Neurocomputing 175 (2016) 826–834

Contents lists available at ScienceDirect

Neurocomputing

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

Two-hidden-layer extreme learning machine for regression and classification

B.Y. Qu [a,b], B.F. Lang [a], J.J. Liang [a,*], A.K. Qin [c], O.D. Crisalle [a]

[a] School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China
[b] School of Electric and Information Engineering, Zhongyuan University of Technology, Zhengzhou 450007, China
[c] School of Computer Science and Information Technology, RMIT University, Melbourne, 3001 Victoria, Australia

ARTICLE INFO

Article history:
Received 28 May 2015
Received in revised form
9 November 2015
Accepted 9 November 2015
Available online 17 November 2015

Keywords:
Extreme learning machine
Two-hidden-layer
Regression
Classification
Neural network

ABSTRACT

As a single-hidden-layer feedforward neural network, an extreme learning machine (ELM) randomizes the weights between the input layer and the hidden layer as well as the bias of hidden neurons, and analytically determines the weights between the hidden layer and the output layer using the least-squares method. This paper proposes a two-hidden-layer ELM (denoted TELM) by introducing a novel method for obtaining the parameters of the second hidden layer (connection weights between the first and second hidden layer and the bias of the second hidden layer), hence bringing the actual hidden layer output closer to the expected hidden layer output in the two-hidden-layer feedforward network. Simultaneously, the TELM method inherits the randomness of the ELM technique for the first hidden layer (connection weights between the input weights and the first hidden layer and the bias of the first hidden layer). Experiments on several regression problems and some popular classification datasets demonstrate that the proposed TELM can consistently outperform the original ELM, as well as some existing multilayer ELM variants, in terms of average accuracy and the number of hidden neurons.

1. Introduction

Single-hidden-layer feedforward neural networks (SLFNs), one of the most popular neural network models [1,2], have a simple structure consisting of one input layer, one hidden layer, and one output layer. A wide range of applications have been used to demonstrate the efficacy of SLFNs [3,4]. However, these techniques suffer from a time-expensive training process that usually adopts gradient-based error back-propagation algorithms, and consequently is prone to getting stuck in local minima. To address this issue, in 2004 Huang et al. [3] proposed an extreme learning machine (ELM) technique aiming at reducing the computational costs incurred by the error back-propagation procedure during the training process. A distinguishing feature of ELMs is that both the connection weights from the input layer to the hidden layer and the hidden neurons' biases are randomly generated, instead of being iteratively learned as in conventional SLFNs. Moreover, the connection weights from the hidden layer to the output layer are analytically determined using the time-efficient least-squares method (LS) [5]. As a result, an ELM features remarkably fast training speed and outstanding generalization performance. The ELM approach has demonstrated its advantages in various fields of applications, including image recognition [6–10], power-load forecasting [11,12], wind speed forecasting [13], and protein structure prediction [14], among others. However, because of the random weights from the input layer to the hidden layer, as well as the random biases of the hidden neurons, the average accuracy of ELM variants is generally low, which calls for further investigation of better hidden-layer parameter calculation approaches.

Many ELM variants have been developed to improve specific aspects of the performance of the original algorithm. Examples include voting-based extreme learning machines (V-ELM) [15], regularized extreme learning machines (RELM) [16,17], evolutionary extreme learning machines (E-ELM) [18], online sequential extreme learning machines (OS-ELM) [19], fully complex extreme learning machines (Fully complex ELM) [4,20], sparse extreme learning machines (Sparse ELM) [21], kernel-based extreme learning machines [22], and pruned-extreme learning machines (P-ELM) [23], among others. However, the problem of how to achieve more satisfactory accuracy remains a challenge to overcome.

To achieve desirable accuracy improvements, we propose a two-hidden-layer extreme learning machine (TELM) algorithm, which adds a hidden layer to the single-hidden-layer ELM architecture, and utilizes a novel method to calculate the parameters

* Corresponding author. Tel.: +86 13526781788.
E-mail addresses: qby1984@hotmail.com (B.Y. Qu), langbofei@hotmail.com (B.F. Lang), liangjing@zzu.edu.cn (J.J. Liang), kai.qin@rmit.edu.au (A.K. Qin), crisalle@gmail.com (O.D. Crisalle).

http://dx.doi.org/10.1016/j.neucom.2015.11.009
0925-2312/

related to the second hidden layer (namely, connection weights between the first and second hidden layer and the bias of the second hidden layer). Based on previous research, two-hidden-layer feedforward neural networks (TLFNs) [24] typically require fewer hidden neurons than SLFNs to achieve a desired performance level. This is an initial basis for considering the two-hidden-layer structure proposed. The foundational ideas for the TELM algorithm are simpler to present by comparing and contrasting its features with other multilayer ELM algorithms.

First consider the hierarchical extreme learning machine (HELM) approach presented in [25], which is based on a hierarchical feedforward neural network (HFNN) structure consisting of two parts, where each part is comprised of one input layer, one hidden layer, and one output. It is therefore possible to regard the output of the first part as an input neuron in the second part. Unlike HELM, the proposed TELM contains only one output layer, and is specifically designed for training the parameters of the hidden layers. Furthermore, HELM is tailored to solving real-time or on-line prediction problems that involve a time-sequence dataset (such as predicting the water quality in a wastewater treatment processes, for example), whereas TELM as no such restriction on the type of training dataset.

Next, consider the multilayer extreme learning machine (ML-ELM) [26] and the alternative H-ELM advanced in [27]. Both techniques involve ELM-based auto-encoder schemes as their building blocks. In fact, this H-ELM method is an improvement over ML-ELM, as it features a sparse ELM auto-encoder for improved performance. Both schemes focus mainly on solving classification problems, as they are involved in feature extraction. In their mode of operation, previous hidden layers specialize on processing for feature extraction, whereas the last hidden layers are mostly intended for least-squares operations. The focus of the proposed TELM is different, as it seeks to obtain improved performance using a reduced number of hidden neurons. However, the TELM can also incorporate ELM-based auto-encoder techniques, hence making it a suitable alternative for seeking improved performance in feature extraction problems under scenarios that call for a reduced number of neurons.

The experimental results presented in this paper for several regression and classification problems demonstrate the superiority of TELM over the original ELM and also over other multilayer ELM variants in terms of average accuracy. Our experiments also investigate the different effect on regression and classification problems observed when using initial orthogonalization procedures applied to the parameters of the first hidden-layer (that is, connection weights between the input weights and the first hidden layer and the bias of the first hidden layer).

The rest of this paper is organized as follows: Section 2 presents a brief review of the original ELM, Section 3 describes the proposed TELM technique, Section 4 reports and analyzes experimental results, and finally, Section 5 draws key conclusions and also discusses future research plans.

2. Extreme learning machine

The ELM approach originally proposed by Huang et al. [3] aims at avoiding a time-consuming iterative training procedure and simultaneously improving the generalization performance. The idea is inspired by the biological thought that the human brain is a sophisticated system that can handle diverse tasks, day and night, without human intervention. Based on this reasoning, some researchers strongly support the idea that there must be some parts of the brain where the neuron configurations do not depend on the external environment [3,24,28–30]. The ELM algorithm takes advantage of this biological argument, and employs tuning-free neurons in the hidden layer to resolve the adverse issues encountered by the back-propagation [31] and Levenberg–Marquardt algorithms [32].

Consider N arbitrary distinct samples $(\boldsymbol{x}_i, \ \boldsymbol{t}_i)(i=1, \ 2, \ \dots, N)$, *i.e.*, there is an input feature $\mathbf{X}=[\boldsymbol{x}_1, \ \boldsymbol{x}_2, \ \dots, \ \boldsymbol{x}_N]^{\mathrm{T}}$ and a desired matrix $\mathbf{T}=[\boldsymbol{t}_1, \ \boldsymbol{t}_2, \dots, \boldsymbol{t}_N]^{\mathrm{T}}$ comprised of labeled samples, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]^{\mathrm{T}} \in \mathbb{R}^n$ and $\boldsymbol{t}_i = [t_{i1}, t_{i2}, \dots, t_{im}]^{\mathrm{T}} \in \mathbb{R}^m$, where the superscript "T" denotes the matrix/vector transposition. Let L denote the number of hidden neurons with activation function $g(\boldsymbol{x})$. The ELM method selects in a random way the input-weight matrix $\mathbf{W} = \left[\boldsymbol{W}_1, \boldsymbol{W}_2, \dots, \boldsymbol{W}_j\right]^{\mathrm{T}} \in \mathbb{R}^{L\times n}$ that links the input layer to the hidden layer, and the bias vector $\mathbf{B} = \left[\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_L\right]^{\mathrm{T}} \in \mathbb{R}^{L\times N}$ of the hidden-layer neurons. Furthermore, $\mathbf{W}$ and $\mathbf{B}$ are determined simultaneously, and they remain fixed during the training phase. This procedure allows transforming the original nonlinear neural-network system to a system described by the linear expression

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T} \tag{1}$$

where $\boldsymbol{\beta} = \left[\boldsymbol{\beta}_1, \ \boldsymbol{\beta}_2, \dots, \ \boldsymbol{\beta}_L\right]^{\mathrm{T}} \in \mathbb{R}^{L\times m}$ is the connection-weight matrix between the hidden layer and the output layer, with vector components $\boldsymbol{\beta}_j = \left[\boldsymbol{\beta}_{j1}, \ \boldsymbol{\beta}_{j2}, \ \dots, \ \boldsymbol{\beta}_{jm}\right]^{\mathrm{T}} (j=1,2,\dots,L)$ that denote the connection weights between the jth hidden neuron and m output neurons, $\mathbf{H} = g(\mathbf{W}\,\mathbf{X}+\mathbf{B}) \in \mathbb{R}^{N\times L}$ is the hidden layer output matrix whose scalar entries $h_{ij} = g(\boldsymbol{W}_j \boldsymbol{x}_i + \boldsymbol{b}_j)$ $(i=1, \ 2, \ \dots, \ N, j=1, \ 2, \ \dots, \ L)$ are interpreted as the output of the jth hidden neuron with respect to $\boldsymbol{x}_i$, $\boldsymbol{W}_j = \left[W_{j1}, W_{j2}, \dots, W_{jn}\right]^{\mathrm{T}}$ is the vector of connection weights between n input neurons and the jth hidden neuron, and where $\mathbf{b}_j$ is the bias of the jth hidden neuron. Finally, the matrix-vector product $\boldsymbol{W}_j\boldsymbol{x}_i$ is interpreted as the inner product between matrix $\boldsymbol{W}_j$ and vector $\boldsymbol{x}_i$.

The only parameter to be calculated in the ELM is the output-weights matrix $\boldsymbol{\beta}$. Using the least-squares method it follows that

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T} \tag{2}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose (MP) generalized inverse of matrix $\mathbf{H}$, which can be calculated using the orthogonal projection method. That is to say, if $\mathbf{H}^{\mathrm{T}}\mathbf{H}$ is nonsingular, then $\mathbf{H}^{\dagger} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{T}}$; otherwise $\mathbf{H}^{\dagger} = \mathbf{H}^{\mathrm{T}}\left(\mathbf{H}\mathbf{H}^{\mathrm{T}}\right)^{-1}$ when $\mathbf{H}\mathbf{H}^{\mathrm{T}}$ is nonsingular. A benefit of using the MP method of solution is that the above formula yields the solution vector $\boldsymbol{\beta}$ of the least two-norm when $\mathbf{H}\mathbf{H}^{\mathrm{T}}$ is nonsingular, a valuable advantage when recognizing that Bartlett [33] observes that smaller weights lead to improved generalization performance.

The implementation of the original ELM proceeds according to the following steps, given N training samples $(\boldsymbol{x}_i, \ \boldsymbol{t}_i)(i=1, \ 2, \dots, N)$ and L hidden neurons with activation function $g(\boldsymbol{x})$:

(i) Randomly assign the connection weights between the input layer and the hidden layer $\mathbf{W}$ and the bias of the hidden layer $\mathbf{B}$.
(ii) Calculate the hidden layer output matrix $\mathbf{H} = g(\mathbf{W}\,\mathbf{X}+\mathbf{B})$.
(iii) Obtain weights between the hidden layer and the output layer using the least-square method $\boldsymbol{\beta} = \mathbf{H}^{\dagger}\mathbf{T}$.

3. Two-hidden-layer extreme learning machine

In 1997 Tamura and Tateishi [34] demonstrated that two-hidden-layer feedforward networks (TLFNs) are superior to SLFNs in terms of the ability to use fewer hidden neurons to achieve the desired performance. They claimed that a TLFN with only $(N/2+3)$ hidden neurons can learn from N training samples to achieve any negligible training error. Huang [24] further demonstrates that by using $2\sqrt{(m+3)N}$ hidden neurons a TLFN can learn from N training samples to achieve an arbitrarily small training error. Such advantage of TLFNs motivates us to translate