Vol 17 No 2, February 2008
1674-1056/2008/17(02)/0536-07

Chinese Physics B

© 2008 Chin. Phys. Soc.
and IOP Publishing Ltd

Multi-step-prediction of chaotic time series based on co-evolutionary recurrent neural network*

Ma Qian-Li(马千里)[a)†], Zheng Qi-Lun(郑启伦)[a)], Peng Hong(彭 宏)[a)],
Zhong Tan-Wei(钟谭卫)[b)], and Qin Jiang-Wei(覃姜维)[a)]

[a)] *College of Computer Science and Engineering, South China University of Technology, Guangzhou* 510640, *China*

[b)] *College of Science, South China Agriculture University, Guangzhou* 510640, *China*

(Received 4 June 2007; revised manuscript received 7 July 2007)

This paper proposes a co-evolutionary recurrent neural network (CERNN) for the multi-step-prediction of chaotic time series, it estimates the proper parameters of phase space reconstruction and optimizes the structure of recurrent neural networks by co-evolutionary strategy. The searching space was separated into two subspaces and the individuals are trained in a parallel computational procedure. It can dynamically combine the embedding method with the capability of recurrent neural network to incorporate past experience due to internal recurrence. The effectiveness of CERNN is evaluated by using three benchmark chaotic time series data sets: the Lorenz series, Mackey–Glass series and real-world sun spot series. The simulation results show that CERNN improves the performances of multi-step-prediction of chaotic time series.

Keywords: chaotic time series, multi-step-prediction, co-evolutionary strategy, recurrent neural networks

PACC: 0545

1. Introduction

In recent years, the prediction of chaotic time series has become an important research domain of chaotic signal processing. Many prediction methods are based on the Takens embedding theorem.[1] Neural network is one of the methods, which has demonstrated great potential for time series prediction where system is nonlinear. With the ability to approximate any nonlinear function universally, neural networks are valuable tools for chaotic time series modelling. Multi-layer perception,[2] radial basis function networks,[3,4] fuzzy neural networks,[5,6] support vector machine (SVM)[7−10] and other modified networks[11,12] were used frequently for nonlinear time series prediction. But traditional neural networks for multi-step-prediction of chaotic time series has some inherent limitations that static networks cannot find an appropriate finite nonlinear model for every dynamical system when increasing the prediction steps.

For the capability of internal recurrence, recurrent neural network (RNN) is an interesting topic in the neural network techniques. RNN can project the dynamic properties of the system automatically, so they are computationally more powerful than feedforward networks, and valuable approximation results were obtained for multi-step-prediction of chaotic time series.[13−15]

Despite the efficiency of RNN and phase space reconstruction method in prediction of chaotic time series, two main problems must be addressed when multi-step-prediction is concerned. First, the design of RNNs is difficult. The performance of neural networks highly depends on the architecture of the networks and their parameters especially when the prediction steps were increased. Therefore, determining the architecture of a network (especially input and hidden neuron size, structure) greatly affects the performance criteria, i.e. learning speed, accuracy of learning, noise resistance and generalization ability. Second, one of the issues in the time-delay embedding is to estimate the dimension m and time delay τ. Method of determining embedding dimension was proposed in Ref.[16], but this kind of methods were not only complex to compute these two parameters of phase space reconstruction but also would obtain improper results, because actual chaotic time series can not meet the condition

*Project supported by the State Key Program of National Natural Science of China (Grant No 30230350) and the Natural Science Foundation of Guangdong Province, China (Grant No 07006474).

†E-mail: MQL206@tom.com

of Takens' assumption that there are no noises in data and the length of observed sequence is infinite.

In this paper, a co-evolutionary recurrent neural network (CERNN) is proposed for the multi-step-prediction of chaotic time series. CERNN is a combination of estimating the proper parameters (m, τ) of phase space reconstruction and optimizing the architecture of RNN by co-evolutionary strategy. Our idea is based on RNNs ability to understand nonlinearities presented in the chaotic time series and on co-evolution algorithm to search a solution in the massively landscape of the different RNNs space. Of course this procedure does not guarantee to find the optimal configuration, but it allows attaining a configuration close to optimal performance. Simulation results for multi-step-prediction of chaotic time series problems show the effectiveness of the proposed method. The results are compared with other techniques used in literature.

The rest of the paper is organized as follows: Section 2 shows our method, and describes the design of the CERNN. In Section 3, the predicted results are presented and compared with other methods. Finally, our conclusions are summarized in the last section.

2. CERNN

2.1. Phase space reconstruction and time delay embedding

Takens' theorem provides the theoretical justification for reconstructing phase space by using time-delay embedding. Suppose that a time series $\{x(t), t = 1, 2, \ldots, n\}$ is a measurement on a chaotic dynamical system in the state space. The vectors (data points) X_t $t = 1, 2, \ldots, N$ in the embedded phase space R^m is represented by

$$W = \begin{bmatrix} X_{1+(m-1)\tau} \\ X_{2+(m-1)\tau} \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_{1+(m-1)\tau} & \cdots & x_{1+\tau} & x_1 \\ x_{2+(m-1)\tau} & \cdots & x_{2+\tau} & x_2 \\ \vdots & \ddots & & \\ x_N & \cdots & x_{N-(m-2)\tau} & x_{N-(m-1)\tau} \end{bmatrix}_{(N-(m-1)\tau)\times 1} \tag{1}$$

where τ is the time delay and m is the embedding dimension. Since the dynamics of the reconstructed phase space contains the same topological information as the original phase space, the characterization and prediction based on the reconstructed phase spaces is as valid as those from the original phase space. Marking a dynamical system as F: $R^m \to R^m$, the mathematical model can be described as: $X_{t+k} = F(X_t)$ $t = 1, 2, \ldots, N$, where k means the step of prediction. Then we can get a function f: $R^m \to R$, that is

$$x_{t+k} = f(x_{t-(m-1)\tau}, x_{t-(m-2)\tau}, \ldots, x_t). \tag{2}$$

Given this fact, the estimation of the embedding dimension m and time delay τ greatly affects the prediction accuracy. A sufficient condition is that m is greater than twice the dimension of the original state space. Furthermore, it is essential to select appropriate time delay τ in the reconstruction phase space strategy. The results obtained by earlier methods, such as false nearest neighbour and mutual information, estimate m and τ respectively which, show that prediction accuracy was not high when the original time series contains both noise and chaotic signals and prediction steps are increased. So in Section 2.3 we will use co-evolutionary algorithm to estimate the parameters of reconstruction phase space and recurrent networks.

2.2. Network representation and genetic operators

In this paper, we use a typical recurrent neural network, known as the Elman network. The Elman network commonly is a two-layer network with feedback from the first layer output to the first layer input. It has tansig neurons in its hidden (recurrent) layer, and purelin neurons in its output layer. The Elman network differs from conventional two-layer networks in that it has an additional input layer, called the context units, receiving as feedback a copy of the ac-

tivations from the hidden layer at the previous time step.

There are two kinds of neural network representations: low-level representation and high-level representation. A low-level representation specifies exactly every network parameter. This causes a large search space, so that the number of iterations increases dramatically. In contrast, a high-level representation is an abstract description of the networks where input/hidden neurons number and parameters to determine the distribution of units characterize the network structure.[17]

In contrast to early works, our representation considers both high-level representation of the Elman network and character of chaotic time series data. The representation is shown in Fig.1 by a dynamic data-structure. We separate it into two subspaces: network parameters space and phase space reconstruction parameters space. The former space specifies number of input neurons m and hidden neurons n and the latter specifics embedding dimension m and time delay τ. Here, m represents not only the number of input dimension but also the embedding dimension.

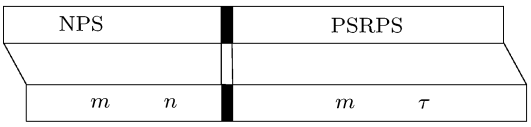


Fig.1. CERNN representation.

In order to prevent incorrect networks from participating in the evolution cycle and save useless training, we set some restrictive condition for the range of m, n and τ. Due to the recursion mechanism, the prediction value x_{t+k}(k is the prediction step) depends on previous values: $x_{t-1}, x_{t-2}, \ldots, x_1$, but the dependence is potentially much weaker compared to x_t especially when $k > 1$. If t is more than 5, the values of x_t ($t \geq 5$) have little influence to the prediction value x_{t+k} ($k > 1$). Therefore, m and τ are all in the range [1, 5]. We set hidden neurons 1 to 4 times as many as the inputs to extend search space. That is to say, the range of n is [1, 20]. For above restrictions, the population size and the number of iterations can be reduced to a number that makes the problem tractable. This is also a kind of pre-structuring which gives some evidence that first-generation networks can solve the given problem and genetic search can focus on finding improved networks. With the representation, estimation of phase space reconstruction parameters can solely be done by the co-evolutionary process.

Three operators are used: mutation, crossover, selection. These operators should maintain genetic diversity and reproduction should inherit features. The mutation operator is applied to RNNs which are chosen with the probability given by the mutation-rate ($p_m = 0.1$). Crossover-rate is 0.3 ($p_c = 0.3$).

2.3. Co-evolving problem process

We use a two level co-evolutionary architecture, the lower level has a population of phase space reconstruction parameters (m, τ) and the higher level has a population of partial trained networks. Evolution at lower level searches for proper parameters of our model and at higher level searches for models with higher prediction accuracy. After training, a set of data for validation are used. Within this co-evolutionary process, both the structure and parametrization of the RNNs as well as the parameters of phase space reconstruction can be evolved. In each generation both populations go through the stages illustrated in Fig.2.

First, in order to establish a starting point for optimum seeking, the population must be initialized. A natural way to initialize the population is to seed it with random values within the given range (m in [1, 5], τ in [1, 5], n in [1×m, 4×m]). Second, using mutation, crossover, we get a group of new RNNs. They are trained by using Levenberg–Marquardt algorithm. Finally, the actual fitness values are calculated for each network according to its performance on the validation data. The CERNN directly provides the error of a network as a measure of fitness. In each generation both populations go through the stages illustrated in Fig.2. Each individual in validation stage is trained for 200 epochs, and then it will be tested on validation data. So the individuals in validation stage undergo partial training. When the elite individuals are selected, they will join in the next generation. The process will not stop until the maximum generation is met or the fitness is acceptable. When the final network is selected, it will be fully trained for 4000 epochs by using Levenberg–Marquardt algorithm. Fitness of RNN R_i is evaluated as

$$\text{FITNESS}(R_i) = 1/\text{mse} \tag{3}$$

where mse is the mean squared error on validation data. It will be described in Eq.(4).

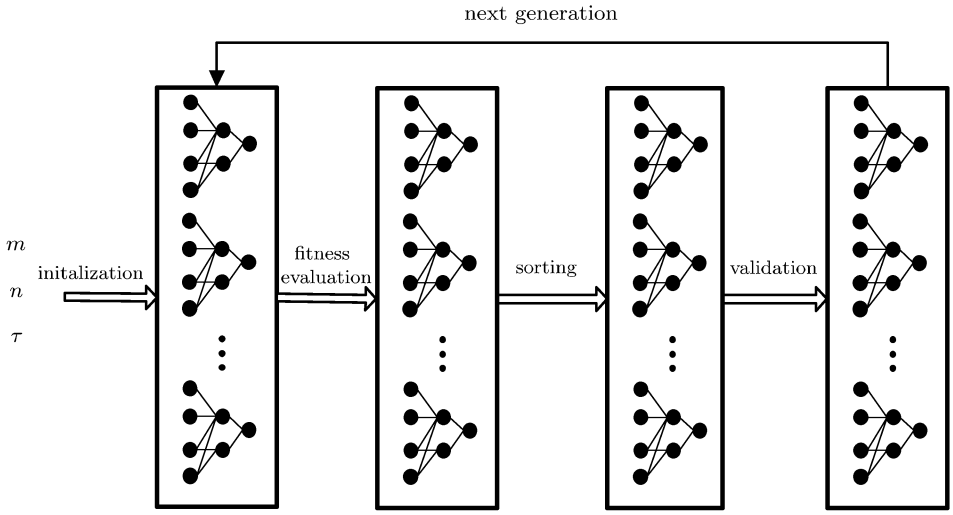


Fig.2. Co-evolutionary stages.

3. Experimental verification

Our experimental verification is conducted by three benchmark chaotic time series: Lorenz, Mackey–Glass, and Sunspots. Each example uses the corresponding 2000 values after omitting initial transient points. The data are divided into three parts: the first 1000 values are the training set,the next 500 values are used to validate the model identified, the remaining 500 values are test set. Four error criterions are used as performance measure: Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Relative Mean Squared Error (RE for short in order to differentiate it with RMSE) and absolute error $e(t)$. They are the main evaluation yardstick almost in all of the experiments. They are showed by the following equations respectively:

$$\mathrm{MSE} = \frac{\sum_{t=1}^{S} [\hat{y}(t) - y(t)]^2}{S}, \tag{4}$$

$$\mathrm{RE} = \frac{\sum_{t=1}^{S} [\hat{y}(t) - y(t)]^2}{\sum_{t=1}^{S} y^2(t)}, \tag{5}$$

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}, \tag{6}$$

$$e(t) = \hat{y}(t) - y(t). \tag{7}$$

where $y(t)$, $\hat{y}(t)$ and S are observed value, prediction value and size of observed data set respectively. In order to reduce the effects of the initial values of free parameters on the final results, for each of the case, 10 runs were performed with randomly set initial parameters. We will compare the prediction performances of CERNN with the other methods existing in the bibliography.

The parameters used for our CERNN model are listed in Table 1.

Table 1. The parameters of CERNN.

Population size	20
p_m	0.1
p_c	0.3
Generations	10
m	[1,5]
τ	[1,5]
n	$[1 \times m, 4 \times m]$
Maximum epochs to train	4000

Example 1 Lorenz chaotic system is described by the following equations:[18]

$$\begin{aligned} \dot{x} &= \sigma(y - x), \\ \dot{y} &= -xz + rx - y, \\ \dot{z} &= xy - bz, \end{aligned} \tag{8}$$

$$\sigma = 10, r = 28, b = 8/3.$$

The 2000 values are normalized in the range [0, 1].

Table 2 lists the parameters and MSE after validation stage for partial training (step=6). It shows the elite individual in each generation and their prediction errors (MSE) on validation data. We can find individuals evolving to similar elite results. Although

each elite individual has different embedding dimension m, time delay τ is equal to 1. It implicates that the time delay should not be larger than 1, otherwise all data points would be almost independent.

Table 2. The elites of each generation and their prediction errors on validation data (step=6).

generation	m	τ	n	MSE
1	3	1	10	2.12×10^{-6}
2	5	1	6	2.76×10^{-6}
3	4	1	8	1.59×10^{-6}
4	4	1	9	1.42×10^{-6}
5	3	1	12	2.83×10^{-6}
6	5	1	12	9.22×10^{-7}
7	5	1	14	3.17×10^{-7}
8	5	1	13	4.15×10^{-7}
9	5	1	14	4.06×10^{-7}
10	5	1	13	3.66×10^{-7}

Figures 3–6 illustrate the prediction results of Lorenz series by the CERNN with step=6, 8 and their corresponding absolute errors $e(t)$ respectively. The absolute errors are in the range $[6\times10^{-4}, -6\times10^{-4}]$ for 6-step prediction and $[1\times10^{-3}, -1\times10^{-3}]$ for 8-step prediction.

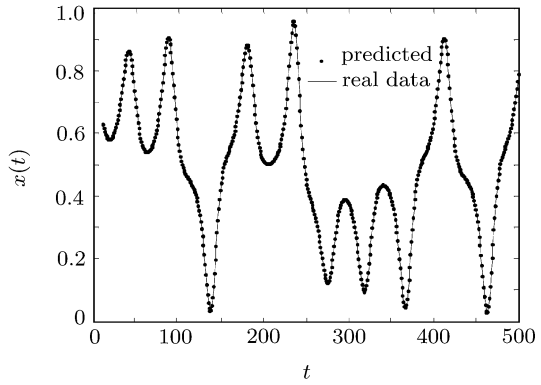


Fig.3. Prediction results of the Lorenz (step=6).

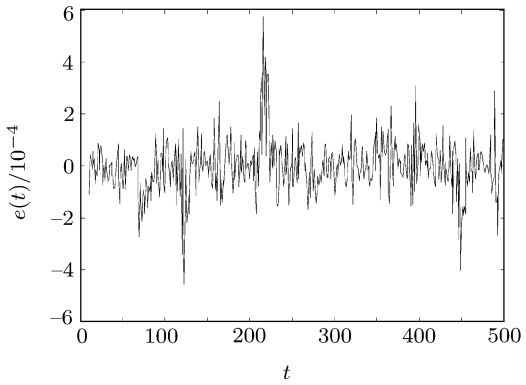


Fig.4. Prediction errors of the Lorenz (step=6).

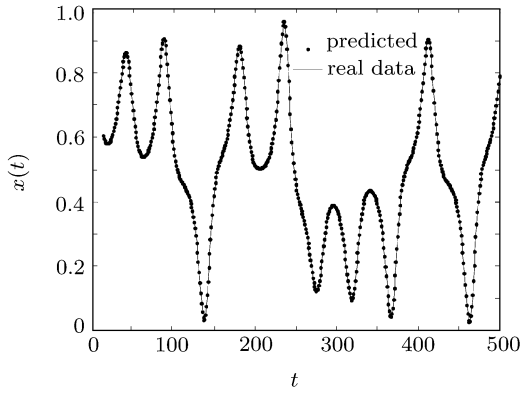


Fig.5. Prediction results of the Lorenz (step=8).

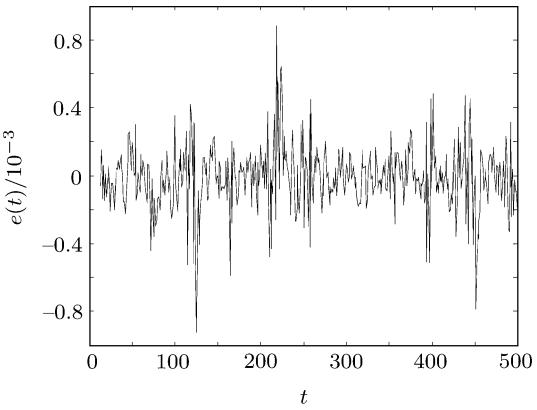


Fig.6. Prediction errors of the Lorenz (step=8).

In order to show the effectiveness of CERNN, we compare the 6 steps prediction performance with local discrete cosine transformation (DCT) domain Volterra prediction method.[19] The MSE and RE of CERNN are all better than local DCT domain Volterra. The prediction errors are improved by two orders of magnitude.

Table 3. Performance of different methods for Lorenz prediction (step=6).

Methods	MSE	RE
Local DCT Domain Volterra	4.1642×10^{-6}	1.3565×10^{-5}
CERNN	1.567×10^{-8}	5.3541×10^{-8}

Example 2 Mackey–Glass system is described by the following equation:[20]

$$\frac{\mathrm{d}x}{\mathrm{d}t} = a\frac{x(t-\tau)}{1+x^c(t-\tau)} - bx(t), \tag{9}$$

$$\tau = 17, a = 0.2, b = 0.1, c = 10.$$

Mackey–Glass was used frequently as a benchmark data set. We use the same initial conditions

with LS-SVM.[7] Table 4 lists RMSE of LS-SVM and CERNN. It can be seen that the prediction based on CERNN is more efficient.

Table 4. RMSE of different methods for Mackey–Glass prediction.

Prediction step	LS-SVM	CERNN
4	0.0113	0.00039
8	0.0245	0.00208
12	0.0496	0.00379

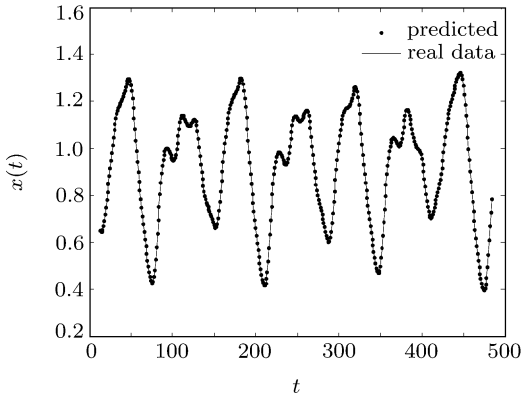


Fig.7. Prediction results of the Mackey–Glass (step=4).

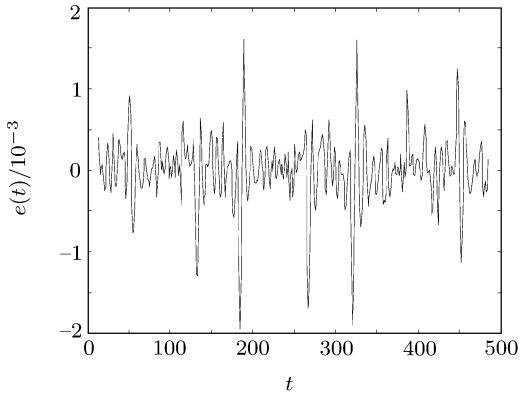


Fig.8. Prediction errors of the Mackey–Glass (step=4).

Moreover, the RMSE on 4 steps prediction of CERNN is better than that in Ref.[21], which is 0.0111. Figures 7–8 show the 4-step prediction results and absolute errors $e(t)$ for Mackey–Glass.

Example 3 The monthly smoothed sunspot number time series has been downloaded from the SIDC (World Data Centre for the Sunspot Index). In order to ensure fair comparison between different modelling methods, we use the same data set with MSRN,[13] monthly sunspot data from January 1749 to March of 1977, and normalize them in the range [0, 1]. Table 5 lists the MSE of MSRN and CERNN with different prediction steps.

Table 5. MSE of different methods for monthly sun spots series prediction.

Prediction step	MSRN	CERNN
4	0.008376	0.000847
8	0.007813	0.002882
12	0.010801	0.005619
18	0.014645	0.008229

4. Conclusions

In this paper, a novel technique for the multi-step-prediction of chaotic time series has been proposed. Unlike those of the general recurrent networks, the CERNN is designed according to co-evolution of the reconstruction phase space and parameters of recurrent network structure. Through the four co-evolutionary stages: initialization, partial training, sorting and validation, CERNN is more efficient to model or predict chaotic time series. Experiments on deterministic chaotic processes were carried to confirm and illustrate the performance of CERNN. The simulation results and prediction errors between the origin and generated time series show that the proposed method can capture the dynamics of the chaotic time series effectively.

References

[1] Takens F 1981 *Dynam. Syst. Turbulence* **898** 366

[2] Zhang J S and Xiao X C 2000 *Chin. Phys.* **9** 408

[3] Li J and Liu J H 2005 *Acta Phys. Sin.* **54** 4569 (in Chinese)

[4] Zhang J F and Hu S S 2007 *Acta Phys. Sin.* **56** 713 (in Chinese)

[5] Tan W, Wang Y N, Zhou S W and Liu Z R 2003 *Acta Phys. Sin.* **52** 795 (in Chinese)

[6] Hu Y X and Gao J F 2005 *Acta Phys. Sin.* **54** 5034 (in Chinese)

[7] Ye M Y and Wang X D 2004 *Chin. Phys.* **13** 454

[8] Cui W Z, Zhu C C, Bao W X and Liu J H 2005 *Chin. Phys.* **14** 922

[9] Yu Z H and Cai Y L 2006 *Acta Phys. Sin.* **55** 1659 (in Chinese)

[10] Zhang J S, Dang J L and Li H C 2007 *Acta Phys. Sin.* **56** 67 (in Chinese)

[11] Sun J C, Zhou Y T and Luo J G 2006 *Chin. Phys.* **15** 1208

[12] Ding G and Zhong S S 2007 *Acta Phys. Sin.* **56** 1224 (in Chinese)

[13] Inés M G and Pedro I 2001 *Neural Process. Lett.* **13** 115

[14] Han M, Xi J H, Xu S G and Yin F L 2004 *IEEE Trans. Signal Processing* **52** 3409

[15] Cheng H B, Tan P N, Gao J and Scripps J 2006 *Adv. Knowl. Disc. Data Mining* **3918** 765

[16] Meng Q F, Peng Y H and Xue P J 2007 *Chin. Phys.* **16** 1252

[17] Martin M 1995 *Evol. Comput.* **2** 584

[18] Lorenz E N and Atmos J 1963 *Science* **20** 130

[19] Zhang J S, Li H C and Xiao X C 2005 *Chin. Phys.* **14** 49

[20] Mackey M C and Glass L 1977 *Science* **197** 287

[21] Liu H, Liu D and Deng L F 2006 *Chin. Phys.* **15** 1196