Neurocomputing ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Neurocomputing

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

Online sequential extreme learning machine with kernels for nonstationary time series prediction

Xinying Wang, Min Han *

Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, Liaoning, China

ARTICLE INFO

Article history:
Received 31 October 2013
Received in revised form
6 March 2014
Accepted 27 May 2014
Communicated by H. Zhang

Keywords:
Online
Time series
Extreme learning machine
Support vector machine
Nonstationary

ABSTRACT

In this paper, an online sequential extreme learning machine with kernels (OS-ELMK) has been proposed for nonstationary time series prediction. An online sequential learning algorithm, which can learn samples one-by-one or chunk-by-chunk, is developed for extreme learning machine with kernels. A limited memory prediction strategy based on the proposed OS-ELMK is designed to model the nonstationary time series. Performance comparisons of OS-ELMK with other existing algorithms are presented using artificial and real life nonstationary time series data. The results show that the proposed OS-ELMK produces similar or better accuracies with at least an order-of-magnitude reduction in the learning time.

1. Introduction

Time series prediction has played a crucial role in the development of techniques for dynamic system modeling, and nonlinear time series prediction has got more and more attention [1,2], for most practical situations, nonlinear signal processing is needed. Support vector machines [3–5], neural networks [6–8], and other machine learning methods [9–11] have been applied to predict time series. Most of the published papers were concerned with stationary time series other than the nonstationary time series. However, practical time series is almost nonstationary, which restricts the stationary methods. Consequently, the research of nonstationary time series prediction has been increasing important [12–14].

According to Takens' embedding theorem [15], a time series can be reconstructed into the phase space by the delay coordinate, and the reconstruction translates time correlation to spatial correlation. With the universal approximation capability, neural networks are able to approximate the spatial correlation effectively. However, gradient-based learning algorithms, which are commonly used in traditional neural networks, converge slowly and are easy to be trapped in local minimum [16,17]. Many learning methods have been developed or improved to speed up the training of neural networks. Among these methods, extreme learning machine (ELM) [18,19] has been one of the most efficient learning methods. ELM is developed for single hidden layer feedforward neural networks. In ELM, the weights connecting the input layer and the hidden layer, and the bias values of the hidden layer are randomly generated before learning and are left fixed during the learning process. At the same time, the weights connecting the hidden layer and the output layer are computed analytically. ELM has overcome the disadvantages of traditional neural networks, and has been successfully applied to regression [20,21], classification [22,23] and time series prediction [24,25].

The original and most variants of ELM are essentially batch learning, which still require elaborate and costly operations, limiting their applicability in real-time or nonstationary situations. Some online types of ELMs have been developed to deal with the problems where training samples are received one-by-one or chunk-by-chunk [26,27]. Moreover, in order to deal with the nonstationary situation, OS-ELM-TV [14] and LAFF-OS-ELM [13] are proposed. However, the optimal number of hidden nodes and basis function also should be determined beforehand by users. On the other hand, a special batch variant of ELM, extreme learning machine with kernels (ELMK) [28], uses unknown kernel mappings instead of known hidden layer mappings (resulting in no need to select the number of hidden nodes) and has been verified to have similar or better prediction performance. On the basis of the aforementioned analysis, in this paper, an online sequential learning algorithm is developed for extreme learning machine with kernels (ELMK) and the resulting model is referred as

* Corresponding author.
E-mail addresses: xinying@mail.dlut.edu.cn (X. Wang), minhan@dlut.edu.cn (M. Han).

http://dx.doi.org/10.1016/j.neucom.2014.05.068
0925-2312/

OL-ELMK, which can learn the training samples one-by-one or chunk-by-chunk. In order to deal with the nonstationary problem, an decremental algorithm is developed for OS-ELMK to remove the trained samples, and a fixed memory prediction scheme is designed to save more computational load and to further improve the prediction performance.

The rest of this paper is organized as follows. Section 2 gives a brief introduction of ELM and ELMK. Section 3 presents the online sequential learning algorithm and the fixed memory prediction scheme. In Section 4, simulation results of three artificial and real life examples are given. Finally, in Section 5, conclusions are drawn.

2. Preliminary

2.1. Extreme learning Machine

ELM is a single hidden-layer feedforward neural network, and has a simple three layer structure: input layer, output layer, and hidden layer which contains a large number of nonlinear processing nodes. The weights connecting the input layer to the hidden layer, and the bias values within the hidden layer are randomly generated and maintained throughout the learning process. The weights connecting the hidden nodes and the output nodes are computed analytically.

For N training samples $(\boldsymbol{x}_i, t_i)$, where $\boldsymbol{x}_i \in \Re^p$ and $t_i \in \Re$, ELM can be mathematically formulated as

$$\sum_{i=1}^{L} w_i g(W_{in(i)} \cdot \boldsymbol{x}_j + b_i) = o_j, \quad j = 1, \ldots, N. \tag{1}$$

where $W_{in(i)} \in \Re^p$ is the weight vector connecting the input nodes to the ith hidden node, $W_{in(i)} \cdot \boldsymbol{x}_j$ denotes the inner product of $W_{in(i)}$ and $\boldsymbol{x}_j$, $b_i \in \Re$ is the bias of the ith hidden node, $g(\cdot)$ is the activation function, $w_i \in \Re$ is the weight connecting the ith hidden node to the output node, $o_j \in \Re$ is the output of ELM, and L is the number of the hidden nodes. $g(\cdot)$ can be any infinitely differential function. Eq. (1) can be further expressed by the following matrix–vector form:

$$H\boldsymbol{w} = \boldsymbol{o}. \tag{2}$$

where

$$H = \begin{bmatrix} g(W_{in(1)} \cdot \boldsymbol{x}_1 + b_1) & \cdots & g(W_{in(L)} \cdot \boldsymbol{x}_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(W_{in(1)} \cdot \boldsymbol{x}_N + b_1) & \cdots & g(W_{in(L)} \cdot \boldsymbol{x}_N + b_L) \end{bmatrix}_{N \times L},$$

$\boldsymbol{o} = [o_1, \ldots, o_N]^T$ and $\boldsymbol{w} = [w_1, w_2, \ldots, w_L]^T$. Matrix H is called the hidden layer output matrix of ELM. The ith row of H, $\boldsymbol{h}_i$ is the hidden layer output vector with respect to inputs $\boldsymbol{x}_i$. Since in the ELM learning framework, $W_{in(i)}$ and b_i are randomly chosen beforehand, $\boldsymbol{h}_i$ is only related to the inputs.

If the ELM model with L hidden nodes can learn these N training samples with no residuals, then it means that there exist w_i so that

$$\sum_{i=1}^{L} w_i g(W_{in(i)} \cdot \boldsymbol{x}_j + b_i) = t_j, \quad j = 1, \ldots, N. \tag{3}$$

where t_j is the target value.

Eq. (3) can be written compactly in the matrix–vector form as

$$H\boldsymbol{w} = \boldsymbol{t} \tag{4}$$

where $\boldsymbol{t} = [t_1, \ldots, t_N]^T$ is the target vector. As the input weights and the hidden layer bias have been randomly chosen in the beginning of learning, (4) becomes a linear parameter system, and the smallest norm least squares' solution of the linear parameter system is

$$\boldsymbol{w} = H^{\dagger}\boldsymbol{t} \tag{5}$$

where $H^{\dagger}$ is the Moore–Penrose generalized inverse of H.

2.2. Extreme learning machine with kernels

The training process of ELM is a simple linear regression, which can effectively overcome the inherent flaws of traditional neural network [19,18]. However, the number of the hidden layer nodes, which is an important parameter of ELM crucial to the performance of prediction model, usually should be selected by some time-consuming methods according to the learning tasks [29,25]. Avoiding the hidden nodes' selection problem, the extreme learning machine with kernels (ELMK) is developed [7,28], which replaces the hidden layer mapping $\boldsymbol{h}(x)$ in extreme learning machine by the kernel function mapping $\phi(\boldsymbol{x})$ in the support vector machine. Consequently, the hidden layer mapping can be unknown.

The kernel matrix of ELM can be defined as follows:

$K_{ELM} = HH^T$:

$K_{ELM(i,j)} = \boldsymbol{h}(\boldsymbol{x}_i) \cdot \boldsymbol{h}(\boldsymbol{x}_j) = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

As a result, the output function can be written as

$$f(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})H^T\left(HH^T + \frac{I}{C}\right)^{-1}\boldsymbol{t} = \begin{bmatrix} K(\boldsymbol{x}, \boldsymbol{x}_1) \\ \vdots \\ K(\boldsymbol{x}, \boldsymbol{x}_N) \end{bmatrix}^T \left(K_{ELM} + \frac{I}{C}\right)^{-1}\boldsymbol{t}$$

The hidden layer mapping in the special kernel implementation of ELM can be unknown, but the corresponding kernel is usually given. Therefore, there is no longer needed to identify the number of hidden nodes (the dimension of the hidden layer feature space).

Given a training set $(\boldsymbol{x}_i, t_i), i = 1, \ldots, N$, where $\boldsymbol{x}_i \in \Re^P$, and $t_i \in \Re$. The original optimization problem of ELMK can be written as

$$\begin{aligned} \min \quad & L_P = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 \\ \text{s.t.} \quad & \phi(\boldsymbol{x}_i)^T\boldsymbol{w} = t_i - \xi_i \end{aligned} \tag{6}$$

where $\boldsymbol{w}$ is a vector in the feature space $\mathbf{F}$, and $\phi(\boldsymbol{x})$ maps the input $\boldsymbol{x}$ to a vector in $\mathbf{F}$. C is the regularization parameter, and ξ is the error. Here, we use $\phi(\boldsymbol{x})$ instead of $\boldsymbol{h}(\boldsymbol{x})$ in order to keep consistent with the support vector machine and explicitly indicate that the mapping is unknown. From the analysis of Section 2.1, we know that the mapping $\boldsymbol{h}(\cdot)$ or $\phi(\cdot)$ is only relative to the inputs.

The corresponding Lagrangian dual problem can be formatted as

$$L_D = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\xi_i^2 - \sum_{i=1}^{N}\theta_i(\phi(\boldsymbol{x}_i)^T\boldsymbol{w} - t_i + \xi_i) \tag{7}$$

where θ_i denotes the ith Lagrangian multiplier.

The KKT optimality conditions of (7) are as follows:

$$\frac{\partial L_D}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{N}\theta_i\phi(\boldsymbol{x}_i) = 0 \rightarrow \boldsymbol{w} = \sum_{i=1}^{N}\theta_i\phi(\boldsymbol{x}_i) \tag{8}$$

$$\frac{\partial L_D}{\partial \xi_i} = C\xi_i - \theta_i = 0 \rightarrow \theta_i = C\xi_i, \quad i = 1, \ldots, N \tag{9}$$

$$\frac{\partial L_D}{\partial \theta_i} = \phi(\boldsymbol{x}_i)^T\boldsymbol{w} - t_i + \xi_i = 0, \quad i = 1, \ldots, N \tag{10}$$

Please cite this article as: X. Wang, M. Han, Online sequential extreme learning machine with kernels for nonstationary time series prediction, Neurocomputing (2014), http://dx.doi.org/10.1016/j.neucom.2014.05.068