Neural Networks for Chaotic Time Series Forecasting

Vladimir Golovko[1], Yury Savitsky[2]
[1] Institute of Engineering Cybernetics, Surganova str. 6, Minsk, Republic of Belarus
gva@brpi.unibel.by
[2] Brest State Technical University, Moscowskaja str. 267, Brest, Republic of Belarus
sjv@brpi.unibel.by

Abstract: *This paper examines neural network in order to predict behavior of chaotic systems. The prediction is performed both on the level of emergent structures and on the level of individual data points. The network is tested using the Henon and Lorenz chaotic time series. The results of experiments and future directions are discussed.*

Keywords: - Chaotic Time Series, Multilayer Neural Network, Horizon of Prediction

1. INTRODUCTION

Problems concerning forecasting have existed for a very long time. People often have tried to predict the future. Traditional approach of time series analysis is based on linear mathematics. However, linearity is not too good tool for investigation of chaotic processes. In last years neural networks have been used to predict chaotic time series.

Chaotic behavior is characterized by highly sensitive to initial conditions and observed for many systems (stock market, EEG patterns of brainwave activity, central nervous system, etc.). The key problem of chaotic time series is unpredictable on the long term, because error at the first step prediction is increased exponentially at time. That's why the improvement of prediction accuracy is of great importance. It permits also to understand the behavior observed nonlinear system and to perform state space reconstruction, taking into account numerical data from complex system. It is based on an embedding theorem [1], which guarantees that a full knowledge of the behavior a system is contained in time series of any a one measurement. As a result the full multivariate phase space can be constructed from the single time series.

To apply the embedding theorem it is necessary to define a suitable embedding dimension and time delay. There exist a lot of methods for estimating the optimal time delay τ (autocorrelation function, mutual information, etc) and embedding dimension m such as the false nearest neighbors, fractal dimension, principal component analysis and so on [2, 3]. The estimation such parameters provide a maximum predictability of chaotic time series and can be used for choosing of optimal window size (number of input units) in forecasting neural network. The paper is organized as follows. Section 2 describes the applying of neural networks for chaotic data prediction and state space reconstruction. In section 3 is presented the computing of upper prediction limit. The section 4 describes the approach in order to increase the predicting horizon and section 5 gives summary.

2. PREDICTION AND STATE SPACE RECONSTRUCTION

The goal of time series prediction can be stated as follows: for a given sequence *x(1), x(2)...,x(l)* it is necessary to find continuation *x(l+1), x(l+2)...* The nonlinear predictive model can be presented, as

$$x(t) = F(x(t-1), x(t-2),...,x(t-k)),$$

where $t = k+1, N$, F – nonlinear function, provided by ANN nonlinear units and k is size of the sliding window, which is equal to number of time series elements simultaneously submitted to inputs of a neural network. We will apply Multilayer Perceptron (MLP)for time series prediction. Such a network is capable of approximating any function. Another important feature of MLP is the ability to generalization. Therefore MLP is powerful tool for design of predicting systems.

For achievement of maximum predictability it is necessary to define embedding parameters. Let's examine applying feed-forward neural networks for chaotic time series forecasting and state space reconstruction. As the chaotic systems, which we want to model are the Lorenz and Henon attractors. The Lorenz attractor is defined by the three-coupled differential equations:

$$\frac{dx}{dt} = G(y-x), \frac{dy}{dt} = -xz + rx - y, \frac{dz}{dt} = xy - bz. \quad (1)$$

This system is chaotic for the parameter values *G=10, r=28* and *b=8/3*.

We solved (1) using a 4-th order Runge-Kutta approach with a time step 0,01. Fig. 1 and Fig. 2 show Lorenz time series (x-axis) and 3-dimensional attractor respectively. Using the mutual information we can define that *?=0,16*. Analogously applying the method of false nearest neighbors we can get embedding dimension *m=5*. From this follows, that window size $k \geq m - 1 = 4$. The Henon attractor is described by the following equations:

$$\begin{cases} x_{n+1} = 1 - \alpha x_n^2 + y_n \\ y_{n+1} = \beta x_n \end{cases},$$

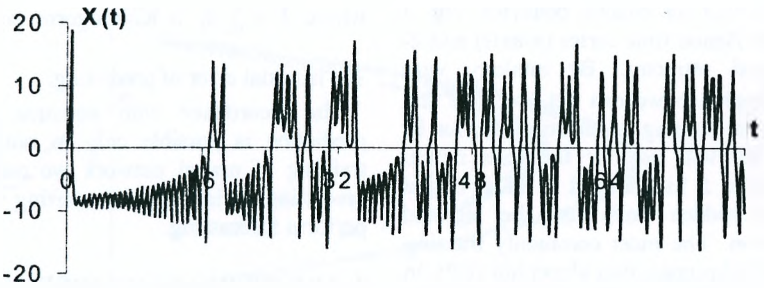

Figure 1. Original Lorenz time series (X-axis)

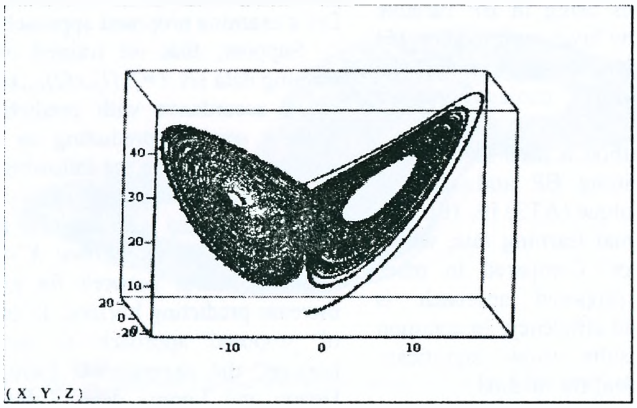

Figure 2. Original Lorenz attractor

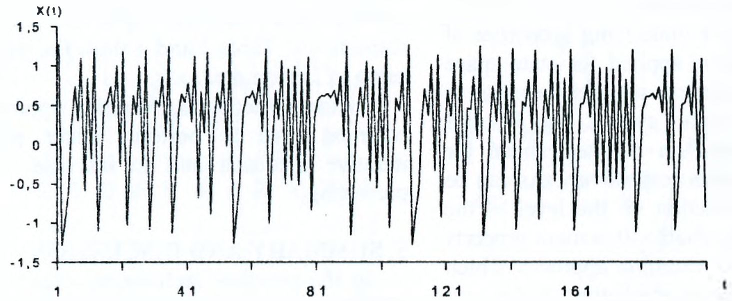

Figure 3. The Henon time series (first 200 elements)

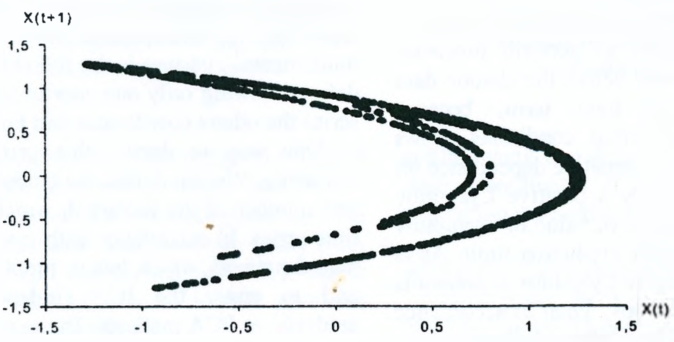

Figure 4. Original Henon attractor constructed for 1500 elements

where ?=1,4 and ?=0,3 for chaotic behavior. Fig. 3 and Fig. 4 show the Henon time series (x-axis) and 2-dimensional original attractor. By analogy with mentioned above approach we can get that $k \geq 2$ and ?=1.Let's examine forecasting of chaotic behavior by means of MLP. We will use for the tests neural network with 7 input, 5 hidden and 1 linear output units. Notice that hidden units use the sigmoid function of activation. The most commonly training method of MLP is backpropagation algorithm (BP). In spite of the face that BP is successfully used for different tasks, it has lacks such as slow convergence, non-stability of convergence and local minimum problems. Many efforts have been made to develop the efficient training methods using in BP variable training step size [4], layer by layer optimization [5] and using for training the Newton method [6], the Levenberg-Marguardt method [7], conjugal-gradient technique [8].

In this work a simple method is used for efficient training of MLP by combining BP and adaptive training step calculation technique (ATS) [9, 10]. The ATS is used to find an optimal learning rate, which minimizes the training error. Compared to other training algorithms, the proposed approach is characterized by simplicity and efficiency. In common case the experimental results show significant improvement over the traditional BP method.

Based on the iterative approach we predicted the Henon data and Lorenz data for 1500 step ahead. The predicted Lorenz and the Henon attractors are shown in Fig. 5 and Fig. 6. As can be seen the neural network has the ability to capture the underlying properties of chaotic behavior and can be applied for state space reconstruction. Thus the neural network permits to predict the behavior of complex system. Fig. 7 and 8 show the prediction results on 30 step ahead for Henon and Lorenz time series respectively. As can be seen from figures the prediction on the level of the individual data points is unreliable. It is main property of chaotic system. We will examine approach which permits to increase horizon of prediction in the next sections.

3. HORIZON OF PREDICTION FOR CHAOTIC TIME SERIES

Horizon of prediction is characterized a range of time on which it is possible to perform precision forecasting. As it is mentioned before the chaotic data are unpredictable on the long term, because measurement error in the initial conditions grows exponentially in time. Such a sensitive dependence on initial conditions is defined by a positive Lyapunov exponent. That's why the positive value of Lyapunov exponent determines the upper prediction limit. As is well known, the sum of positive Lyapunov exponent is equal to the Kolmogorov entropy. Then in accordance with chaos theory the horizon of prediction can be represented as follows:

$$T \approx \frac{1}{K} \cdot \ln\left(\frac{1}{d_0}\right) \qquad (2)$$

where $K = \sum_i \lambda_i$ is Kolmogorov entropy and $\lambda_i > 0$, d_0 is initial error of predicting.

In accordance with equation (2) the accurate prediction is possible only in range T. Thus, after training of neural network we can find horizon of predicting for initial point, starting with which one we perform forecasting.

4. THE INCREASE OF PREDICTING HORIZON

As it has been mentioned above, the horizon of prediction for chaotic behavior is limited in accordance with equation (2). One way to increase of predicting horizon is to retraining neural network. Let's examine proposed approach more detailed.

Suppose, that we trained neural network using training data set $X=\{x(1),x(2),..,x(N)\}$.

In accordance with predicting horizon we can perform accurate predicting on T point ahead. As a result we can define the following predicting points:

$x(N+1),x(N+2),\ldots,x(N+T)$.

The next step is to organize the new training data set, for instance, as follows: $X'=\{x(1),x(2),..,x(N+T)\}$. Training neural network for new data set we can increase predicting horizon. In order to test the ability of proposed approach to increase of predicting horizon, the experiments have been performed on Henon and Lorenz data. Table 1 and 2 show the comparative results of iterative and retraining approaches. The MSE1 and MSE2 are mean square error for the predicted points $x(N+1),x(N+2), x(N+3), x(N+4)$ and $x(N+5), x(N+6), x(N+7), x(N+8)$ respectively. Table 3 and 4 show the similar results by using of Lorenz data.

As can be seen the retraining approach permits in common case to perform better prediction than iterative approach and to increase the horizon of predicting.

5. SUMMARY AND DISCUSSION

In the previous sections we examined the simple predicting approach, when for given time-series up to time N it is necessary to find the continuation of time series. The more complex problem is described as follows: given a d-dimensional chaotic system, which is defined by the d differential equations. However we have only one-dimensional observed data. Then the fundamental question is the following: Is it possible to define knowing only one coordinate of observed time series the others coordinates and how to do it?

One way to decide this problem may be the following. We can define the embedding dimension m and number of the factors d, which influence on this time series. In accordance with it we can construct the neural network which has m input and d output units and to apply the ICA (independent component analysis) or PCA methods. But experiments on Lorenz data show, that these methods are not suitable for such a task. For instance, we try to mix the coordinates of the Lorenz data and after this to get the original sources. As a result traditional ICA is not able to do it.

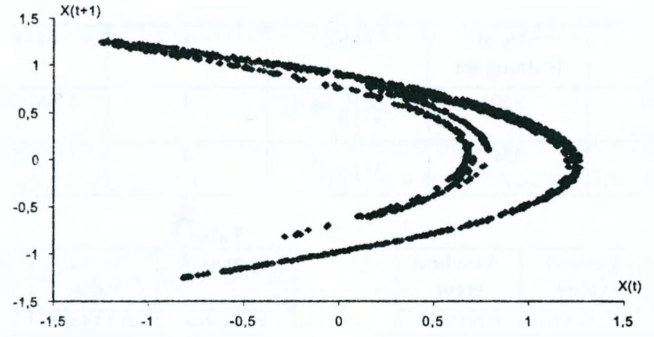

Figure 5. Predicted Henon attractor constructed for 1500 predicting iterations

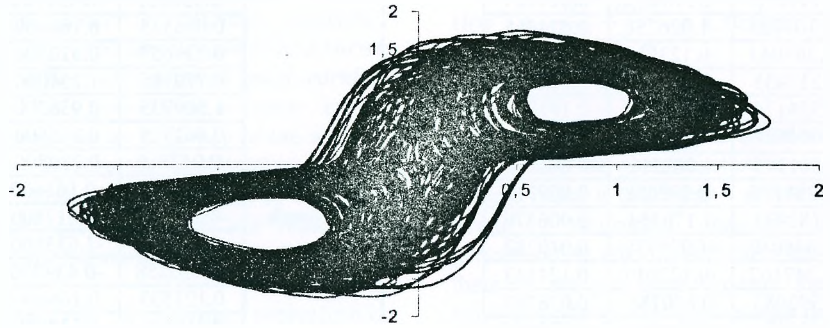

Figure 6. Predicted Lorenz attractor constructed for 1500 predicting iterations

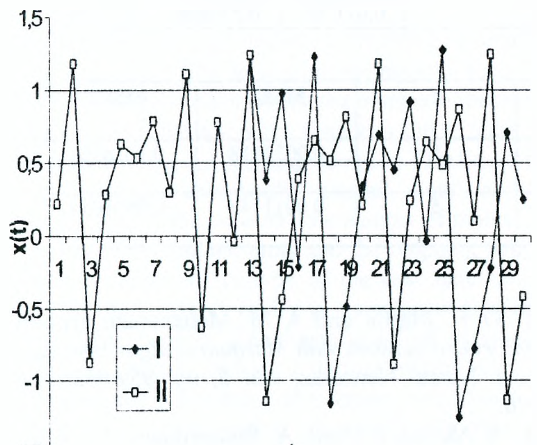

Figure 7. Henon process: prediction results for 30 predicting iterations (using retraining approach) I – prediction, II – original sequence

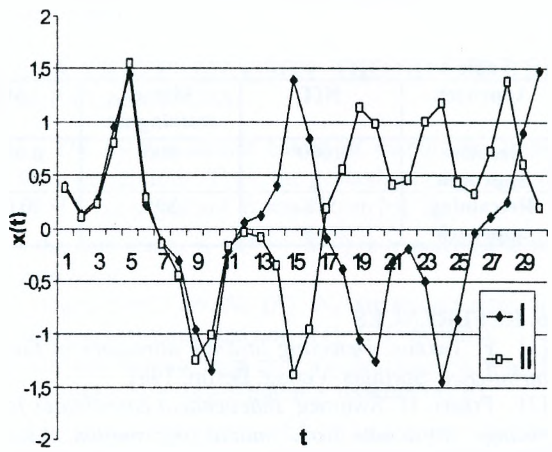

Figure 8. Lorenz process: prediction results for 30 predicting iterations (using retraining approach) I – prediction, II – original sequence

As a rule the ICA method gives the good results if we use non-gaussian (deterministic) data and PCA – on contrary. The chaotic data are more gaussian with comparison with other data. That's why by using PCA we got better results in comparison with ICA. But it should be noted, that PCA also doesn't give the suitable decision of given problem.

That's why the next task is to develop a powerful tool for separation of chaotic time series and for getting using one-dimensional chaotic time series the others coordinates. It can permit to analyze the past and define the hidden factors.

ACKNOWLEDGMENTS

This paper is supported by INTAS program "INTAS OPEN 97-0606" and by Belarus and Russian Foundations.

Table 1

Approach	NIT	Size of training set	MSE	T	MSE1	MSE2
Iterative approach	308	950	$3 \cdot 10^{-4}$	4	0.00022275	0.031198
Retraining approach	276	954	$3 \cdot 10^{-4}$	4	0.00003325	0.00804275

Table 2

Approach	Actual value	Desired value	Absolute error
Iterative approach	0.365621	0.363170	0.002451
	0.992627	1.002511	0.009884
	-0.274204	-0.298088	0.023884
	1.191078	1.176354	0.014724
	-1.101723	-1.026758	0.074965
	-0.363043	-0.123019	0.240024
	0.512435	0.670785	0.158350
	0.524174	0.333160	0.191014
Retraining approach	0.364677	0.363170	0.001507
	1.001295	1.002511	0.001216
	-0.288775	-0.298088	0.009313
	1.182933	1.176354	0.006579
	-1.046040	-1.026758	0.019282
	-0.247162	-0.123019	0.124143
	0.592083	0.670785	0.078702
	0.434126	0.333160	0.100966

Table 3

Approach	Actual value	Desired value	Absolute error
Iterative approach	-0.155480	-0.163600	0.008120
	-0.556713	-0.617800	0.061087
	-1.573766	-1.633100	0.059334
	-0.536221	-0.439700	0.096521
	0.085535	0.186400	0.100865
	0.237657	0.520500	0.282843
	0.719185	1.254000	0.534815
	1.509935	0.938200	0.571735
	0.461715	0.245600	0.216115
	-0.042810	0.230800	0.273610
Retraining approach	-0.169124	-0.163600	0.005524
	-0.613167	-0.617800	0.004633
	-1.598149	-1.633100	0.034951
	-0.430258	-0.439700	0.009442
	0.121533	0.186400	0.064867
	0.317614	0.520500	0.202886
	0.940051	1.254000	0.313949
	1.336355	0.938200	0.398155
	0.301510	0.245600	0.055910
	0.011798	0.230800	0.219002

Table 4

Approach	NIT	Size of training set	MSE	T	MSE1	MSE2
Iterative approach	1000	800	0.001357	5	0.0053618	0.1628954
Retraining approach	578	805	0.0014	5	0.0011142	0.0698684

6. REFERENCES

[1] F. Takens. *Detecting Strange attractors in fluid turbulence*. Springer-Verlag, Berlin, 1981.

[2] Fraser, H. Swinneg. *Independent coordinates for strange attractors from mutual information*. Phys. Rev. A 33, 1134 (1986).

[3] M. Kennel, R. Brown, H. Abarbanel. *Determining embedding dimension for phase-space reconstruction using a geometrical construction*. Phys. Rev. A 45, 3403 (1992).

[4] X.-H. Yu and G.-A. Chen, *Efficient backpropagation learning using optimal learning rate and momentum*. Neural Networks, vol. 10, no. 3, pp. 517-527, 1997.

[5] G.-J. Wang and C.-C. Chen. *A fast multilayer neural-network training algorithm based on the layer-by-layer optimizing procedures*. IEEE Trans. Neural Networks, vol. 7, pp. 768-775, May 1996.

[6] R. Battiti. *'First- and second-order methods for learning: Between steepest descent and Newton methods*. Neural Comput., vol. 4, pp. 141-166, 1992.

[7] Ò. Í . Martin and Å M. Mohammad. *Training feedforward network with Marquardt algorithm*. IEEE Trans. Neural Networks, vol. 5, pp. 959-963, Nov. 1996.

[8] S. Ahmed, J. Cross, A. Bouzerdoum. *A new self-adaptive backpropagation training method*. Proc. of Int. Joint Conf. on Neural Networks IJCNN'2000, Como, Italy. – 2000.

[9] V. Golovko Yury Savitsky, Theodore Laopoulos, Anatoly Sachenko, Lucio Grandinetti. *Technique of learning rate estimation for efficient training of MLP*. Proc. of Int. Joint Conf. on Neural Networks IJCNN'2000, Como, Italy. – 2000. – p. 323-329.

[10] Vladimir Golovko. *Neural Networks: training, organization and application*. Moscow, Radiotechnica 2001, – 350p.