Applied Soft Computing xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

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

A self-organizing cascade neural network with random weights for nonlinear system modeling

Fanjun Li[a,b,c], Junfei Qiao[a,c,*], Honggui Han[a,c], Cuili Yang[a,c]

[a] College of Electronic Information and Control Engineering, Beijing University of Technology, Beijing 100124, China
[b] School of Mathematical Science, University of Jinan, Shandong 250022, China
[c] Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing 100124, China

A R T I C L E   I N F O

Article history:
Received 15 May 2015
Received in revised form
31 December 2015
Accepted 10 January 2016
Available online xxx

Keywords:
Feedforward neural network
Self-organizing cascade neural network
Nonlinear system modeling
Random weights
Wastewater treatment plant

A B S T R A C T

In this paper, a self-organizing cascade neural network (SCNN) with random weights is proposed for nonlinear system modeling. This SCNN is constructed via simultaneous structure and parameter learning processes. In structure learning, the units, which lead to the maximal error reduction of the network, are selected from the candidates and added to the existing network one by one. A stopping criterion based on the training and validation errors is introduced to select the optimal network size to match with a given application. In parameter learning, the weights connected with the output units are incrementally updated without gradients or generalized inverses, while the other weights are randomly assigned and no need to be tuned. Then, the convergence of SCNN is analyzed. Finally, the proposed SCNN is tested on two benchmark nonlinear systems and an actual municipal sewage treatment system. The experiment results show that the proposed SCNN has better performance on nonlinear system modeling than other similar methods.

1. Introduction

System models are crucial for effective analysis and control in practice [1]. A good model should adequately describe the behavior of a system. When no or partial knowledge of a system is available, results on system modeling are used to obtain the process variable trajectory information from a set of measured input–output data [2]. Most industrial systems are nonlinear in nature, hence nonlinear system modeling has attracted a lot of attention in many areas, such as control [3], process monitoring [4] and soft sensor [5,6].

In fact, it is a difficult task to model industrial systems because of the need to estimate the structure and parameters of such nonlinear systems [7]. Although the methods based on conventional mathematics have been widely used for nonlinear system modeling in recent decades [8,9], they often become not suitable when the industrial systems exhibit complicated characteristics in terms of strong nonlinearity, multivariable coupling, complex biochemical reaction, variations of operation conditions together with unknown model structure and parameters [10].

Theoretically, feedforward neural networks (FNNs) can approximate any continuous function defined on a compact set to any desired degree of accuracy [11,12]. This universal approximation property makes FNNs suitable to model nonlinear systems, especially those which are hard to be mathematically described [13]. Hence, FNNs have attracted a lot of attention for nonlinear system modeling [7,14–19]. For example, a novel self-organizing radial basis function neural network is proposed for nonlinear system identification and modeling [14]. An automatic axon–neural network (AANN) is proposed to establish an artificial neural network with self-organizing architecture and suitable learning algorithm for nonlinear system modeling [18]. A multiple neural network, which integrates the structure selection, parameter identification and hysteresis network switching with guaranteed neural identification performance, is introduced to model a kind of nonlinear systems [19]. In these scenarios noted above, the architecture of multilayer perceptron (MLP) has been employed, and the gradient-based algorithms as well. It is noted that the performance of FNNs depends on the network architecture and learning algorithms greatly [18]. Selecting suitable network architecture and fast learning algorithm for FNNs is still challenging for nonlinear systems modeling.

Recent research results show that the architecture of cascade neural networks (CNNs) is more powerful than that of MLP in most cases [20,21]. Although some algorithms have been proposed to

* Corresponding author at: College of Electronic Information and Control Engineering, Beijing University of Technology, Beijing 100124, China.
E-mail address: isibox@sina.com (J. Qiao).

http://dx.doi.org/10.1016/j.asoc.2016.01.028
1568-4946/© 2016 Published by Elsevier B.V.

train the CNNs with fixed topology, it is difficult for them to determine the number of hidden units to match with a given application [22,23]. Cascade correlation (CC) algorithm, one of the most popular constructive algorithms, can be used to automatically construct CNNs [24,25]. Instead of just adjusting the weights of a network with fixed topology, the CC algorithm begins with a minimal network with no hidden unit, then trains and adds new hidden units to the existing network one by one. Once a new hidden unit has been added, its input weights are frozen in the later process. Every time, only the new hidden unit and output units are trained. Each hidden unit added to the network has the maximal magnitude of the correlation between its output and the last error. However, this objective function (correlation) used to train and select new hidden units may result in a large network with poor generalization performance [26,27]. Hence, some improved objective functions have been evaluated in [26]. An orthogonal least squares based cascade network (OLSCN) has been proposed to construct CNNs by using a novel objective function, with which the sum of squared errors (SSE) of the network can be maximally reduced after a new hidden unit is added [27]. In these scenarios noted above, gradient-based algorithms have been used to optimize their objective functions. However, gradient-based algorithms suffer from local minima problem, slow convergence and very poor sensitivity to learning rate setting [28].

To overcome these difficulties faced by gradient-based algorithms, random vector functional link (RVFL) networks have been studied [28–31], where the weights between the input layer and the hidden layer can be randomly assigned and no need to be tuned. Then, the weights connected with the output units can be calculated by solving a linear regression problem. Such a randomized learning scheme dramatically reduces the training time and many experimental results indicate that the learner's generalization performance is favorably good [28–31]. It has been proved that RVFL networks can universally approximate any continuous function [29,32]. For its simplicity and effectiveness, this flat-net architecture with random weights has been successfully applied to many areas [33–36]. However, to the best of our knowledge, there are seldom similar algorithms for CNNs.

As a result of the research noted above, a self-organizing cascade neural network (SCNN) with random weights is proposed to model nonlinear systems. This proposed SCNN has a powerful growing cascade architecture, the connections of which can be across all the layers. Firstly, the mathematical expression of the growing architecture is presented, and the contribution value (CV) of units is derived by the orthogonal least squares (OLS) method to select new input and hidden units from candidates for SCNN. With the derived CV, each of the selected units affords the maximal reduction of the sum of squared errors (SSE). Secondly, the weights feeding into the output units are updated in an incremental way without gradients or generalized inverses during the training process, while the input weights of hidden units are randomly assigned and no need to be tuned. Thirdly, a stopping criterion based on training and validation errors is introduced to select the optimal network size. SCNN is a self-organizing network which automatically determines its weights and architecture to match with a given application. Finally, SCNN is test on two benchmark problems (Nonlinear dynamic system modeling and Mackey–Glass time-series prediction) and an actual municipal sewage treatment system in a wastewater treatment plant (WWTP). The experiment results show that SCNN can model nonlinear systems effectively.

The rest of this paper is organized as follows. The details of the proposed SCNN algorithm are described in Section 2, including the growing architecture, the derived contribution value of units, the method used to update the weights, the stopping criterion, the detailed steps and the proof of convergence. Section 3 presents the experiment results which demonstrate the performance of the proposed method compared with some existing algorithms. Finally, we conclude this work in the last section.

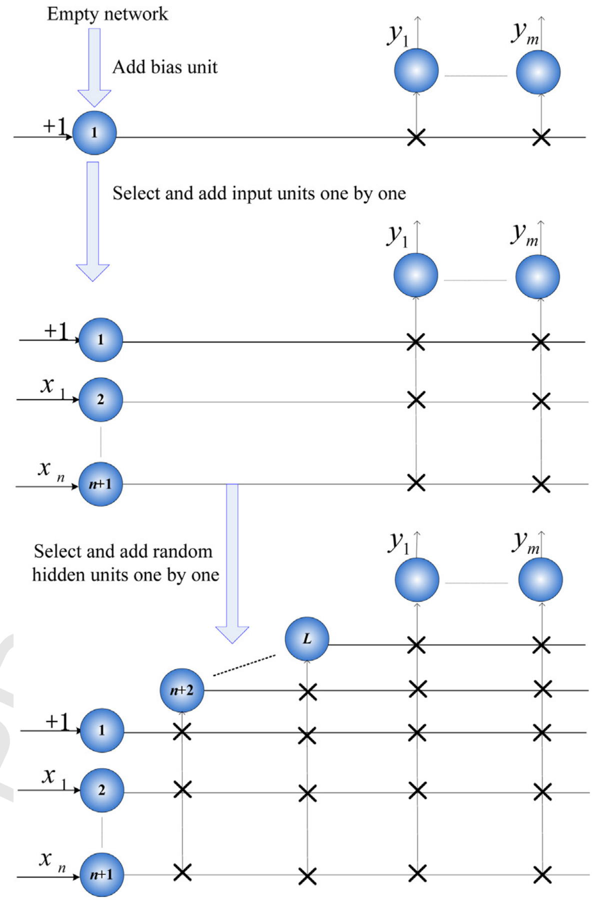

Fig. 1. The growing architecture of SCNN with random weights.

2. Self-organizing cascade neural network with random weights

2.1. Architecture of SCNN and its mathematical expression

As shown in Fig. 1, the architecture of SCNN is growing and similar to that of the cascade neural network [24,27], in which all hidden units connect with each other and the connections are across all the layers. Each hidden unit receives the connections from the input units as well as the pre-existing hidden units. All original inputs and the outputs of hidden units are fed to every output unit. Furthermore, there are some connections which connect the bias input (permanently set to +1) to all hidden and output units. It can be seen from Fig. 1 that the architecture of SCNN is growing step by step. Firstly, the bias unit with +1 is added to an empty network without input and hidden units, then all the input and hidden units are added one by one. To mathematically formulate the growing architecture, in the following, the bias unit, input units and hidden units are numbered in sequence as done in [27],

$$\text{the } j\text{th unit} = \begin{cases} \text{bias unit} & j = 1 \\ \text{the } (j-1)\text{th input unit} & 2 \le j \le n+1 \\ \text{the } (j-n-1)\text{th hidden unit} & j > n+1 \end{cases} \quad (1)$$

Please cite this article in press as: F. Li, et al., A self-organizing cascade neural network with random weights for nonlinear system modeling, Appl. Soft Comput. J. (2016), http://dx.doi.org/10.1016/j.asoc.2016.01.028