model. The defining equation for the NARX model is:

$$y(t) = f(y(t-1), y(t-2), \ldots\ldots, y(t-n_y), u(t-1), u(t-2), \ldots\ldots, u(t-n_u)) \qquad (1)$$

Where the next value of the dependent output signal $y(t)$ is regressed on previous values of the output signal and previous values of an independent (exogenous) input signal. You can implement the NARX model by using a feed-forward neural network to approximate the function (f). A diagram of the resulting network is shown in Fig. (6):

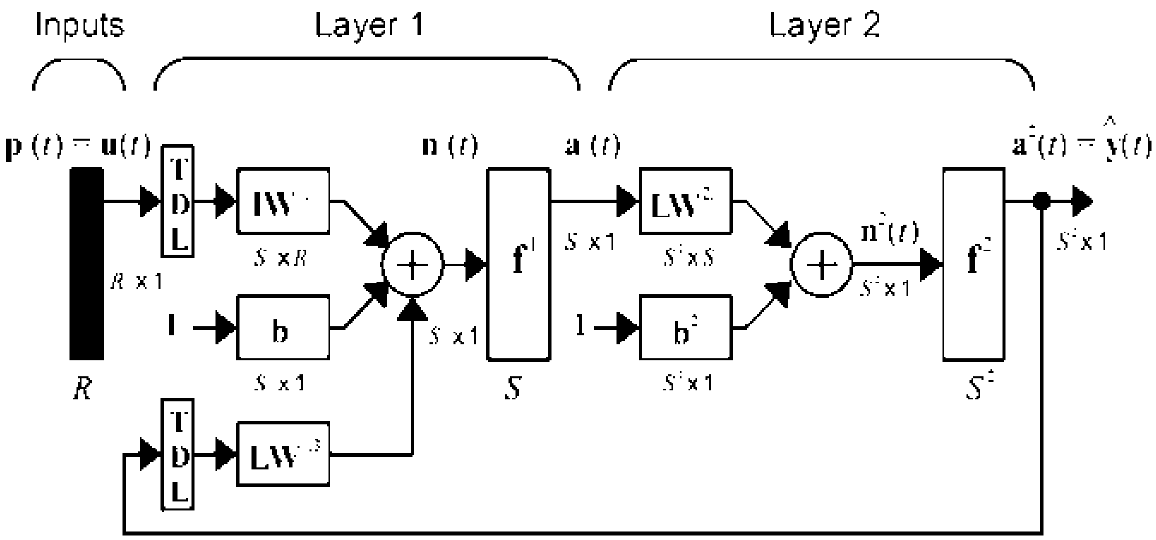


Fig. 6. Nonlinear autoregressive network with exogenous inputs (NARX) architecture.

where a two-layer feedforward network is used for the approximation. This implementation also allows for a vector ARX model, where the input and output can be multidimensional. You can consider the output of the NARX network to be an estimate of the output of some nonlinear dynamic system that you are trying to model. The output is fed back to the input of the feed-forward neural network as part of the standard NARX architecture, as shown in the Fig. (7a). because the true output is available during the training of the network, you could create a series-parallel architecture [5,14]], in which the true output is used instead of feeding back the estimated output, as shown in the Fig. (7b). This has two advantages. The first is that the input to the feed-forward network is more accurate. The second is that the resulting network has a purely feed-forward architecture, and backpropagation (BP) can be used for training [12].

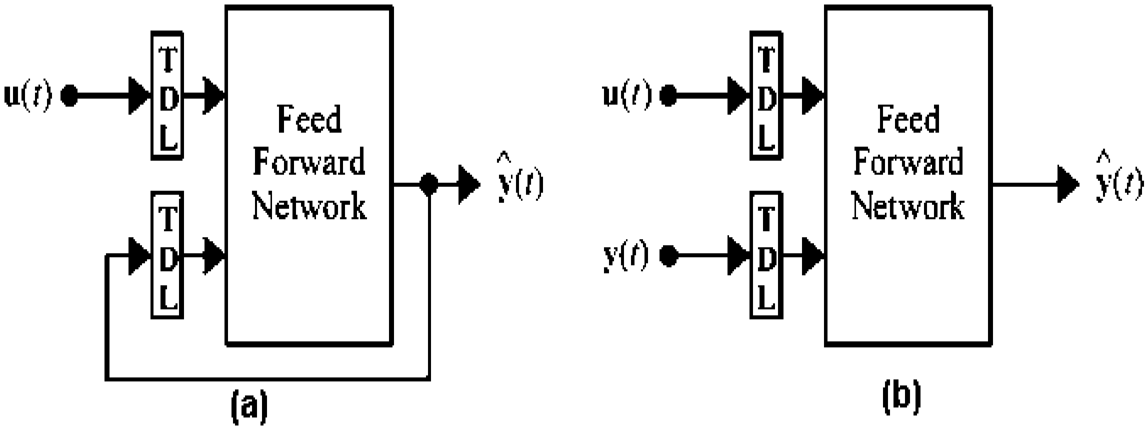


Fig. 7. (a) Parallel architecture, (b) Serial-parallel architecture

3. Learning Algorithm of DNN

There are some researches in the dynamic neural network (DNN). Such as the static ANN with integrator, NARX network, Jordan network [3]. The learning algorithm of the recurrent neural network (RNN) is usually the backpropagation (BP) algorithm. Because it tends to convergent to the local minimum and flexibility is not so good.

Genetic algorithm (GA) was proposed [15] which based on the certain kind of RNN. In order to compare the performance of the four DNNs, the Layer-Recurrent Network (LRN), Elman network, focused time-delay neural network (FTDNN), and NARX network in the nonlinear dynamic system identification, they are used for identification of the same nonlinear dynamic system based on the GA. It illustrates the advantages and disadvantages of the dynamic neural networks. In order to avoid the disadvantage of BP, the genetic algorithm (GA) is adopted to train the weights of four kinds of DNNs.

- *Code*: It is the key step to design the coding of the GA because, it directly affects the choice, crossover and mutation. Due to the large number weights of the ANN, the real-number coding is used in this paper.
- *Fitness function*: the fitness function is confirm by the target function and is standard to distinguish the individual worth in the hole population. At same time the fitness function value is always positive. In any cases the bigger this value is, the better it is as follows:

$$f = \frac{1}{2}\sum_{t=1}^{p} Y(t) - y(t))^2 \qquad (2)$$

Where $Y(t)$ is the output sample, and $y(t)$ is the output of ANN, p is the sum of the sample. The fitness function in GA is shown as follows:

$$F = 1/f \qquad (3)$$

- *Elitist GA*: the elitist is a kind of improved GA. A new generation population is produced by crossover and mutation for the N dyads in the standard GA (SGA). But for the elitist GA, A new generation population is produced by crossover and mutation for the $N-1$ dyads and the biggest fitness value individual is selected in population as the last individual in new generation. The monotony of the population fitness value is kept by this modify. The convergence is very important in the theory for the algorithm. After adopting the elitist GA, the GA's convergence is proved in real number coding. So the elitist GA is used in this paper.
- *Selection*: it is roulette selection in the elitist GA.
- *Crossover*: the heuristic crossover operator can confirm the search direction of the genetic by the means of the value of the target functions. The next generation x will be birth by the parent $x1$ and $x2$. The formula is as follows:

$$X = r * (x2 - x1) + x2 \qquad (4)$$

where r is a random number between one and zero.

- *Mutation*: the Gauss mutation is adopted that can search local area of the indivtual's neighborhood. The main parameters are selected as follows:

Population size: 40
Crossover rate: 0.85
Mutation rate: 0.1

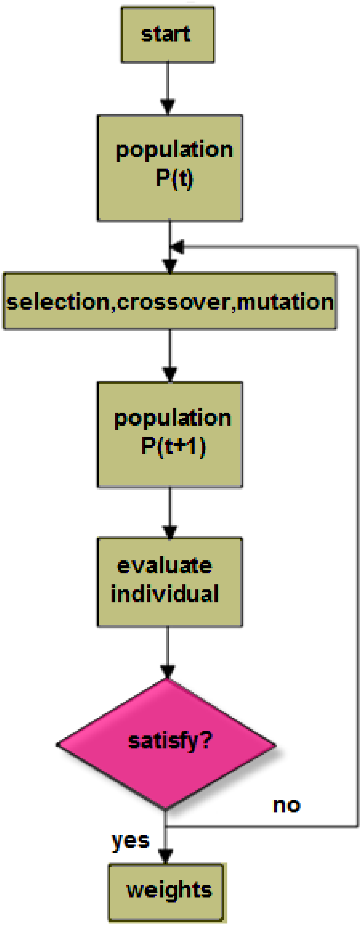


Fig. 8. Sketch map of genetic algorithm (GA).

Elite reservation rate: 0.1

The stop criteria of the algorithm is tags the optimization solution is not evolution after 30 generation. The sketch map is shown in Fig.(8).

5. Simulation Results of DNN

In this simulation, the nonlinear dynamic plant to be identified is expressed as follows:

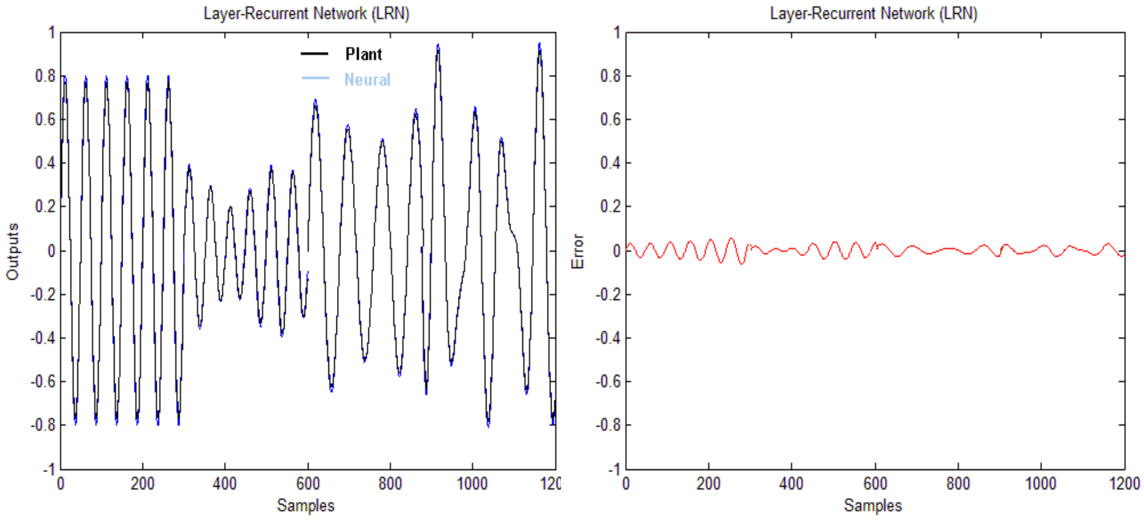
$$y(k) = 0.4y(k-1) + u(k-1)/(1+u^2(k-1)) \qquad (5)$$

The train signal is the random sequence which is submitted to the uniform range between 1 and -1, and the test input chosen:

$$f(k) = \begin{cases} 0.8sin\frac{\pi k}{25} & 0 < k \le 300 \\ 0.3sin\frac{\pi k}{25} + 0.1sin\frac{\pi k}{32} & 300 < k \le 600 \\ 0.1sin\frac{\pi k}{32} + 0.6sin\frac{\pi k}{40} & 600 < k \le 900 \\ 0.3sin\frac{\pi k}{25} + 0.1sin\frac{\pi k}{32} + 0.6sin\frac{\pi k}{40} & 900 < k \le 1200 \end{cases} \qquad (6)$$

The data was collected at a sampling interval of 1.0 seconds to for four time series. The result is displayed in the following plot. You can see that the errors are very small.

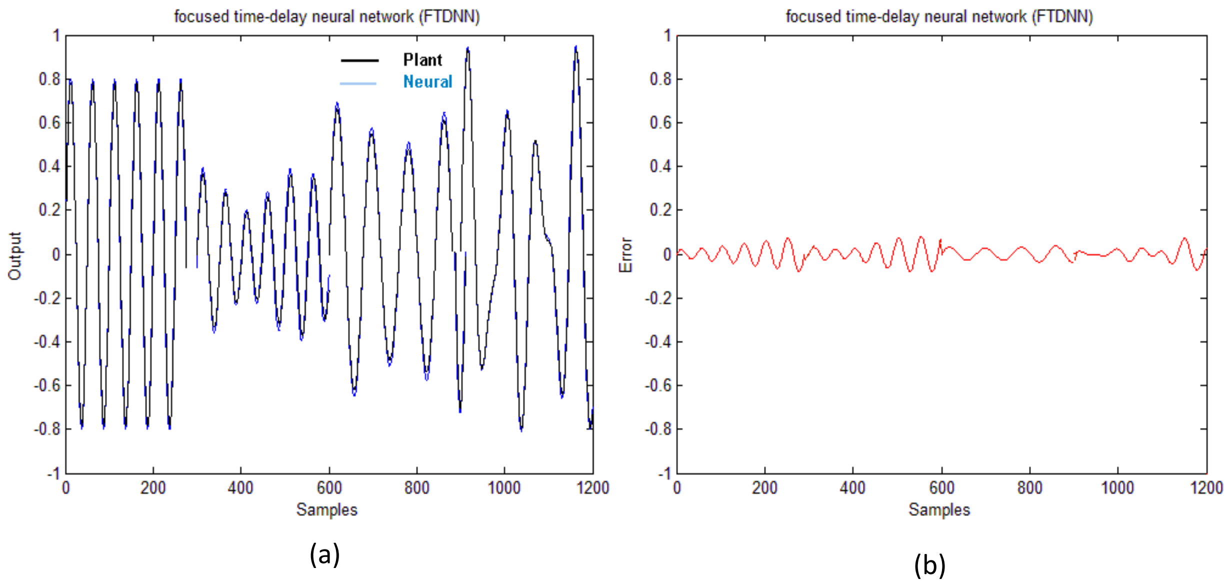
The structure of LRN is decided by modeling. There is one node in the input layer, five nodes in the hidden layer, and one node in the output layer in this research. The simulation figure of the testing is shown in Fig.(9):



(a) (b)

Fig. 9. (a) Output of the plant, and identification model, (b) error curve, for LRN.

For the focused time-delay neural network (FTDNN), there is one node in the input layer, five nodes in the hidden layer, and one node in the output layer. The simulation figure of the testing is shown in Fig.(10):

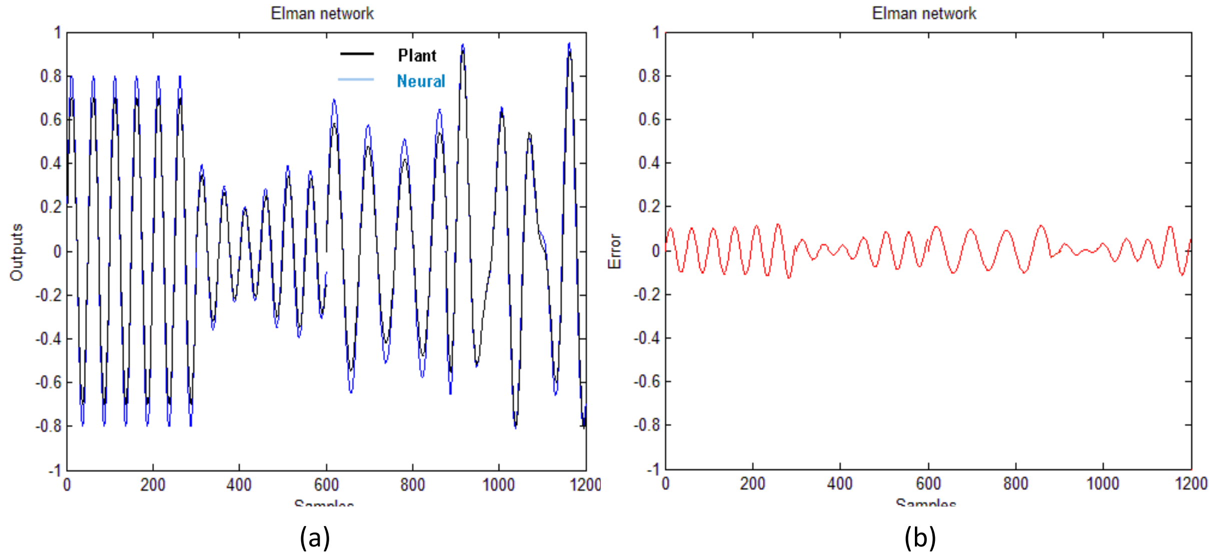


(a)

(b)

Fig. 10. (a) Output of the plant, and identification model, (b) error curve, for FTDNN.

The structure of Elman Network is the same of the LRN, and the simulation figure of the testing is shown in Fig.(11):

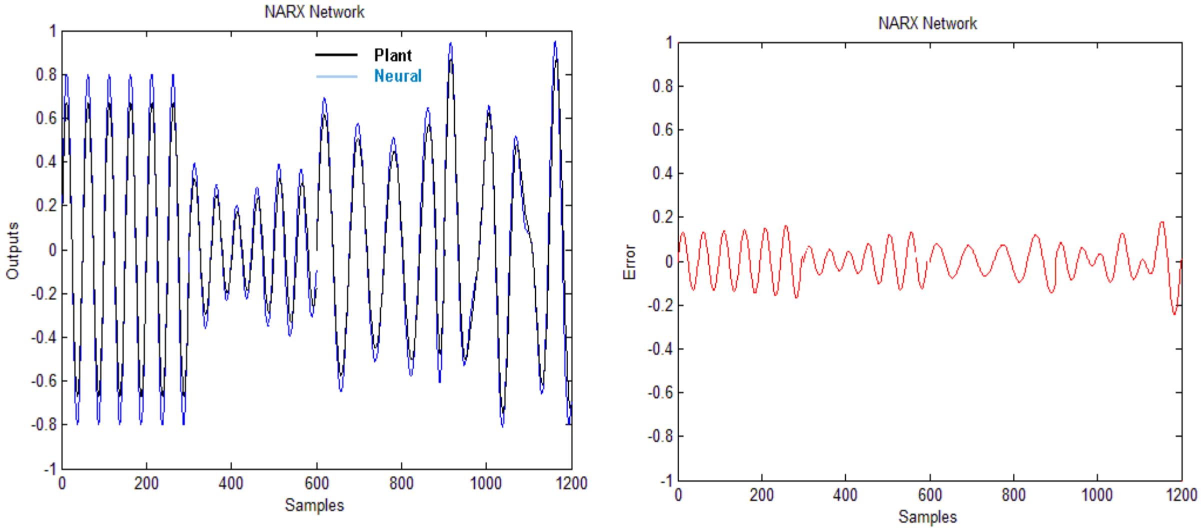


(a)

(b)

Fig. 11. (a) Output of the plant, and identification model, (b) error curve, for Elman Network.

For the NARX network , there are two nodes, $u(t)$, and $y(t)$ in the input layer, five nodes in the hidden layer and one node in the output layer. The simulation result of the testing is shown in Fig.(9)



(a)

(b)

Fig. 12. (a)Output of the plant, and identification model, (b) error curve, for NARX network.

6. Results Discussion

In all four cases, the normalized MSE, between the desired output and the output from the network, decreases, while learning, to mean squared error (MSE) values between 0.01 and 0.005. If when tested, the error increases in an important way then more coefficients were added and also new training signals were used. In Fig 9 to Fig 12 the outputs from the real system (black trace) and the outputs from the trained ANN (blue trace) are compared. In table 1 , it can be known that all four kinds of DNNs can provide the high precesion of model of nonlinear dynamic system after the training based on the GA. When the test signal is used to test the model, there are different test error in these DNNs. The test errors of the LRN and FTDNN networks are the smaller. But they are bigger in the Elman and NARX neural networks. The train time of LRN is the shortest.

Table 1. Performance Comparison of the four DNNs

Dynamic NN	MSE	Learning Time	Error Value
Layer-Recurrent Network (LRN)	0.006	59 sec	0.12
focused time-delay neural network (FTDNN)	0.005	74 sec	0.27
Elman network	0.009	109 sec	0.32
NARX network	0.01	137 sec	0.39

7. Conclusion

The use of DNNs have shown encouraging results, and been useful for the identification of nonlinear dynamic systems. As classical methodologies, our structure allows the extraction of a mathematical model of the system, but it adds the advantages of GA with neural networks to learn and generalize. The training time of the four DNN is shorter and the convergence speed is faster. At the same time have high precision identification. Although the training error of the LRN network is the smallest among the four DNNs , the generalization ability of the Elman network and the NARX network are better than the FTDNN , and the LRN networks. The LRN, the NARX and the Elman network can depict the complex dynamic system due to the back-forward in the DNN. So the order of the system need not to be set in advance. Because the FTDNN network is the feed-forward multilayer NN. The order of the system will be set before the training.

References

[1] N. Homma, L. Jin and M. M. Gupta, Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory, Wiley, John and Sons, Inc., 2002.

[2] Paulo S.R. Diniz, "Adaptive filtering algorithms and practical implementation", Third Edition, Springer, 2008.

[3] M. M. Gupta, L. Jin, and N. Homman, Static and Dynamic Neural Networks, John Wiley &Sons, Hookien, New Jersey, U.S., 2003.

[4] K. S Narendra, K. Parthasarathy. “Identification and control of dynamical systems using neural”[J]. IEEE Transactions on Neural Networks, 1990,1(1):4-26.

[5] Howard Demuth, Mark Beale, Martin Hagan(2006)," Neural Network Toolbox for use with MATLAB ", Version 5,Math works Inc.

[6] Elman, J.L., “Finding structure in time,” Cognitive Science, Vol. 14, 1990, pp. 179–211.

[7] Hopfield J J. “Neural networks and physical systems with emergent collective computational abilities” [A]. Proceedings of the Academy of Science[C], U S,1982,79: 2554-2558

[8] Jordan M I. “Supervise leaning and systems with excess degrees of freedom”[R]. Massachusetts Institute of Technology: COINS Technical Report, 1988. 88-27.

[9] Pham D T, Liu X. “Dynamic system identification using partially recurrent neural networks”. Jounal of Systems Engineers. 1992:2(2):90-97

[10] Ku C C, Lee K Y. “Diagonal recurrent neural networks for dynamic systems control”[J]. IEEE Transactions on Neural Networks. 1995,6(1):144-155.

[11] Yamada, T. and Yabuta T. “Dynamic System Identification Using Neural Networks. IEEE Trans on Systems, Man, and Cybernetics”. January/February (1993), 204-211.

[12] H. Garnier and P. Young, “Time-domain Approaches to Continuous-time Model identification of Dynamic System From Sampled Data”, in Proc. American Control Conference, Boston, MA, 2004, pp. 668-672.

[13] X. M. Ren, A. B. Rad, P. T. Chan, and Wai Lun Lo, “Identification and control of cntinuous-time nonlinear systems via dynamic neural networks,” IEEE Trans. Indus. Elec., vol. 50, no. 3, pp. 478 - 486, June 2003.

[14] Yamada, T. and Yabuta T. (1993) Dynamic System Identification Using Neural Networks. IEEE Trans. on Systems, Man, and Cybernetics. January/February, 204-211.

[15] Goldberg. A. Genetic Algorithms in Search and Optimization. Adison Wesley. 1993.

The work was carried out at the college of Engineering. University of Mosul