AJIT-e: Online Academic Journal of Information Technology
2019 Bahar/Spring – Cilt/Vol: 10 - Sayı/Num: 37
DOI: 10.5824/1309-1581.2019.2.002.x
http://www.ajit-e.org/?menu=pages&p=details_of_article&id=431

Received: 28.05.2019 Editorial Process Begin: 28.05.2019 Published: 15.06.2019

Comparison of Feedforward and Recurrent Neural Network in Forecasting Chaotic Dynamical System

Engin KANDIRAN, Yeditepe Üniversitesi, Ticari Bilimler Fakültesi, İstanbul-TÜRKİYE, *engin.kandiran@yeditepe.edu.tr, https://orcid.org/0000-0002-6171-1346*

Avadis HACINLIYAN, *Yeditepe Üniversitesi, Fen-Edebiyat Fakültesi, Fizik Bölümü, İstanbul-TÜRKİYE, ahacinliyan@yeditepe.edu.tr, https://orcid.org/0000-0002-4667-9659*

ABSTRACT

Artificial neural networks are commonly accepted as a very successful tool for global function approximation. Because of this reason, they are considered as a good approach to forecasting chaotic time series in many studies. For a given time series, the Lyapunov exponent is a good parameter to characterize the series as chaotic or not. In this study, we use three different neural network architectures to test capabilities of the neural network in forecasting time series generated from different dynamical systems. In addition to forecasting time series, using the feedforward neural network with single hidden layer, Lyapunov exponents of the studied systems are forecasted.

Keywords: *Chaos, Artificial Neural Networks, Lyapunov Exponents, time series forecasting*

ÖZ

Yapay sinir ağları global fonksiyon tahmininde genel olarak kabul görmüş başarılı bir araçtır. Bu nedenle, kaotik zaman serisi tahmininde şarılı bir yaklaşım olarak bir çok çalışmanın konusu olmuşlardır. Verilen bir zaman serisi için, Lyapunov üsteli, verilen serinin kaotik olup olmadığını karakterize etmekte iyi bir parametredir. Bu çalışmada, üç değişik yapay sinir ağı mimarisini, değişik dinamik sistemlerden türetilmiş zaman serilerini tahmin etmede kullandık. Zaman serilerinin tahminine ek olarak, çalışılan sistemlerin Lyapunov üstellerini tek gizli katmanlı ileribeslemelisinir ağı kullanarak tahmin ettik.

Anahtar Kelimeler: *Kaos, Yapay Sinir Ağları, Lyapunov üsteli, zaman serisi tahmini.*

1. Introduction

In many scientific study fields, estimation of future values of a system of interest is a very important topic to understand its dynamical evolution. The physical systems, which are governed by a set of equations of motion, are commonly investigated by integrating them forward in time and thus the evolution of the system is studied. In chaos theory, for a given time series originating from a dynamical system, it is important to understand that the time series is chaotic or not which means that there is a sensitive dependence on initial conditions or not. One of the most commonly accepted parameters that characterize the existence of chaos in given time series is its Lyapunov exponent which gives the rate of convergence (or divergence) of nearby trajectories in state space. For a time, series, positive Lyapunov exponent is indicator of chaos. In addition to this, Lyapunov exponents are used to calculate important measures of a dynamical system. Dimension of the attractor is calculated by the

Kaplan-Yorke conjecture using Lyapunov exponents: Kolmogorov-Sinai entropy can be related to Lyapunov Exponents as well.

However, in most cases, governing equations of motion for a given time series is unknown, such as stock market indices in economics. To forecast chaotic time series, there are many methods other than neural networks such as Taylor series expansion, radial basis functions, and nonparametric kernel regressions. These methods are based on interpolation and approximation of unknown function by use of scattered data points. In Taylor series expansion approach the main disadvantage is rapidly increasing order of expansion and according to Casdagli (Casdagli, 1989), there is no guaranteed order of convergence for dimension n>1, and higher polynomial degrees has a wide oscillation tendency. Nonparametric kernel regression is a method that depends on estimating probability density function from observed time series, but it has some drawbacks mentioned in (Gencay & Tung, Nonlinear modelling and prediction with feedforward and recurrent networks, 1997). On the other hand, in literature, there are many studies which demonstrate the capabilities of neural networks in forecasting chaotic time series. For example, two-layered feedforward neural networks are used in (de Oliveira, Vanucci, & da Silva, 2000), it gives promising results in estimation of chaotic time series generated from the Lorenz system, Henon and Logistic maps. In another study (Gencay,1994), Gencay demonstrates that nonlinear noisy time series can be modelled quite accurately by single-layer feedforward neural networks.

For a given sets of dynamical system, Lyapunov exponents can be calculated using a Jacobian algorithm (Wolf, Swift, Swinney, & Vastano, 1985).. That's why, numerical methods are studied to find Lyapunov exponents based on Taken's embedding theorem (Takens, 1980) such as the Ruelle-Eckman algorithm (Eckman & Ruelle, 1985).

In this study, we use three different neural network architectures to test capabilities of neural networks in forecasting time series generated from different dynamical systems. This study is organized as follows: in the second section, we give summary of neural networks' working mechanism and define feedforward and recurrent networks. In the third section, chaotic systems that we use in this study to test performance ANNs are defined. In the fourth section, we present the performance of each network in terms of forecasting. In addition to forecasting time series, using the feedforward neural network with single hidden layer, Lyapunov exponents of the studied systems are calculated with the algorithm developed in (Gencay,1995). In the last section, the final discussion is given.

2. Artificial Neural Networks

Neural Networks are the mathematical model which tries to imitate the working mechanism of neurons in our brain. Earliest form of neural networks, called perceptrons were developed in the 1950s and 1960s by the scientist Frank Rosenblatt. Yet, they do not share completely same mechanism with biological neurons. They are very commonly used tools in many science fields: such as image recognition, classification, time series forecasting, pattern recognition etc...

Working mechanism of artificial neuron is actually very simple. To understand this mechanism, we need to define basic components of artificial neuron: Input weights and biases are Wij and bj. Weigths Wij define rate of effect for input xi from neuron i to neuron j. Biase value bj allows you to shift the activation function to the left or right, which may be critical for successful learning. Another important parameter of artificial neuron is the activation function Ψ. Activation function has the role of threshold in biological neuron.

Networks are composed of neurons which are stored in layers li. The first layer that the data enters is called the input layer. The layer that the prediction or result is given is called the output layer. The layer(s) that actual computation or approximation occur is called as hidden layer(s). In a layer li, there can be one or many neurons which are connected neurons in li-1 and li+1 layers.

Artificial neurons create an output by the following way:

Suppose n input values xi are coming to neuron j in lth layer from (l-1)th layer. Then in a neuron all inputs are summed with following way:

$$y_{j,t} = b_j + \sum_{i=1}^{n} W_{ij} x_i \tag{1}$$

Then, yj,t passes through the activation function Ψj and output of jth is $\hat{o}_j$:

$$\hat{o}_j = \Psi(y_{j,t}) \tag{2}$$

Where wij, bj are learned or estimated parameters through learning process in learning cycles or epochs. At this step it is important to mention about the role of the activation function. Purpose of the activation function is to convert the input signal of a node, which is then used as an input in the next layer. Activation function should be applied, otherwise the output will be linear function. Since neural networks are used to work with nonlinear or complicated data sets, such as images, videos, audio, speech, time series, nonlinear activation functions are used. The most commonly used are: sigmoid, hyperbolic tangent, ReLu - Rectified linear units.

Another important feature of artificial neural networks is their ability of learning. In general, there are two general types of learning algorithms: supervised and unsupervised learning. Since we are interested in time series forecasting, we use supervised learning. Role of weights and biases are already discussed previously and learning in neural network is related to updating connection weights and biases in neuron. That's why after many training weights and biases are updated according to these weights and biases in neurons that's why network gives true prediction for given input values. But the question is how to update the weights and biases.

The answer is Error (or Cost) Function E:

$$E(y, \hat{o}) = \frac{1}{n} \sum_{i=1}^{n} (y - \hat{o}_i)^2 \tag{3}$$

Where y is target variable. Then the weights are adjusted by the following way:

$$W_{ij}^{new} = W_{ij}^{old} - \alpha \frac{\partial E}{\partial w_{ij}} \tag{4}$$

Where $0 < \alpha < 1$ learning rate which is generally chosen as 0.01. At this point, it is understood that activation function should be differentiable. This learning schema is called as backpropagation. The significance of backpropagation is that it enables us to simultaneously compute all the partial derivatives ∂E/∂wij in just one pass so the total cost of backpropagation is roughly the same as making just two forward passes through the network.

The learning rate is a relatively small constant that indicates the relative change in weights and biases if the learning rate is too low, the network will learn very slowly. If the learning rate is too high, the network may oscillate around overshooting the lowest point with each weight adjustment, but never actually reaching it. Some modifications to the backpropagation algorithm allow the learning rate to decrease from a large value during the learning process.

In this study, two types of neural networks are used: Feedforward Neural Network and Recurrent Neural Network. Feedforward neural networks are the networks where connections between neurons in layers do not form a cycle. Which means the input propagates only in the forward direction (from input layer to output layer). If the network composed of more than one hidden layer they are called as multilayer feedforward neural networks (multilayer perceptrons). When feedforward neural networks are extended to include feedback connections, they are called recurrent neural networks. Since neuron's in layer has self-connection, they are considered as networks with a memory. Schematic diagram of both network types is given in Fig.1.

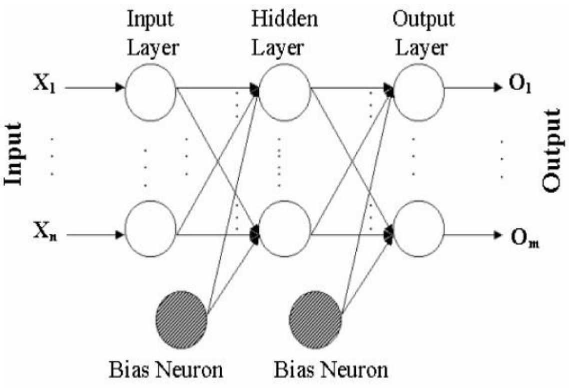

Figure 1.a: Schematic diagram of feedforward neural Network

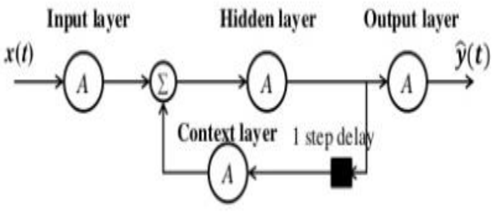

Figure 1.b: Schematic diagram of recurrent neural network

There are some tricky points about neural networks. There is no general solution to decide how many hidden layers and how many neurons in networks in deep learning community.

But in general, to approximate more complicated dynamics, we need more hidden layers. Although, we mentioned about backpropagation learning algorithm, there many variants of it, so choice of learning algorithm is not unique and depends on the problem on hand.

3. Chaotic Dynamical Systems

In order to compare the capability of neural networks in forecasting chaotic time series, two systems are used: Duffing oscillator, Rössler System. In the following subsections, the mentioned systems are defined:

3.1. Duffing Oscillator

The chaotic dynamics of Duffing oscillator has been studied in many works in literature. The Duffing equation that is used in this study is:

$$\ddot{x} + \delta\dot{x} + \beta x + \alpha x^3 = \gamma \sin(wt) \tag{5}$$

The given system displays chaotic behavior with for parameter values $\gamma = 0.42, \delta = 0.5, \alpha = -1, \beta = 1, w = 1$ and initial conditions (x0, y0)=(0.5021,0.17606) and trajectory of the system is given in the Fig. 1.

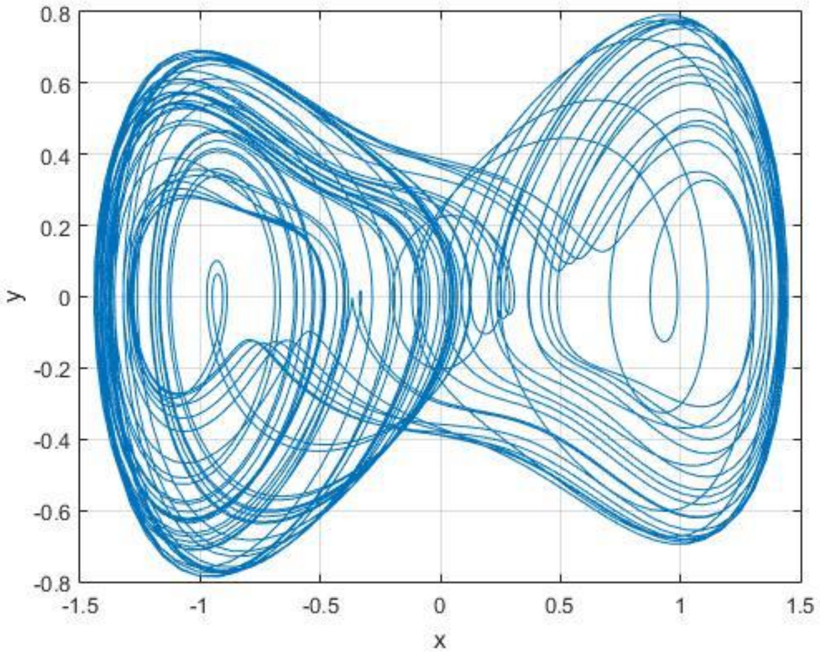

Figure 2: Duffing oscillator with chaotic behavior

3.2. Rössler System

The second system we are interested in is the Rössler system or attractor:

$$\frac{dx}{dt} = -y - z$$

$$\frac{dy}{dt} = x + ay \tag{6}$$

$$\frac{dz}{dt} = b + z(x - c)$$

Where a, b, c are real constant parameters of the system. For a = b = 0.2, c = 5.7, the system has corresponding Lyapunov exponents: (0.0714, 0, -5.3943). Since there exists a positive Lyapunov exponent the system displays chaotic behavior. (see Fig. 3)

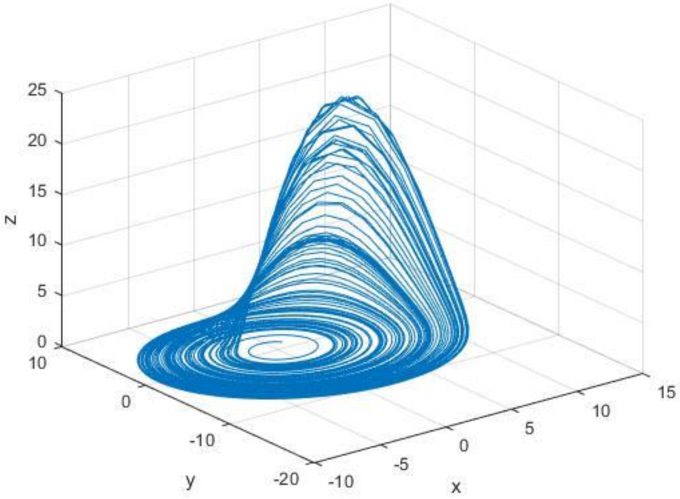

Figure 3: Rössler Attractor for given set of initial conditions and parameters

4. Prediction of Chaotic Time Series with Neural Networks

The systems that we are going to use for comparing the capability of neural networks in forecasting chaotic time series were defined in the third section, In this section, we compare the performance of three different neural networks: multilayer feedforward neural network, single layer feedforward and single layer recurrent neural network.

In this study, we have used firstly, feedforward neural network with one hidden layer with m number of input units. Duffing oscillator is two-dimensional system (d=2). According to Taken's theorem, an embedding dimension must be less than or equal to 2d+1. We choose embedding dimension m=4. Then we use multilayer feedforward neural with the following architecture (m: 2m: m: 1) which is the same architecture used in (de Oliveira, Vannucci, &

da Silva, 2000). As a third architecture we use Elman's recurrent neural network with one hidden layer.

In our simulations, performances of the three mentioned neural networks are compared using data with and without additive noise. For performance comparison of neural network without noise case we let $y_t = x_t$ where y_t is our target variable. For the noisy data generation, we add noise to our data in the following way:

$$y_t = x_t + u_t \tag{7}$$

Where ut represents the noise component and $u_t = \eta\sigma g_t$ where σ is sample standard deviation of xt and η is noise level which takes different values between $0 < \eta < 1$and gt is standard normal random variable.

In the result, for forecast performance of each network root-mean-square error (rmse) is used as performance criteria. Using the multilayer feedforward neural network (MFN) with architecture (4:8:4:1), the neural network predicts the actual time series perfectly with rmse=3.003x10-5 for Duffing oscillator (without noise) and predicted and original time series are plotted in Fig. 4.a. For Duffing system, we use the MFN with architecture (4:8:4:1) and for without noise, the network is very efficient. In Fig 4.b, predicted and original time series are for Rössler system without noise case is given.

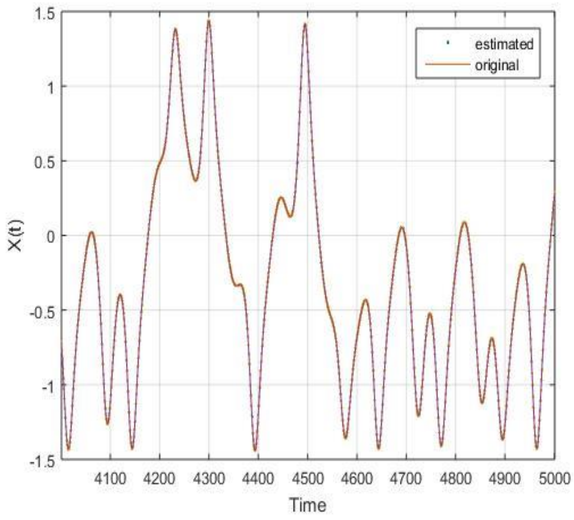

Figure 4.a: Prediction of Duffing oscillator time series without noise for MFFN(4:8:4:1)

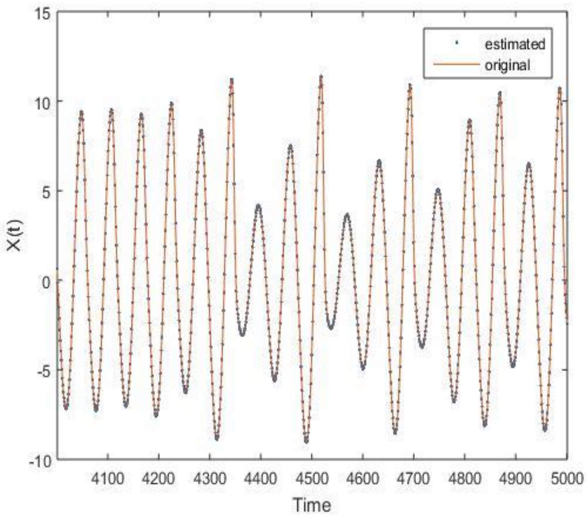

Figure 4.b: Prediction of Rössler time series without noise for MFN(6:12:6:1)

For the same data set we have used Elman's recurrent neural net in predicting Duffing oscillator, which is very similar to single hidden layer feedforward neural network, but it also uses previous estimation value as an input which makes it a neural network with a memory. For single hidden layer we try to choose optimal number of neurons. We choose optimum number of neurons as 8 which is two times the embedding dimension m=4. For forecasting result, Elman network display performance as good as multilayer feedforward neural network. Elman's neural network gives rmse=3.349x10-5. Estimation of Elman's neural network and original data is plotted in Fig. 5. It is also important to note that with the given architectures of networks, Elman's network gives results for smaller number of epochs than the multilayer feedforward one.

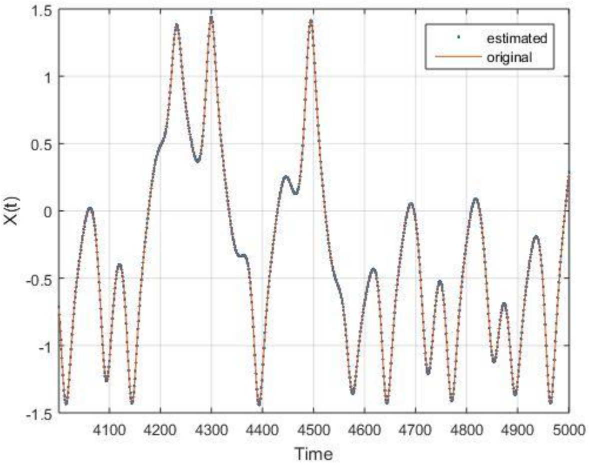

Figure 5: Elman's neural network prediction of Duffing oscillator

Finally, for the Duffing oscillator without noise, we test the performance of feedforward neural network with single hidden layer. For the single hidden layer, number of neurons is chosen as again 2m which is same as Elman's recurrent network. With single hidden layer neural network, performance of forecast is slightly increasing with compared to recurrent one with rmse=3.328x10-5 but it is worse than the multilayer neural network. In addition, the speed of convergence of single hidden layer feedforward neural network is higher than the recurrent one.

After forecasting the Duffing data set without noise, we test the noise filtering capabilities of these three types of neural networks. For η=0.01, multilayer neural network, forecast performance is good but as it is expected it is worse than without noise (with rmse=0.059). Estimated and original data is plotted in Fig. 6. For η=0.1, performance of multilayer feedforward decrease as expected compared to previous case (Fig. 7). It gives rmse=0.5755. For η=0.2, rmse increases to 0.765(see Fig. 8).

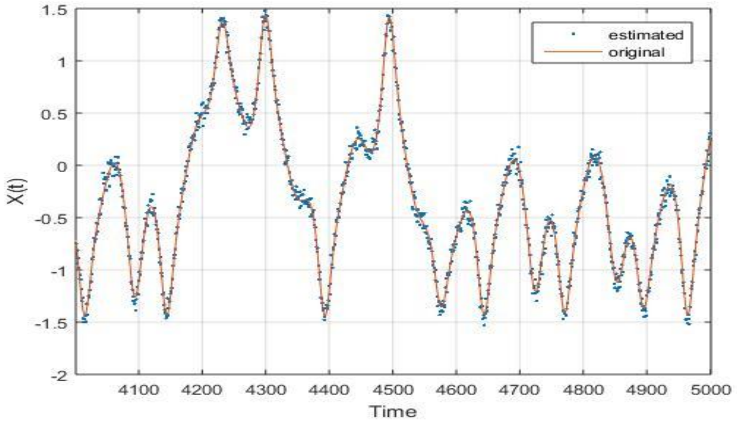

Figure 6: Duffing system prediction of MFFN with error rate η=0.01

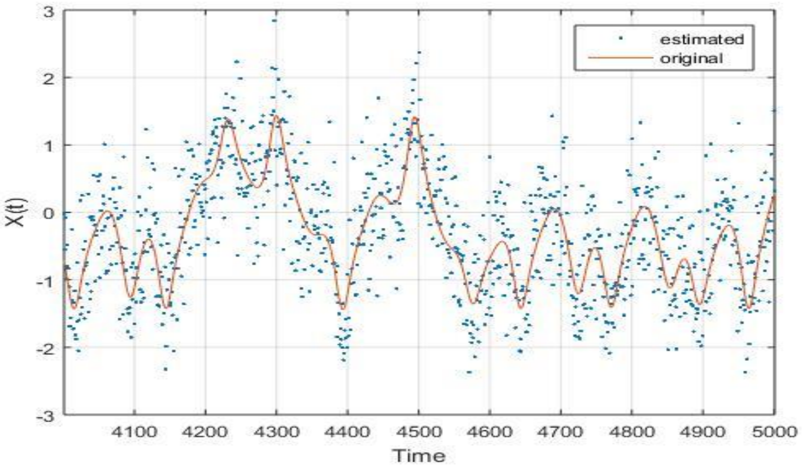

Figure 7: Duffing system prediction of MFFN with error rate η=0.1

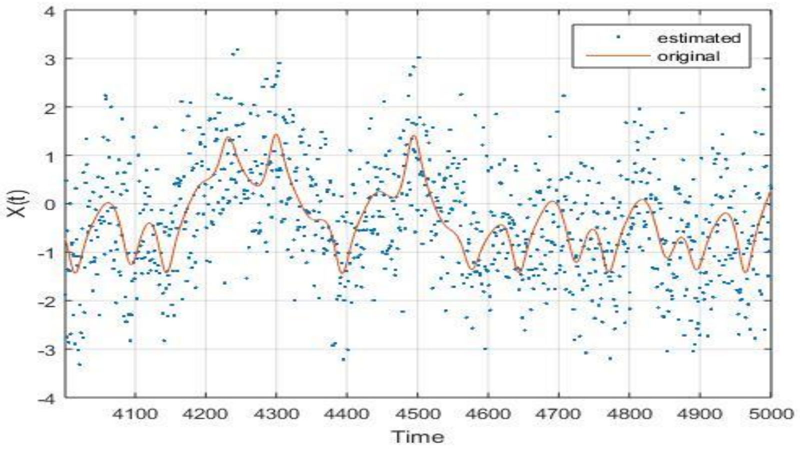

Figure 8: Duffing system prediction of MFFN with error rate η=0.2

To increase the forecasting performance of MFFNN, we increase neuron numbers in layer from 4 to 5. With this change, we train the network and make prediction for data with error rate η=0.2. Neural network gives better estimate with rmse=0.6635 and forecast of modified MFFNN is plotted in Fig. 9.

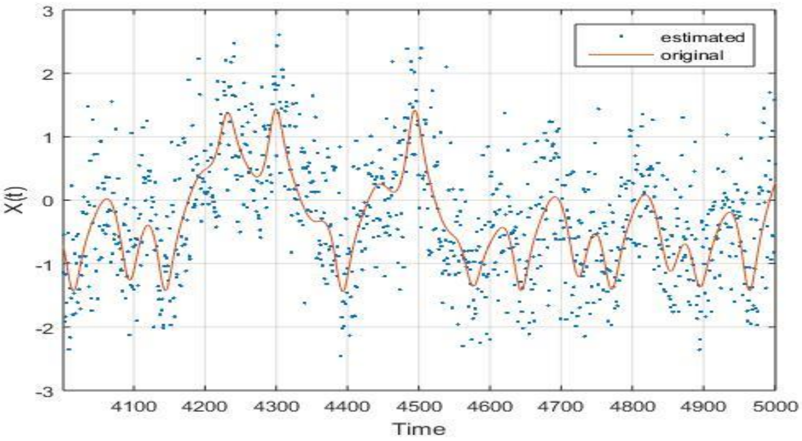

Figure 2: Duffing system prediction of MFFN with error rate η=0.2 where network with network architecture (5:10:5:1)

After analyzing noise filtering performance of MFN, we apply same tests to Elman type recurrent network. In (Gencay &Tung, 1997), they demonstrated that recurrent neural networks are more accurate than single layer feedforward neural networks. In our analysis, we observe that for η=0.01, it performs a little bit better than MFN but there is no significant improvement (with rmse=0.053). When we increase the error rate to η=0.1, performance of recurrent network is again close to MFN (with rmse=0.5498). For η=0.2, performance of recurrent network clearly better than MFN with rmse=0.6487. Prediction of recurrent network with error rate η=0.2, is plotted in Fig. 10.

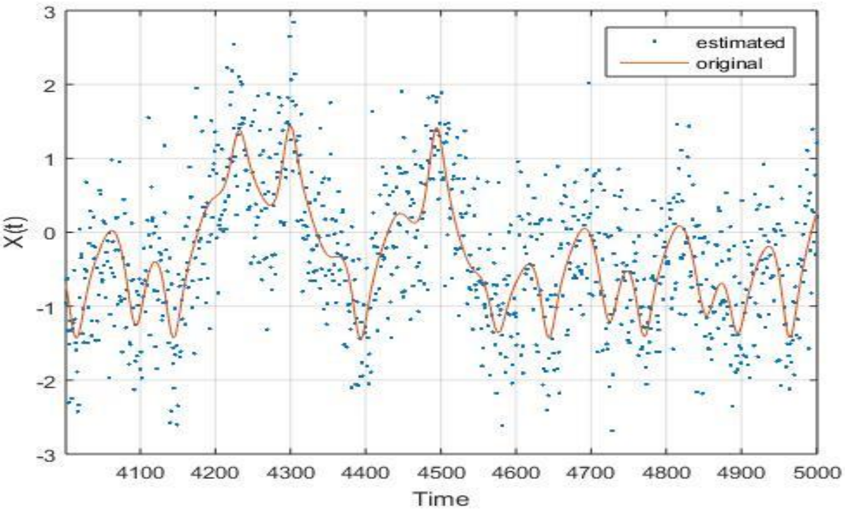

Figure 10: Duffing system prediction of recurrent network with error rate η=0.2

Single layer feedforward network, for time series with error rate η=0.01, performs as well as both of the other two neural nets. For η=0.1, performance of this network is worse than both recurrent and multilayer feedforward one. Finally, for η=0.2, forecast performance is getting worse than previous cases as it is expected. In Table 2, *rmse* values are given for prediction performance of each neural network with and without noise.

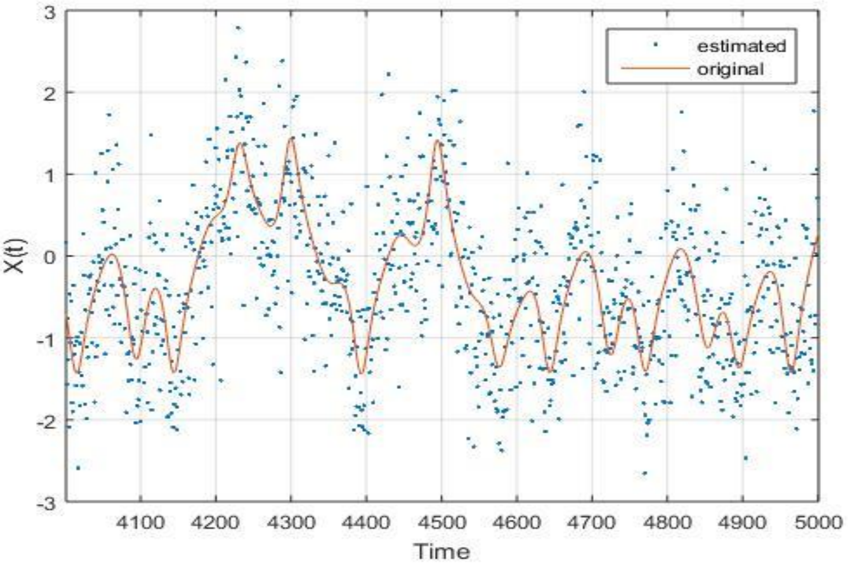

Figure 3: Duffing system prediction of single layer feedforward network with error rate η=0.2

It can be seen in previous analysis results, as it is mentioned in (Gencay &Tung, 1997), RN is better predictor model than FN in noisy time series. For higher noisy level (η=0.2) interestingly performance of MFN is not as good as RN and FN. In Table 1, *rmse* values of each prediction for Duffing system is given.

For the Rössler system, we test each network architecture. In summary Rössler system also can be estimated very efficiently by described network architecture in Table 2. Similar to Duffing oscillator, when we increase the noise level, prediction error rates increase. In Table 2, *rmse* error values for each prediction of Rössler system are given.

As final numerical study, we test the performance of one hidden layer feedforward neural network in estimating Lyapunov exponents. In general, if the system has a positive Lyapunov exponent, the system can be classified as chaotic. To test neural networks for estimation of Lyapunov exponents of chaotic time series, R.Gencay (Gencay & Dechert, 1992) developed an algorithm which is mainly based on the Ruelle-Eckman algorithm (Eckmann, 1987). In his study, they test many chaotic systems such as Lorenz system, they obtain great accuracy in their estimation. In this study, we extend their test by testing their routines using Rössler and Duffing oscillator. In Table 3, network estimates for Rössler and Duffing oscillator is given. As it can be seen, single layer neural network is very efficient in esittmation of Lyapuonv exponents.

Table 1: rmse values of each network with respect to changing error rate η for Duffing oscillator

Network Model	*rmse*
η =0 (no noise)	

MFN(4:8:4:1)	3.003×10^{-5}
RN(8)	3.349×10^{-5}
FN(8)	3.328×10^{-5}
η =0.01	
MFN(4:8:4:1)	0.059
RN(8)	0.053
FN(8)	0.058
	η =0.1
MFN(4:8:4:1)	0.5755
RN(8)	0.5498
FN(8)	0.5997
η =0.2	
MFN(4:8:4:1)	0.7650
RN(8)	0.6487
FN(8)	0.6766

Table 2: rmse values of each network with respect to changing error rate η for Rössler System

Network Model	*rmse*
η =0 (no noise)	
MFN(6:12:6:1)	3.3625×10^{-5}
RN(12)	3.399×10^{-5}
FN(12)	3.388×10^{-5}
η =0.01	
MFN(6:12:6:1)	0.048
RN(8)	0.053
FN(8)	0.054
η =0.1	
MFN(4:8:4:1)	0.8755
RN(8)	0.7498
FN(8)	0.8997
η =0.2	
MFN(4:8:4:1)	0.9650
RN(8)	0.8487
FN(8)	0.9887

Table 3: Lyapunov Exponent prediction values by feedforward neural network

Network	Rössler System	
	Lyapunov exponents	Estimated exponents
FN(6)	(0.0714, 0, -5.3943)	(0.0796,0.0021,-5.5223)
	Duffing Oscillator	
	Lyapunov exponents	Estimated exponents
FN(8)	(0.3675, 0.4256)	(0.3254, 0.4613)

5. Conclusion

In this study, we compare and test the performance of three neural network architecture for prediction of chaotic time series generated form dynamical system. We observe very simple network structures are very efficient estimation of nonlinear phenomena compared to other well-known technics such least square estimates. According to our observations, their success originated from their ability in nonlinear function estimation. They are also very trustful tool prediction of chaotic parameters such as Lyapunov exponents as we reported.

References

1. Casdagli, M. (1989). Nonlinear prediction of chaoti time series. *Physica D*, 335-356.
2. de Oliveira, K. A., Vanucci, A., & da Silva, E. C. (2000). Using artificial neural networks to forecast chaotic time series. *Physica A*, 393-404.
3. de Oliveira, K., Vannucci, A., & da Silva, E. C. (2000). Using artificial neural networks to forecast chaotic time series. (284).
4. Eckman, J., & Ruelle, D. (1985). Ergodic theory of chaos and strange attractors. *Rev Mod Phys*.
5. Eckmann, J.-P. &. (1987). Liapunov exponents from time series. *Physical review. A*, 4971-4979.
6. Gencay, R., & Dechert, D. W. (1992). An algorithm for the n Lyapunov exponents of an n-dimensional unknown dynamical system. *Physica D*, 142-157.
7. Gencay, R., & Tung, L. (1997). Nonlinear modelling and prediction with feedforward and recurrent networks. *Physica D: Nonlinear Phenomena*, 119-134.
8. Takens, F. (1980). Detecting strange attractors in turbulance. *Dynamical Systems and turbulance*.

9. Wolf, A., Swift, J., Swinney, H., & Vastano, J. (1985). Determining Lyapunov exponents from atime series. *Phys D*, 285-317.