ISA Transactions ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

ISA Transactions

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

Adaptive control of nonlinear system using online error minimum neural networks☆

Chao Jia [a], Xiaoli Li [b,*], Kang Wang [a], Dawei Ding [a]

[a] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, P.R. China
[b] College of Electronic Information and Control Engineering, Beijing University of Technology, Beijing 100124, P.R. China

ARTICLE INFO

Article history:
Received 26 January 2016
Received in revised form
11 July 2016
Accepted 26 July 2016
This paper was recommended for publication by Jeff Pieper

Keywords:
ELM
Adaptive control
Neural networks
OS-ELM
EM-ELM

ABSTRACT

In this paper, a new learning algorithm named OEM-ELM (Online Error Minimized-ELM) is proposed based on ELM (Extreme Learning Machine) neural network algorithm and the spreading of its main structure. The core idea of this OEM-ELM algorithm is: online learning, evaluation of network performance, and increasing of the number of hidden nodes. It combines the advantages of OS-ELM and EM-ELM, which can improve the capability of identification and avoid the redundancy of networks. The adaptive control based on the proposed algorithm OEM-ELM is set up which has stronger adaptive capability to the change of environment. The adaptive control of chemical process Continuous Stirred Tank Reactor (CSTR) is also given for application. The simulation results show that the proposed algorithm with respect to the traditional ELM algorithm can avoid network redundancy and improve the control performance greatly.

1. Introduction

In recent years, neural network technology has been widely used in data classification, fitting, modeling, control and many other fields [1–6], especially in the field of adaptive control. Adaptive control method which is based on neural networks has been extensively studied. However, due to limitations of neural network, it is too difficult to be widely used in actual production. In the early days, SLFNs (Single-hidden Layer Feedforward Neural networks) represented by BP neural network are the first one to be known and also widely applied to identification and modeling [7]. BP algorithm needs error back propagation, so they are at a slow learning rate. At the same time, it is easy for them to fall into a local minimum. In 2006, Huang et al. proposed a learning machine concept (Extreme Learning Machines, ELM) [8]. Compared with conventional SLFNs, Extreme Learning Machine is a new single hidden layer of the neural network and widely used in machine learning, data classification, forecasting and many other field [9,10]. It can randomly select the network input layer weights while training by only adjust weights between the hidden layer and the output layer. By this way, the multiple iterations of the training process for parameters in traditional neural network parameters are converted into the solvation linear equations. The entire training process can be achieved, at the time to adjust the network parameters can be reduced greatly.

Training of batch processing ELM algorithms can be realized only after all the training samples are ready. But in actual application, the training data may come one by one or chunk by chunk. Therefore, Hung et al. proposed the OS-ELM (Online Sequential ELM) [11] algorithm which can adjust its output weight correspondingly when new observations are received. Unlike the batch learning algorithms, as soon as the learning process for actual training data is complemented, the data will be discarded. Thus it avoids retraining of the previous observations. Furthermore, there is no index guiding mechanism in the ELM and OS-ELM neural network. The improper selection of hidden layer nodes will lead to lose learning ability of neural network. Therefore, Huang et al. proposed the ELM neural network learning algorithm based on minimum error named EM-ELM (Error Minimized ELM) [12]. The algorithm can add the number of hidden nodes as needed as possible, avoid the redundancy of the network, and evaluate the network structure constantly, make the network gradually to meet application needs.

On one hand, in the actual production process, data are often got online. On the other hand, self-adjustment structure of the neural networks is also expected. Considering these requirements, a learning algorithm who combines the advantage of OS-ELM

☆This work is partially supported by the Fund of National Natural Science Foundation of P.R. China (Grant nos. 61473034, 61104013), Specialized Research Fund for the Doctoral Program of Higher Education (20130006110008), the Start-up Funding of Content development and the Introduced Talent Research, Beijing Nova Programme Interdisciplinary Cooperation Project.

* Corresponding author.
E-mail addresses: lixiaoli@hotmail.com, lixiaolibjut@bjut.edu.cn (X. Li).

http://dx.doi.org/10.1016/j.isatra.2016.07.012
0019-0578/© 2016 Published by Elsevier Ltd. on behalf of ISA.

algorithm with the benefit of EM-ELM algorithm which is named OEM-ELM (Online Error Minimization Extreme Learning Machines) algorithm is proposed. The core ideas of this OEM-ELM algorithm are as follows: online learning, evaluation of network performance, and the increase of the number of hidden nodes as needed as possible. First, it is necessary to execute OS-ELM algorithm learning. Then in order to evaluate whether the network structure meets the needs or not, an evaluation of the current network structure has to be done after the network has been studied for a certain stage. The analysis will go on if the data meet the demand of accuracy. Otherwise the EM-ELM algorithm will be proceeded to add network nodes. In this paper, our interest will be kept in the identification and control of nonlinear dynamic plants by using OEM-ELM neural networks. As we known, over the past few years, there has been a variety of derivative ELM neural network structure, such as A-ELM [13], PR-ELM [14], ODOC-ELM [15], MI-ELM [16] and so on. Compared with the above algorithm, OEM-ELM algorithm we proposed in this paper has a significant difference and innovation. On one hand, OEM-ELM algorithm is a kind of on-line dynamic adjustment algorithm of neural node which is based on OS-ELM and EM-ELM algorithm. On the other hand, according to the algorithm proposed in this paper, the OEM-ELM adaptive controller is designed, which is applied to the adaptive control of the dynamic adjustment of the neural network structure algorithm.

Meanwhile, the proposed algorithm is applied to a chemical process CSTR. The simulation results in Matlab R2012 demonstrate that the proposed algorithm with respect to the traditional ELM algorithm can avoid network redundancy and can also improve the identification accuracy. The motivation of this paper is: for the traditional neural network, the structure of neural network (the number of hidden nodes) and initial parameters will no longer be adjusted again once given. To the best of our knowledge, there are only few of published results regarding as changing the structure of neural network during the identification. In this paper, we attempt to propose a novel neural network learning algorithm used to identification and adaptive control. We attempt to provide some ideas of variable structure neural network adaptive control.

This paper is structured as follows: first, OS-ELM neural network and adaptive control based on OS-ELM neural network are briefly introduced in Section 2. Then, OEM-ELM algorithm and adaptive control of nonlinear dynamic system by using OEM-ELM are stated in Section 3. Simulation results for OEM-ELM control in CSTR are also shown in Section 4. Finally some conclusions are brought in Section 5.

2. OS-ELM neural network and adaptive control based on OS-ELM neural network

2.1. OS-ELM neural network

For the SLFNs, the structure can be found in Fig. 1. In this section we will introduce the basic principle of the OS-ELM algorithm [11]:

OS-ELM Algorithm:

Given the initial condition as follows,

Training set: $S=\{(\mathbf{x}_i,\mathbf{t}_i)|\mathbf{x}_i\in\Re^n,\mathbf{t}_i\in\Re^m,i=1,\ldots\}$.

Activation function: $G(\mathbf{a}_i,b_i,\mathbf{x})$.

Hidden node number: L.

Step 1. **Initialization phase**:

A small chunk of training data $S_0=\{(\mathbf{x}_i,\mathbf{t}_i)\}_{i=1}^{N_0}$ is used to initialize the learning, where $N_0\ge L$.

(a) Assign random parameters of hidden nodes $(\mathbf{a}_i,b_i)$, where $i=1,\ldots,L$.

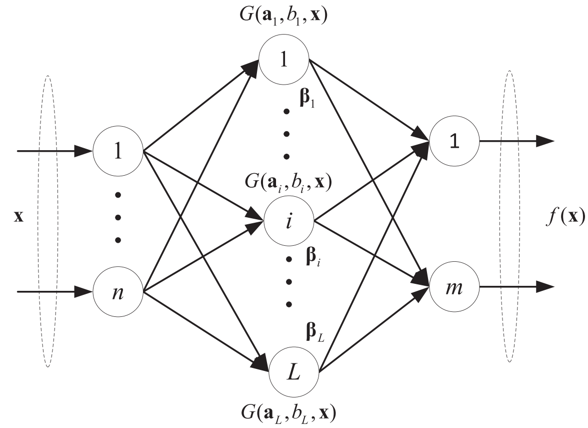

Fig. 1. The structure of single-hidden layer feedforward neural networks.

(b) Calculate the initial hidden layer output matrix $\mathbf{H}_0$.

$$\mathbf{H}_0=\begin{bmatrix} G(\mathbf{a}_1,b_1,\mathbf{x}_1)\cdots G(\mathbf{a}_L,b_L,\mathbf{x}_1)\\ \vdots \quad \cdots \quad \vdots\\ G(\mathbf{a}_1,b_1,\mathbf{x}_{N_0})\cdots G(\mathbf{a}_L,b_L,\mathbf{x}_{N_0})\end{bmatrix}_{N_0\times L}$$

(c) Calculate the initial output weight $\beta_0=\mathbf{P}_0\mathbf{H}_0^T\mathbf{T}_0$, where $\mathbf{P}_0=(\mathbf{H}_0^T\mathbf{H}_0)^{-1}$ and $\mathbf{T}_0=[\mathbf{t}_1\ \ldots\ \mathbf{t}_{N_0}]^T$.

(d) Set k=0, where k is the number of chunks which is trained currently.

Step 2. **Sequential Learning phase**:

(a) Present the $(k+1)$ th chunk of new observations $S_{k+1}=\{(\mathbf{x}_i,\mathbf{t}_i)\}_{i=\left(\sum_{j=0}^{k}N_j\right)+1}^{\sum_{j=0}^{k+1}N_j}$, here N_{k+1} denotes the number of observations in the $(k+1)$ th chunk.

(b) Calculate the partial hidden layer output matrix $\mathbf{H}_{k+1}$ for the $(k+1)$ th chunk of data S_{k+1}:

$$\mathbf{H}_{k+1}=\begin{bmatrix} G\left(\mathbf{a}_1,b_1,\mathbf{x}_{\left(\sum_{j=0}^{k}N_j\right)+1}\right) & \ldots & G\left(\mathbf{a}_L,b_L,\mathbf{x}_{\left(\sum_{j=0}^{k}N_j\right)+1}\right)\\ \vdots & \ldots & \vdots\\ G\left(\mathbf{a}_1,b_1,\mathbf{x}_{\sum_{j=0}^{k+1}N_j}\right) & \ldots & G\left(\mathbf{a}_L,b_L,\mathbf{x}_{\sum_{j=0}^{k+1}N_j}\right)\end{bmatrix}$$

Set $\mathbf{T}_{k+1}=\left[\mathbf{t}_{\left(\sum_{j=0}^{k}N_j\right)+1},\ \ldots,\ \mathbf{t}_{\sum_{j=0}^{k+1}N_j}\right]^T$

(c) Calculate the output weight

$$\mathbf{P}_{k+1}=\mathbf{P}_k-\mathbf{P}_k\mathbf{H}_{k+1}^T(\mathbf{I}+\mathbf{H}_{k+1}\mathbf{P}_k\mathbf{H}_{k+1}^T)^{-1}\mathbf{H}_{k+1}\mathbf{P}_k \tag{1}$$

$$\beta_{k+1}=\beta_k+\mathbf{P}_{k+1}\mathbf{H}_{k+1}^T(\mathbf{T}_{k+1}-\mathbf{H}_{k+1}\beta_k) \tag{2}$$

(d) Set $k=k+1$. Go to step 2(a).

Detail description can be referred to Huang [11]. From (1) and (2), it can be seen that the OS-ELM algorithm is similar to recursive least-squares (RLS) in [17]. Hence, all the convergence results of RLS can be applied here.

2.2. Adaptive control based on OS-ELM neural network

The adaptive control problem using OS-ELM neural networks can be described in this section. The controlled system can be described as follows:

$$y_{k+1}=f_0[\mathbf{x}_k]+g_0[\mathbf{x}_k]u_k \tag{3}$$

where $\mathbf{x}_k=[y_{k-n+1},\ldots,y_k,u_{k-m},\ldots,u_{k-1}]$. Let us first assume that there exist exact weights $\mathbf{w}^*$ and $\mathbf{v}^*$, the functions $\hat{f}[\mathbf{x}_k,\mathbf{w}^*]$ and $\hat{g}[\mathbf{x}_k,\mathbf{v}^*]$ which can approximate the functions $f_0[\mathbf{x}_k]$ and $g_0[\mathbf{x}_k]$. The