Expert Systems with Applications 38 (2011) 11406–11411

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Prediction of chaotic time series using computational intelligence

B. Samanta *

Department of Mechanical Engineering, Villanova University, Villanova, PA 19085, USA

ARTICLE INFO

Keywords:
Time series prediction
Single multiplicative neuron model
Computational intelligence
Particle swarm optimization
Nonlinear time series
Biomedical signal analysis

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.

1. 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 (Box, Jenkins, & Reinse, 1994; De Gooijer & Hyndman, 2006; Mackey & Glass, 1997). 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 (Haykin, 1999; Kennedy & Eberhart, 1995; Kennedy, Eberhart, & Shi, 2001; Poli, Kennedy, & Blackwell, 2007; Samanta & Nataraj, 2009, 2008).

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 (Haykin, 1999). 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 (Koch, 1997; Koch & Segev, 2000). The SMN model is much simpler in structure than the more conventional multi-layer ANN and can offer better performances, if properly trained (Herz, Gollisch, Machens, & Jaeger, 2006; Schmitt, 2001). 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 (1995) 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 (Kennedy et al., 2001; Poli et al., 2007). The popularity of PSO is growing with applications in diverse fields of engineering, biomedical and social sciences, among others (Poli et al., 2007; Samanta & Nataraj, 2009, 2008).

In the present work, the SMN model parameters have been estimated using PSO (Yadav, Kalra, & John, 2007; Zhao & Yang, 2009). A variation of PSO with co-operative sub-swarms, COPSO, has been used in this work. 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 (Yen & Langari, 1999; Zadeh, 1965). 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,

* Current address: Department of Mechanical and Electrical Engineering, Georgia Southern University, P.O. Box 8045, Statesboro, GA 30460, USA.

E-mail addresses: bsamanta@georgiasouthern.edu, bsamanta@ieee.org.

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.03.013

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 by Jang (1993), 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 paper, 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) (http://www.cs.colostate.edu). 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 2 briefly discusses the SMN model. In Section 3, the basic PSO algorithm is presented. A brief discussion on ANFIS is presented in Section 4. Section 5 presents the results and conclusions are in Section 6.

2. Single multiplicative neuron (SMN) model

Fig. 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 $\mathbf{x}$. 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 $\mathbf{p}$ goes through the multiplication node and gets transformed to y through the nonlinear function of *logsig* as follows:

$$\mathbf{p} = \mathbf{W}\mathbf{x} + \mathbf{b}, \tag{1}$$

$$q = \prod_{i}^{n} p_i, \tag{2}$$

$$y = \frac{1}{1 + e^{-q}}, \tag{3}$$

$$e = y_d - y, \tag{4}$$

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).

3. Particle swarm optimization (PSO)

3.1. Standard particle swarm optimization (PSO)

In this section, a brief introduction to PSO algorithm is presented, for details text (Kennedy et al., 2001) can be referred to. Recent overviews of PSO and its variants are presented in Poli et al. (2007). 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 ith particle ($i = 1, m$) in the jth direction ($j = 1, n$). The fitness of a particle is assessed by calculating the value of the target or the 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. 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:

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

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

where $U(0,1)$ represents 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. In this work, $c_1 = 2.0$ and $c_2 = 2.0$ were used.

3.2. 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 Eq. (5) is rewritten as follows:

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

where $l = 1, \ldots, s$, 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.

3.3. 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.

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

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 (8). A population size

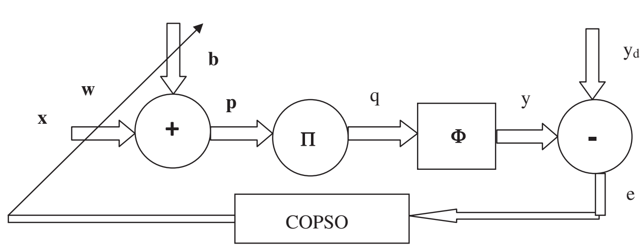

Fig. 1. Structure of single multiplicative neuron model with COPSO laerning (COPSO-SMN).