Neurocomputing 194 (2016) 260–270

Contents lists available at ScienceDirect

Neurocomputing

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

Efficient Leave-One-Out Cross-Validation-based Regularized Extreme Learning Machine

Zhifei Shao, Meng Joo Er *

School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore

A R T I C L E I N F O

Article history:
Received 1 November 2015
Received in revised form
20 December 2015
Accepted 16 February 2016
Available online 4 March 2016

Keywords:
Extreme Learning Machine (ELM)
Regularized ELM (RELM)
Ridge regression
LOO-CV
Leave-One-Out Cross-Validation

A B S T R A C T

It is well known that the Leave-One-Out Cross-Validation (LOO-CV) is a highly reliable procedure in terms of model selection. Unfortunately, it is an extremely tedious method and has rarely been deployed in practical applications. In this paper, a highly efficient Leave-One-Out Cross-Validation (LOO-CV) formula has been developed and integrated with the popular Regularized Extreme Learning Machine (RELM). The main contribution of this paper is the proposed algorithm, termed as Efficient LOO-CV-based RELM (ELOO-RELM), that can effectively and efficiently update the LOO-CV error with every regularization parameter and automatically select the optimal model with limited user intervention. Rigorous analysis of computational complexity shows that the ELOO-RELM, including the tuning process, can achieve similar efficiency as the original RELM with pre-defined parameter, in which both scale linearly with the size of the training data. An early termination criterion is also introduced to further speed up the learning process. Experimentation studies on benchmark datasets show that the ELOO-RELM can achieve comparable generalization performance as the Support Vector Machines (SVM) with significantly higher learning efficiency. More importantly, comparing to the trial and error tuning procedure employed by the original RELM, the ELOO-RELM can provide more reliable results by the virtue of incorporating the LOO-CV procedure.

1. Introduction

A simple and fast learning algorithm for the Artificial Neural Networks (ANN) named Extreme Learning Machine (ELM) has been proposed in recent years [1]. Its salient feature is that the parameters of its hidden neurons can be randomly generated instead of being exhaustively tuned, thus saving a great deal of computing resources. It has been shown to achieve superior performance than traditional learning algorithms such as Backpropagation (BP) [2] and Support Vector Machines (SVM) [3]. In addition, the ELM also overcomes common issues faced in the ANN, such as defining the learning rate, number of epochs, stopping criterion and running into local minima [1,4,5]. Because of these appealing features of the ELM, it has been successfully and widely implemented [6–10]. The differences between the ELM and other ANNs based on randomness feature are elaborated in [9,11].

Various approaches have been used to enhance the performance of the original ELM, including ensemble approaches, where a group of ELMs are generated and the final output is combination of individual ELM results [12–16]; growing and pruning approaches, where the size of the ELM hidden layer grows or is pruned according to different criteria [5,4,17,18]; optimization approaches, where the parameters are adjusted using various techniques [19]; parsimonious structure, where the size of the hidden layer is pruned and restricted to achieve better generalization performance [20,21].

Recently, the combination of ELM with regularization methods to form the Regularized ELM (RELM) has become increasingly popular. The idea is to restrict the norm of the output weights of the ELM so that the RELM can resolve the ill-conditioning issue of the hidden layer matrix, where the ELM fails to achieve. Among the regularization methods, ridge regression is the most commonly used in the RELM [3,22–26]. Others including lasso and elastic net [27,28] are also used [29]. Apart from ensuring the invertibility of the hidden layer matrix during the learning procedure, regularization is also an effective way to resolve the overfitting problem by sacrificing the bias so as to reduce the estimation variance, and thus may improve the overall generalization performance [3,25,30]. In spite of many advantages, several open issues remain to be solved for the ELM algorithm. One such issue is the determination of optimal parameters for a specific task. In the RELM, the ridge parameter C plays the most important role as long as the size of the hidden layer L is large enough [3,31]. However, the parameter C in the original RELM algorithm is usually searched

* Corresponding author.
E-mail addresses: zshao1@ntu.edu.sg (Z. Shao), emjer@ntu.edu.sg (M.J. Er).

http://dx.doi.org/10.1016/j.neucom.2016.02.058
0925-2312/

through a trial and error manner. This method is straightforward, but requiring computational and human efforts and cannot guarantee the stability or near optimal performance.

Several ELM variants have been proposed in recent years to circumvent this problem. An extended localized generalization error model is used in [32], where the authors also mentioned that Cross-Validation (CV) is a more reliable, yet time-consuming approach. Another method called risk-estimation-based criterion C_p is used in [33]. However, a meaningful C_p depends on a reliable estimation of the error variance σ^2, which cannot be guaranteed [33]. It is possible that the selected model will be the one where C_p is a particularly severe underestimate of the testing error, therefore C_p cannot completely guard against overfitting.

Optimally Pruned ELM (OP-ELM) uses the Leave-One-Out CV (LOO-CV) error as the selection criterion for a suitable architecture [34], which is nearly unbiased if the training and testing sets are drawn from the same distribution. To address the slow execution speed of the LOO-CV, the Allen's PREdiction Sum of Squares (PRESS) statistics of [30] is utilized in the OP-ELM, which provides a direct and exact formula for the calculation of the LOO-CV error for linear models like the ELM. However, the PRESS formula is used for every newly added hidden node, which involves computing the inverse of a $L\times L$ matrix, and thus can still be computationally intensive. Furthermore, no regularization is implemented in the OP-ELM.

In this paper, a novel Efficient LOO-CV-based RELM (ELOO-RELM) is proposed. Similar to the OP-ELM, the LOO-CV error is used as the architecture's selection criterion, and the PRESS formula is employed to compute it. However, instead of applying the PRESS formula to evaluate every single ridge parameter C, the Singular Value Decomposition (SVD) is used to achieve efficient LOO-CV error calculation (economy size decomposition [35]). To further improve the learning efficiency of the ELOO-RELM, a sparse + instensive search strategy and dynamic termination criteria η are proposed. The parameters in the ELOO-RELM are carefully designed so that the default values can handle many cases (only default parameters are used for the ELOO-RELM in the simulations presented in this paper). Additional tuning guidelines are given so that the user tuned parameters have limited impact on the final generalization performance of the ELOO-RELM.

The rest of this paper is organized as follows: Section 2 introduces the preliminary theory of ELM, RELM, LOO-CV and the PRESS formula. Section 3 describes the calculation procedure of the LOO-CV error and introduces the ELOO-RELM. Performance evaluations on benchmark datasets are carried out in Section 4. Conclusions are drawn in Section 5.

2. Preliminaries

The ELM is a novel paradigm for single hidden layer feedforward neural networks [2]. Its salient feature is that the input weights and hidden biases are randomly chosen instead of being exhaustively tuned, and the output weights are analytically determined using the Moore–Penrose generalized pseudoinverse [3]. The ELM aims to achieve the smallest training error as well as the smallest norm of output weights. Consequently, it has been reported to provide better generalization performance with much faster learning speed, and avoid traditional ANN issues such as specifying learning rate, stopping criterion, number of training epochs, and running into local minima [1,4,5]. In this section, the preliminaries of the ELM, its variant RELM and the LOO-CV are introduced.

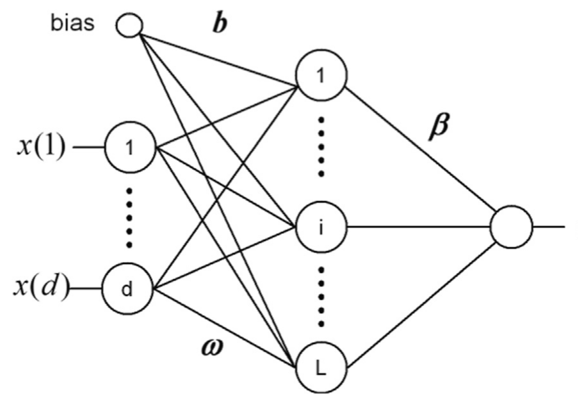

Fig. 1. ELM structure.

2.1. Original ELM

The structure of the original ELM is shown in Fig. 1. For the sake of simplicity, the usual setup of the ELM for regression with a single output is considered in this paper.

The output t with L hidden nodes can be represented by

$$t=\sum_{i=1}^{L}\beta_i g_i(\mathbf{x})=\sum_{i=1}^{L}\beta_i G(\boldsymbol{\omega}_i,b_i,\mathbf{x})=\mathbf{H}\boldsymbol{\beta} \tag{1}$$

where $\mathbf{x},\boldsymbol{\omega}_i\in\mathbb{R}^d$, g_i denotes the ith hidden node output function $G(\boldsymbol{\omega}_i,b_i,\mathbf{x})$ and $\mathbf{H}$ and $\boldsymbol{\beta}$ are the hidden layer output matrix and output weight matrix respectively. For N distinct samples (x_j,t_j), $j=1,\ldots N$, Eq. (1) can be rewritten as

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

$$\mathbf{H}=\begin{bmatrix}\mathbf{h}_1\\ \vdots\\ \mathbf{h}_N\end{bmatrix}=\begin{bmatrix}\mathbf{h}(x_1)\\ \vdots\\ \mathbf{h}(x_N)\end{bmatrix}=\begin{bmatrix}G(\boldsymbol{\omega}_1,b_1,x_1) & \cdots & G(\boldsymbol{\omega}_L,b_L,x_1)\\ \vdots & \cdots & \vdots\\ G(\boldsymbol{\omega}_1,b_1,x_N) & \cdots & G(\boldsymbol{\omega}_L,b_L,x_N)\end{bmatrix}_{N\times L} \tag{3}$$

$$\boldsymbol{\beta}=\begin{bmatrix}\beta_1\\ \vdots\\ \beta_L\end{bmatrix}_{L\times 1};\quad \mathbf{T}=\begin{bmatrix}t_1\\ \vdots\\ t_N\end{bmatrix}_{N\times 1} \tag{4}$$

where $\mathbf{T}$ is the target matrix.

Since the input weights of its hidden neurons $(\boldsymbol{\omega}_i,b_i)$ can be randomly generated instead of being exhaustively tuned [3], the only parameter that needs to be calculated in the ELM is the output weight matrix $\boldsymbol{\beta}$, which can be easily done through the Least Squares Estimate (LSE)

$$\boldsymbol{\beta}=\mathbf{H}^{\dagger}T \tag{5}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of matrix $\mathbf{H}$, which can be calculated through orthogonal projection, i.e., $\mathbf{H}^{\dagger}=(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$.

The procedure of designing an ELM is as follows:

1. Randomly generate the hidden neuron parameters $(\boldsymbol{\omega},\mathbf{b})$.
2. Calculate the hidden-layer output matrix $\mathbf{H}$ using Eq. (3).
3. Calculate the output weights $\boldsymbol{\beta}$ using Eq. (5).

2.2. Regularized ELM

Although several regularization methods have been used in the RELM, the ridge regression of [36] is employed in the proposed algorithm. It is one of the most common regularization methods used in the RELM as well as in the regularized regression problem. It also has a relatively simple format and therefore can be