TIME SERIES PREDICTION USING COMPUTATIONAL INTELLIGENCE

B. Samanta
biswanath.samanta@villanova.edu
Department of Mechanical Engineering
Villanova University, Villanova, PA 19085.

Abstract: In this paper, two CI techniques, namely, single multiplicative neuron (SMN) model and adaptive neuro-fuzzy inference system (ANFIS), have been proposed for time series prediction. A variation of particle swarm optimization (PSO) with co-operative sub-swarms, called COPSO, has been used for estimation of SMN model parameters leading to COPSO-SMN. The prediction effectiveness of COPSO-SMN and ANFIS has been illustrated using commonly used nonlinear, non-stationary and chaotic benchmark datasets of Mackey-Glass, Box-Jenkins and biomedical signals of electroencephalogram (EEG). The training and test performances of both hybrid CI techniques have been compared for these datasets.

Key words: Time series prediction; single multiplicative neuron model; computational intelligence; particle swarm optimization; nonlinear time series; biomedical signal analysis.

I. Introduction

Time series prediction involves predicting the system behavior in future based on information of the current and the past status of the system. Prediction of time series has widespread applications in the fields of science, engineering, medicine and econometrics, among others. Several methods have been used for prediction of real life complex, nonlinear time series commonly encountered in various such application domains [1-3]. In recent years, there is also a growing interest in incorporating bio-inspired computational algorithms, commonly termed as computational intelligence (CI), in discovering knowledge from data, both in education and research [4-9].

Among various CI techniques, artificial neural networks (ANNs) have been developed in form of parallel distributed network models based on biological learning process of the human brain. Among different types of ANNs, multi-layer perceptron (MLP) neural networks are quite popular [4]. Recently single multiplicative neuron (SMN) model has been proposed as an alternative to the general MLP type ANN. The SMN model derives its inspiration from the single neuron computation in neuroscience [10, 11]. The SMN model is much simpler in structure than the more conventional multi-layer ANN and can offer better performances, if properly trained [12, 13]. However, the

success of the SMN model depends on estimation of the model parameters in the training stage, similar to ANN.

Another CI technique, namely, particle swam optimization (PSO) was proposed by Kennedy and Eberhart [5] as a population based stochastic optimization technique inspired by the social behavior of bird flocking. PSO is a computationally simple algorithm based on group (swarm) behavior. The algorithm searches for an optimal value by sharing cognitive and social information among the individuals (particles). PSO has many advantages over evolutionary computation techniques like genetic algorithms in terms of simpler implementation, faster convergence rate and fewer parameters to adjust [6, 7]. The popularity of PSO is growing with applications in diverse fields of engineering, biomedical and social sciences, among others [7-9].

In the present work, the SMN model parameters have been estimated using PSO [14, 15]. A variation of PSO with co-operative sub-swarms, COPSO, has been used in this work [14]. The resulting combination is termed as COPSO-SMN.

Fuzzy logic (FL) has been used in many practical engineering situations because of its capability in dealing with imprecise and inexact information [16, 17]. The powerful aspect of fuzzy logic is that most of human reasoning and concept formation is translated into fuzzy rules. The combination of incomplete, imprecise information and the imprecise nature of the decision-making process make fuzzy logic very effective in modeling complex engineering, business, finance and management systems which are otherwise difficult to model. This approach incorporates imprecision and subjectivity in both model formulation and solution processes. The major issues involved in the application of FL or fuzzy inference system (FIS) are the selection of fuzzy membership functions (MFs), in terms of number and type, designing the rule base simulating the decision process as well as the scaling factors used in fuzzification and defuzzification stages. These parameters and the structures are, in general, decided based on multiple trials and expert knowledge. In adaptive neuro-fuzzy systems (ANFIS) proposed in [18], the advantages of FL and ANNs were combined for adjusting the MFs, the rule base and related parameters to fit the training dataset.

In this work, two CI techniques, COPSO-SMN and ANFIS, have been used for time series prediction. The prediction effectiveness of these techniques has been illustrated using commonly used nonlinear, non-stationary and chaotic benchmark datasets of Mackey-Glass, Box-Jenkins and biomedical signals of electroencephalogram (EEG) [19]. The training and test performances of both hybrid CI techniques have been compared for these datasets.

The rest of the paper is organized as follows. Section II briefly discusses the SMN model. In Section III, the basic PSO algorithm is presented. A brief discussion on ANFIS is presented in section IV. Section V presents the results and with conclusions in section VI.

II. Single Multiplicative Neuron (SMN) Model

Figure 1 shows the schematic of a general single multiplicative neuron (SMN) model with a learning algorithm for modeling a system with a single output y and the input vector **x**. The input vector $\mathbf{x}=\{x_i\}$with diagonal weight matrix $\mathbf{W}=[w_{ii}]$ and bias vector $\mathbf{b}=\{b_i\}$forms the intermediate

vector $\mathbf{p}=\{p_i\}$, i=1,n where n is the size of the input vector. The vector **p** goes through the multiplication node and gets transformed to y through the nonlinear function of logsig as follows:

$$\mathbf{p} = \mathbf{W}\mathbf{x} + \mathbf{b} \quad (1), \quad q = \prod_i^n p_i \quad (2), \quad y = \frac{1}{1+e^{-q}} \quad (3), \quad e = y_d - y \quad (4)$$

The aim of the SMN model is to minimize the error (e) between the target output y_d and the model output for the same input vector. The model parameters w_{ii} and b_i are adapted using the learning algorithm based on COPSO to minimize this error (e).

III. Particle Swarm Optimization (PSO)

A. Standard Particle Swarm Optimization (PSO)

In this section, a brief introduction to PSO algorithm is presented, for details text [8] can be referred to. Recent overviews of PSO and its variants are presented in [9] and [11]. For a problem with *n*-variables, each possible solution can be thought of as a *particle* with a position vector of dimension *n*. The population of *m* such individuals (particles) can be grouped as the *swarm*. Let x_{ij} and v_{ij} represent respectively the current position and the velocity of *i*th particle (*i=1,m*) in the *j*th direction (*j=1, n*). The fitness of a particle is assessed by calculating the value of the target or objective function for the current position of the particle. If the value of the objective function for the current position of the particle is better than its previous best value then the current position is designated as the new *best individual* (*personal*) location *pbest, p_{bij}*. The best current positions of all particles are compared with the historical best position of the whole swarm (*global* or *neighborhood*) *gbest, p_{bgj}*, in terms of the fitness function and the global best position is accordingly updated if any of the particle individual best (*pbest, p_{bij}*) is better than the previous global best (*gbest, p_{bgj}*). The current position and the velocity decide the trajectory of the particle. The velocity of the particle is influenced by three components, namely, inertial, cognitive and social. The inertial component controls the behavior of the particle in the current direction. The cognitive and the social components represent the particle's memory of its personal best position (*pbest*) and the global best position (*gbest*). The velocity and the position of the particle are updated for the next iteration step (*k+1*) from its values at current step *k* as follows:

$$v_{ij}(k+1) = v_{ij}(k) + c_1 U(0,1)(p_{bij}(k) - x_{ij}(k)) + c_2 U(0,1)(p_{bgj}(k) - x_{ij}(k)), \quad (5)$$

$$x_{ij}(k+1) = x_{ij}(k) + v_{ij}(k+1), \quad (6)$$

where r_1 and r_2 represent uniformly distributed random numbers in the range of (0,1). These random numbers present the stochastic nature of the search algorithm. The constants c_1 and c_2 define the magnitudes of the influences on the particle velocity in the direction of the individual and the global optima. N represents the maximum number of iterations (epoch). In many early applications, good results were obtained when the inertia term ω was decreased from 0.9 to 0.4

linearly enhancing the exploration at the beginning and exploitation towards the end of the solution process. In this work, c_1=2.0, c_2= 2.0 were used.

B. Co-operative Particle Swarm Optimization (COPSO)

In standard PSO, there is only one population (swarm). However, at times, especially for complex problems, it is advantageous to employ multiple co-operative swarms (sub-swarms). In this version, named as co-operative PSO (COPSO), multiple sub-swarms run in parallel to explore different segments of the search space and the particles exchange the *gbest* of all sub-swarms randomly in updating their velocity and position. The velocity updating equation (5) is rewritten as follows:

$$v_{ijl}(k+1) = v_{ijl}(k) + c_1 U(0,1)(p_{ijl}(k) - x_{ijl}(k)) + c_2 U(0,1)(p_{gj}(r) - x_{ijl}(k)) \quad (7)$$

where l=1,…,s, s being the number of sub-swarms and r is a random integer between 1 and s, representing the random index of the sub-swarm whose *gbest* is selected in the velocity update.

C. COPSO Based Learning of SMN Model Parameters

The aim of the present approach is to select the SMN model parameters (w_{ii} and b_i) such that an objective function representing the mean square error (MSE) is minimized.

$$J = \frac{1}{N}\sum_{o=1}^{N}(y_{do} - y_o)^2 \quad (8)$$

where o is the observation (sample) index and N represents the total number of samples. In the present work, COPSO was used to select the SMN model parameters from a user-given range [-15, 15] for each minimizing the objective function (7). A population size of 30 individuals split equally in three sub-swarms was used starting with randomly generated particle positions and velocities. The objective function (8) was used as the fitness function. The maximum generation of 1000 was used as the termination criterion.

IV. Adaptive Neuro-Fuzzy Inference System (ANFIS)

In this section, the main features of ANFIS are briefly discussed. Readers are referred to [18] for details. A typical ANFIS structure for a system consisting of m inputs $(x_1 \ldots x_m)$ each with n MFs, R rules and one output (y) is shown in Fig. 2. In the case of the time-series prediction, the output is $y=x_{t+r}$, i.e., the network is used to predict the series (y) r time steps ahead based on the current and the previous m values. For the present case of one step ahead prediction, r=1. The network consisting of five layers is used for training Sugeno-type fuzzy inference system (FIS) through learning and adaptation. Number of nodes (N) in layer 1 is the product of numbers of inputs (m) and MFs (n) for each input, i.e., $N=m\,n$. Number of nodes in layers 2-4 is equal to the number of rules (R) in the fuzzy rule base.

It requires a training dataset of desired input/output pair (x_1, $x_2 \ldots x_m$, y) depicting the target system to be modeled. ANFIS adaptively maps the inputs (x_1, $x_2 \ldots x_m$) to the outputs (y) through

MFs, the rule base and the related parameters emulating the given training data set. It starts with initial MFs, in terms of type and number, and the rule base that can be designed intuitively. ANFIS applies a hybrid learning method for updating the FIS parameters. It utilizes the gradient descent approach to fine-tune the premise parameters that define MFs and applies the least-squares method to identify the consequent parameters that define the coefficients of each output equation in the Sugeno-type fuzzy rule base. The training process continues till the desired number of training steps (epochs) or the desired root mean squared error (RMSE) between the desired and the generated output is achieved. In the present work, two MFs of generalized bell type were used for each input variable. In this work, the maximum epoch and the RMSE target were set at 100 and 10^{-4} respectively.

V. Results and Discussions

In this paper, the application of COPSO-SMN and ANFIS in time series prediction is illustrated using three datasets, namely, Mackey-Glass (MG) time series, Box-Jenkins gas furnace dataset and electroencephalogram (EEG) datasets [19]. The prediction performances of the CI algorithms are compared.

A. Mackey-Glass time series

The dataset of chaotic, non-convergent time-series was generated using Mackey-Glass equation (9) with initial condition of y(0)=1.2 and delay time τ=17.

$$\frac{dy(t)}{dt} = \frac{0.2y(t-\tau)}{1+y^{10}(t-\tau)} - 0.1y(t) \qquad (9)$$

The normalized response y(t) (within 1) of 950 data points after the initial transients was used to train and test both CI predictors. The aim was to predict y(k+1) from the values of previous time steps y(k), y(k-6), y(k-12) and y(k-18). The first 450 data points were used for training and next 500 points were used for testing the generalization capability of the predictors. Figure 3 shows the variations of the performance index (J) for three sub-swarms over the generations for COPSO-SMN. All sub-swarms converged to a very small value within 500 generations. First part of Table 1 shows the typical values of the SMN model weights and biases. Figures 4(a) and 4(b) show the predicted time series for training and test using SMN-COPSO. Figures 5(a) and 5(b) show the predicted time series tracking the target values quite closely in case of ANFIS. The prediction performance is represented in terms of normalized RMSE (NRMSE) which is the ratio of RMSE and the standard deviation of the target signal. The first part of Table 2 shows prediction NRMSE and the traing time for both CI techniques.. ANFIS performs better than SMN-COPSO both in terms of NRMSE and training time.

B. Box –Jenkins gas furnace data

The Box-Jenkins dataset represents the CO_2 concentration as output, y(t), in terms of input gas flow rate, u(t), from a combustion process of a methane-air mixture (BJ). From a total set of 296 data pairs, first 140 data points were used for training and next 140 were used for test. The aim is to predict y(k) in terms of y(k-1) and u(k-4). Table 2 presents the SMN parameters of the trained

model. Table 2 shows the training and test results. Again, ANFIS outperforms SMN-COPSO. Figures 6 and 7 show the predicted time series with SMN-COPSO and ANFIS respectively.

C. Electroencephalogram (EEG) dataset

EEG dataset [19] was used to illustrate the procedure. Both training and test datasets consist of 750 samples. The aim is to predict y(k) in terms of y(k-1), y(k-2), y(k-4) and y(k-8) using the CI predictors. Table 1 shows the SMN parameters and Table 2 show the prediction performance. The error levels are higher for both SMN and ANFIS compared to the first datasets. ANFIS performs better than SMN. Figures 8 and 9 show the predicted EEG signals for SMN and ANFIS.

VI. Conclusions

Results are presented for prediction of nonlinear, chaotic and non-stationary time series using two bio-inspired computational intelligence techniques. The single multiplicative neuron model parameters were estimated using a learning algorithm based on a cooperative particle swarm optimization PSO. Though both techniques show reasonably good results, ANFIS performs better than COPSO-SMN for all three datasets. The role of bio-inspired CI techniques in time series prediction is illustrated using three well known benchmark datasets.

References:

1. De Gooijer, J. G. and Hyndman, R. J. *25 years of time series forecasting. International Journal of Forecasting*, vol. 22, 2006, pp. 443-473.
2. Box, G.E.P., Jenkins, G.M., and Reinse, G.C. *Time Series Analysis: Forecasting and Control*, Prentice Hall, Englewood Cliffs, NJ, 1994.
3. Mackey, M. and Glass, L. *Oscillation and chaos in physiological control systems. Science*, vol. 197, 1997, pp. 287-289.
4. Haykin, S. *Neural Networks: A Comprehensive Foundation*, 2nd Edition, Prentice Hall, New Jersey, USA, 1999.
5. Kennedy, J. and Eberhart R. C. *Particle swarm optimization. Proc. IEEE Intl. Conf. on Neural Networks IV*, Piscataway, NJ: IEEE Service Center, 1995, 1942–1948.
6. Kennedy, J., Eberhart, R.C., and Shi, Y. *Swarm Intelligence*, Morgan Kaufmann Publishers, San Francisco, CA, 2001.
7. Poli, R., Kennedy, J., and Blackwell, T. *Particle swarm optimization an overview. Swarm Intelligence*, vol. 1, 2007, pp. 33-57.
8. Samanta, B. and Nataraj, C. *Use of particle swarm optimization for detection of machine condition. Engineering Applications of Artificial Intelligence*, vol.22, 2009, pp. 308-316.
9. Samanta, B. and Nataraj, C. *Prognostics of machine condition using soft computing. Robotics and Computer-Integrated Manufacturing*, vol. 24, 2008, pp. 816-823.
10. Koch, C. *Computation and single neuron. Nature*, vol. 385, 1997, pp. 207-210.
11. Koch, C. and Segev, I. *The role of single neurons in information processing. Nature Neuroscience supplement*, vol. 3, 2000, pp. 1171-1177.
12. Herz, A. V. M., Gollisch, T., Machens, C. K., and Jaeger, D. *Modeling single-neuron dynamics and computations: a balance of detail and abstraction. Science*, vol. 314, 2006, pp. 80-84.
13. Schmitt, M. *On the complexity of computing and learning with multiplicative neural networks. Neural Computation*. vol. 14, 2001, pp. 241-301.
14. Zhao, L. and Yang, Y. *PSO-based single multiplicative neuron model for time series prediction. Expert Systems with Applications*, vol. 36, 2009, pp. 2805-2812.

15. Yadav, R. N., Kalra, P.K., and John, J. *Time series prediction with single multiplicative neuron model. Applied Soft Computing*, vol. 7, 2007, pp. 1157-1163.
16. Zadeh, L.A. Fuzzy sets. *Information and Control*, vol. 8, 1965, pp. 338-353.
17. Yen, J. and Langari, R. *Fuzzy logic: intelligence, control and information*, Prentice Hall, Upper Saddle River, NJ, 1999.
18. Jang, J. S. R. *ANFIS: Adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, 1993, pp. 665-685.
19. http://www.cs.colostate.edu

Table 1: Neural Network parameters obtained from Co-operative PSO module

Dataset	Inputs	Swarm 1		Swarm 2		Swarm 3	
		w	b	W	b	w	b
Mackey-Glass	y(k)	-0.275	-1.085	-0.261	-0.878	-0.563	-0.030
	y(k-6)	0.640	-0.140	-0.328	-1.062	-0.065	-0.720
	y(k-12)	-2.387	-0.293	-4.409	1.093	-0.904	-6.045
	y(k-18)	-10.920	5.524	-1.007	0.406	-9.120	4.869
Box Jenkins	y(k-1)	-2.833	-3.518	-1.919	0.958	-0.676	0.337
	u(k-4)	-0.857	0.399	-1.384	3.272	-2.508	7.972
EEG	y(k-1)	0.322	-0.477	-0.161	0.333	0.023	-0.327
	y(k-2)	-0.245	0.517	-0.303	0.510	0.218	-0.511
	y(k-4)	-0.945	-1.225	1.352	1.648	-0.035	1.777
	y(k-8)	15.000	-7.473	15.000	-7.515	15.000	-7.561

Table 2: Comparison of Prediction performance between COPSO-MSN and ANFIS

Dataset	Training/ Test data	COPSO-MSN				ANFIS	
		Training time (sec)	NRMSE			Training time (sec)	NRMSE
			Swarm 1	Swarm 2	Swarm 3		
Mackey-Glass	Training	52.86	0.3223	0.5121	0.3651	5.34	0.0064
	Test		0.3243	0.5209	0.3621		0.0064
Box-Jenkins	Training	18.33	0.2151	0.2150	0.2150	0.20	0.0374
	Test		0.3416	0.3390	0.3416		0.0640
EEG	Training	91.54	0.5378	0.5364	0.5357	8.63	0.1565
	Test		0.5762	0.5724	0.5618		0.2189

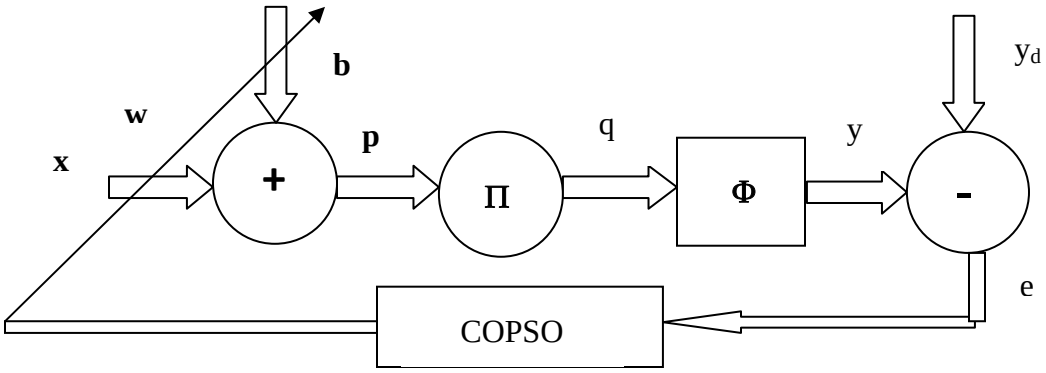

Fig. 1 Structure of single multiplicative neuron model with COPSO laerning (COPSO-SMN)

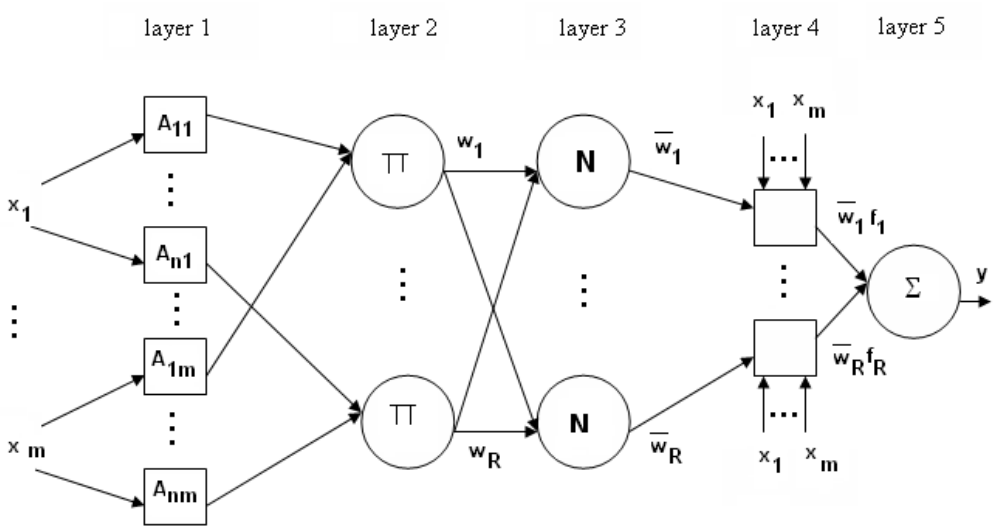

Fig. 2 Basic structure of ANFIS

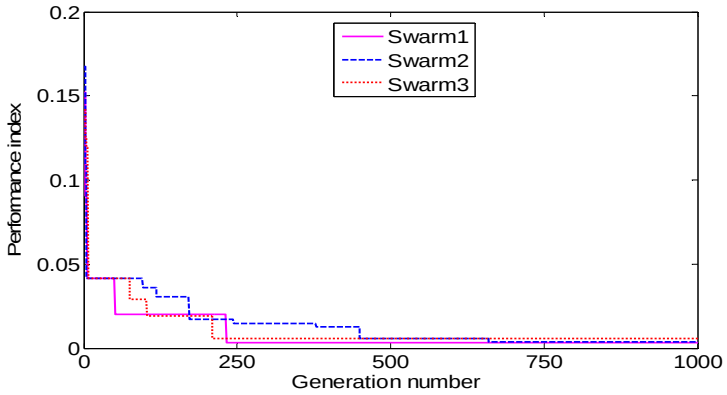

Fig. 3 Variation of performance index of sub-swarms

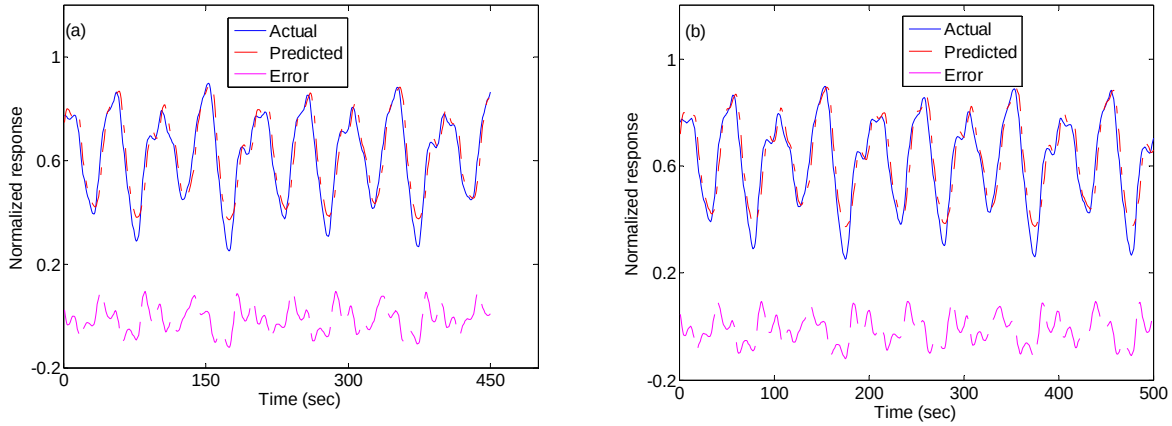

Fig. 4 Prediction of Mackey-Glass time series using COPSO-MSN (a) training, (b) test

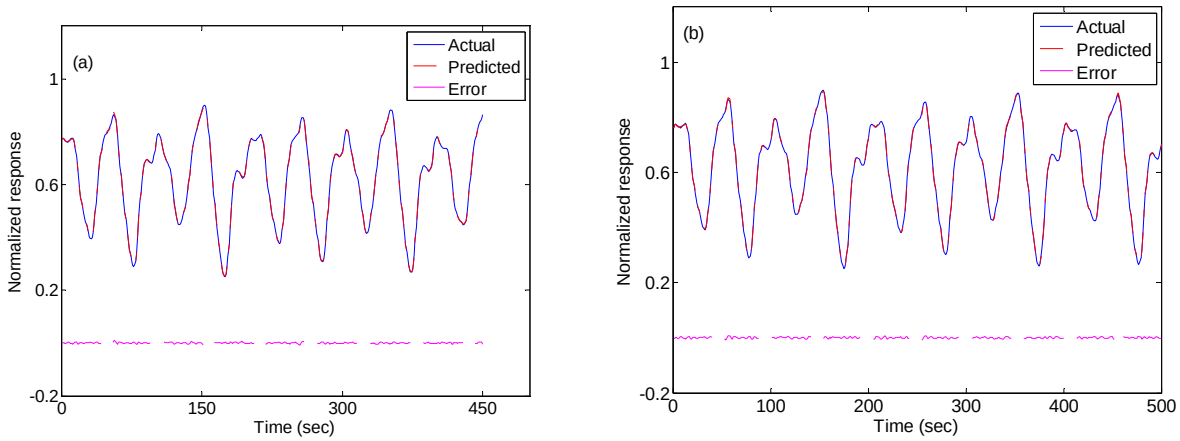

Fig. 5 Prediction of Mackey-Glass time series using ANFIS (a) training, (b) test

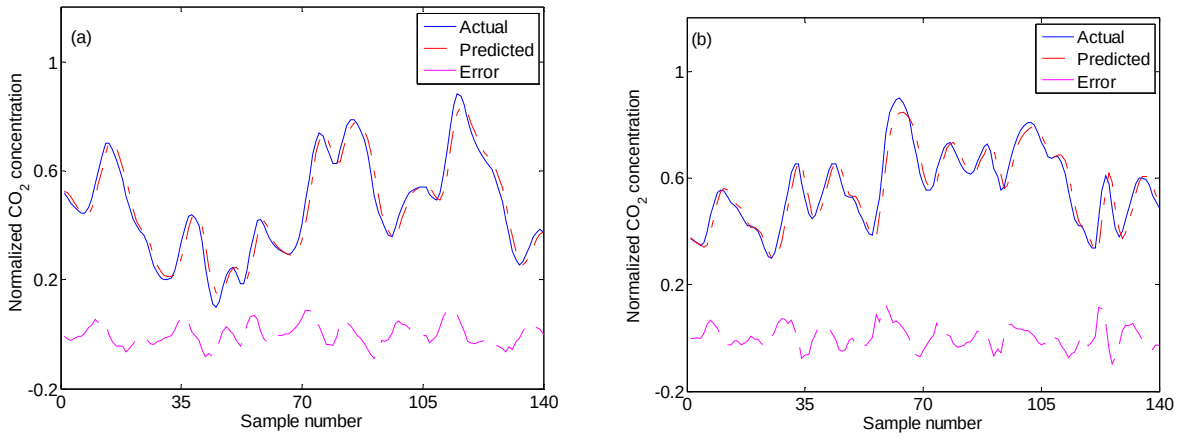

Fig. 6 Prediction of Box Jenkins time series using COPSO-SMN (a) training, (b) test

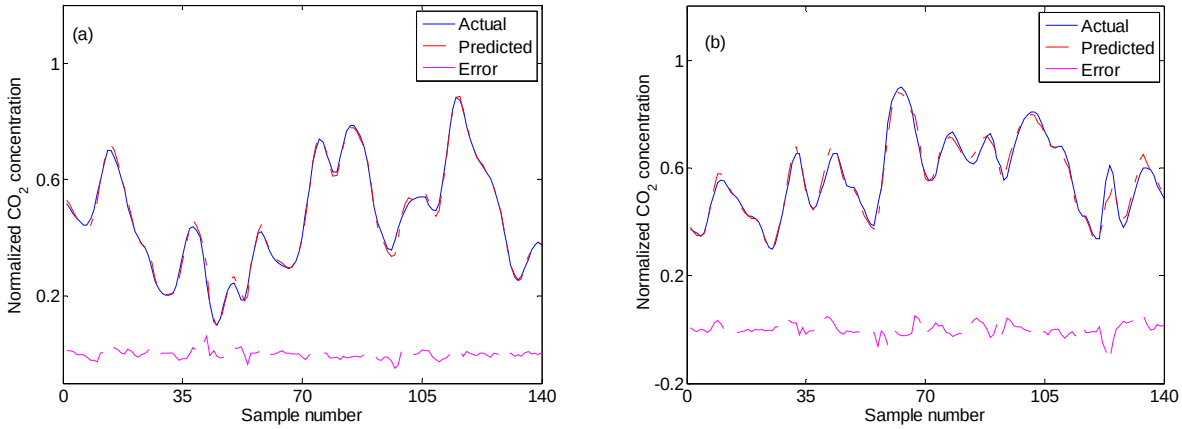

Fig. 7 Prediction of Box Jenkins time series using ANFIS (a) training, (b) test

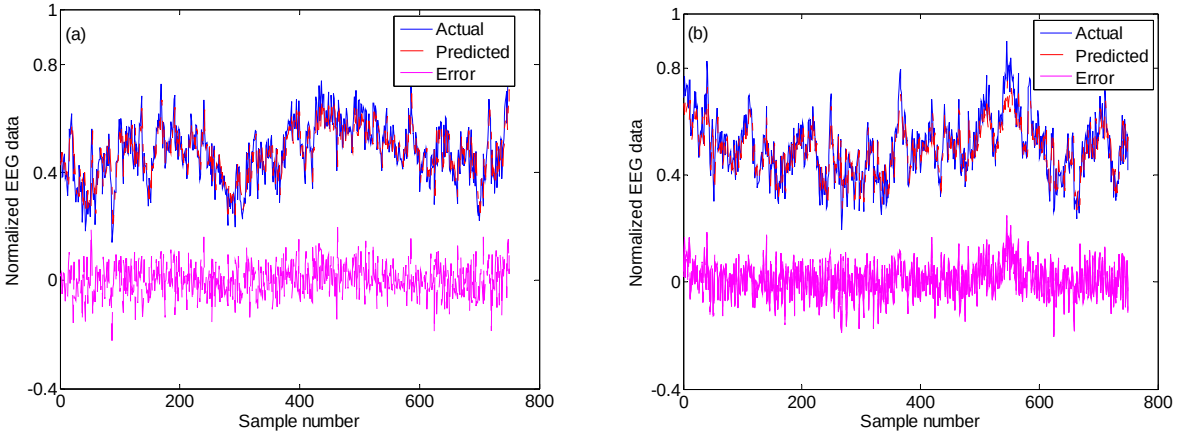

Fig. 8 Prediction of EEG time series using COPSO-SMN (a) training, (b) test

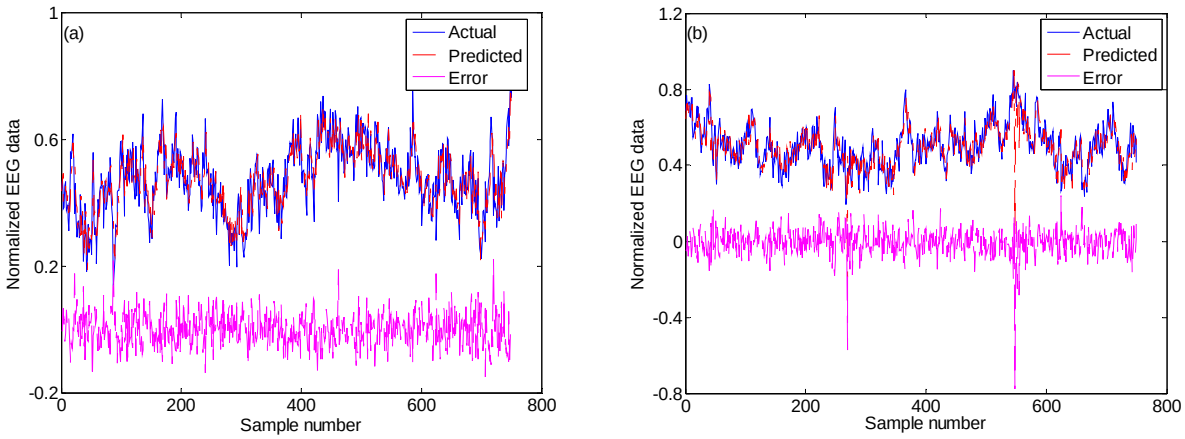

Fig. 9 Prediction of EEG time series using ANFIS (a) training, (b) test