Applied Soft Computing 21 (2014) 677–684

Contents lists available at ScienceDirect

Applied Soft Computing

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

Least learning machine and its experimental studies on regression capability

Shitong Wang[a,b,*], Fu-Lai Chung[b], Jun Wu[a], Jun Wang[a]

[a] School of Digital Media, Jiangnan University, Wuxi, Jiangsu, China
[b] Department of Computing, Hong Kong Polytechnic University, Hong Kong, China

ARTICLE INFO

Article history:
Received 23 March 2011
Received in revised form 29 October 2013
Accepted 1 April 2014
Available online 24 April 2014

Keywords:
Feedforward neural network
Extreme learning machine
Hidden-feature-space ridge regression
Least learning machine

ABSTRACT

Feedforward neural networks have been extensively used to approximate complex nonlinear mappings directly from the input samples. However, their traditional learning algorithms are usually much slower than required. In this work, two hidden-feature-space ridge regression methods HFSR and centered-ELM are first proposed for feedforward networks. As the special kernel methods, the important characteristics of both HFSR and centered-ELM are that rigorous Mercer's condition for kernel functions is not required and that they can inherently be used to propagate the prominent advantages of ELM into MLFN. Except for randomly assigned weights adopted in both ELM and HFSR, HFSR also exploits another randomness, i.e., randomly selected examplars from the training set for kernel activation functions. Through forward layer-by-layer data transformation, we can extend HFSR and Centered-ELM to MLFN. Accordingly, as the unified framework for HFSR and Centered-ELM, the least learning machine (LLM) is proposed for both SLFN and MLFN with a single or multiple outputs. LLM actually gives a new learning method for MLFN with keeping the same virtues of ELM only for SLFN, i.e., only the parameters in the last hidden layer require being adjusted, all the parameters in other hidden layers can be randomly assigned, and LLM is also much faster than BP for MLFN in training the sample sets. The experimental results clearly indicate the power of LLM on its application in nonlinear regression modeling.

1. Introduction

In recent years, feedforward neural networks have been extensively used for nonlinear regression modeling. Its widespread popularity in regression modeling is mainly due to their strong ability to approximate complex nonlinear mappings directly from the input samples. From a mathematical viewpoint, existing works about the approximation capability of feedforward neural networks can be categorized into two types: universal approximation on compact input sets and approximation of a finite set of samples. Theoretical results about the universal approximation of feedforward neural networks have been obtained by Hornik and Lesino, see [28,29]. In real-world applications, since feedforward neural networks are trained on a finite set of samples, much more endeavors should be taken for the approximation capability of the second type. Typically, gradient descent based learning algorithms like BP [1–5] of feedforward neural networks have been developed and extensively applied in the last decades. When these learning algorithms are used, all the parameters of feedforward neural network need to be adjusted in a backward way and thus there exists the dependence relationship between different layers of parameters in the network. Due to iterative learning steps, these learning algorithms generally converge very slowly and even to local minima. On the other hand, cross-validation and/or early stopping are sometimes adopted to circumvent the over-fitting phenomena.

In order to overcome these shortcomings of the existing learning algorithms, Huang et al. demonstrated and proved that the traditional iterative techniques are not required in adjusting parameters of SLFNs at all. Based on the universal approximation capability of SLFNs with random hidden nodes, Huang et al. proposed a simple and efficient learning method referred to as extreme learning machine (ELM) [1–14,17–21]. They proved that the input weights and the hidden layer biases can be randomly assigned if the activation function in the hidden layer is infinitely differentiable. Once the input weights and the hidden layer biases are randomly assigned, SLFN can be considered as a linear system and the output weights of SLFN can be analytically solved by using the simple generalized inverse operation of the hidden layer output matrix. With its easy implementation, ELM can tend to reach both the smallest training error and the smallest norm of weights and thus provide good

* Corresponding author. Tel.: +86 13182791468.
E-mail address: wxwangst@aliyun.com (S. Wang).

http://dx.doi.org/10.1016/j.asoc.2014.04.001
1568-4946/

generalization performance at extremely fast learning speed, for example, thousands of times faster than BP in many applications [5].

Up to now, many variants of ELM have been developed. Huang et al. [2,10,11,33] gave an intensive survey on ELM and its variants, especially on batch learning mode of ELM [4,5,32], fully complex ELM [12], online sequential ELM [3], incremental ELM [2,10,11], and ensemble of ELM [8,31]. As stated by Huang [5], however, ELM at its present form can only be applied to SLFNs. For many real-world applications, a multiple hidden layer feedforward neural network is more suitable for nonlinear regression modeling since it can approximate large number of samples with less hidden nodes than SLFN can [34].

Although extreme learning machine is able to learn thousands of times faster than conventional popular learning algorithms for SLFNs, developing a fast learning method for MLFNs is still an open problem. In this paper, this problem is well investigated by building the link between extreme learning machine (ELM) and the variants of ridge regression. Two variants of ridge regression, i.e., the hidden-feature-space ridge regression HFSR and centered ridge regression Centered-ELM, for both SLFN and MLFN are first proposed. As the special kernel methods, the virtues of both HFSR and Centered-ELM exist in that rigorous Mercer's condition for kernel functions is not required and that it plays a bridging role in naturally propagating the prominent advantages of ELM into MLFN by using randomly assigned parameters and randomly selected samples for kernel activation functions. Through constructing the transformed data set from the training dataset in a forward layer-by-layer way, we can easily extend HFSR and Centered-ELM to MLFN. Accordingly, as the unified framework for HFSR and Centered-ELM, the least learning machine (LLM) is proposed for both SLFN and MLFN with a single or multiple outputs. LLM keeps the same virtues of ELM only for SLFN, i.e., only the parameters in the last hidden layer require being adjusted, all the parameters in other hidden layers can be randomly assigned, and LLM is much faster than BP in training the sample sets. The experimental results on regression datasets clearly indicate the power of LLM on nonlinear regression modeling.

It should be worth pointing out that the objective of this paper does not pursue for the performance advantage of LLM over ELM. Our contribution exists in two aspects: (1) Through LLM, we can extend ELM to MLFN with keeping the same virtues of ELM only for SLFN; (2) LLM indeed gives a new forward encoding learning way rather than a backward gradient-descent learning way in the widely used learning algorithm BP. It views the behavior of MLFN between the last hidden layer and the input layer as the successive encoding procedure for the input data in a difficult-to-understand way. To large extent, this new understanding can also help us answer why MLFN behaves like a black box.

The remainder of this paper is organized as follows. In Section 2, we briefly review ELM for SLFN. In Section 3, we first propose the hidden-feature-space ridge regression HFSR and Centered-ELM, and then build the link between ELM and them for SLFN. Finally, we give the least learning machine LLM as the unified framework of HFSR and Centered-ELM for SLFN and MLFN with a single or multiple outputs. In Section 4, we report the obtained experimental results about Centered-ELM for SLFN and LLM for MLFN on artificial or benchmarking datasets. Section 5 concludes the paper.

2. ELM for SLFN

In this section, we give a brief review of the extreme learning machine for a single hidden layer feedforward neural network. For easy interpretation and derivation hereafter and without loss of generality, we first consider a single hidden layer feedforward neural network (SLFN for brevity) with a single output here. Given N arbitrary distinct samples $(\boldsymbol{x}_j, t_j)$, $\boldsymbol{x}_j=[x_{j1}, x_{j2}, ..., x_{jn}]^T \in \boldsymbol{R}^n$, $t_j \in \boldsymbol{R}$, $j = 1, 2, ..., N$, SLFN with $\tilde{N}$ hidden nodes and the activation function $g(\boldsymbol{x})$ and a single output can be mathematically modeled as

$$\sum_{i=1}^{\tilde{N}} \beta_i g_i(\boldsymbol{x}_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(\boldsymbol{w}_i^T \boldsymbol{x}_j + b_i) = O_j \quad j = 1, 2, \ldots, N \tag{1}$$

where $\boldsymbol{w}_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$ is the weight vector connecting the ith hidden node and the input nodes, $\beta = [\beta_1, \beta_2, \ldots, \beta_{\tilde{N}}]^T$ is the weight vector connecting all the hidden nodes and the output node, b_i is the threshold of the ith hidden node, and $\boldsymbol{w}_i^T \boldsymbol{x}_j$ denotes the inner product of $\boldsymbol{w}_i$ and $\boldsymbol{x}_j$.

We desire that the above SLFN with a single output can approximate these N samples with zero error, that is to say,

$$\sum_{j=1}^{N} ||O_j - t_j||^2 = 0, \tag{2}$$

$$\text{i.e.} \quad \sum_{i=1}^{\tilde{N}} \beta_i g(\boldsymbol{w}_i^T \boldsymbol{x}_j + b_i) = t_j, \quad j = 1, 2, \ldots, N$$

The above N equations can be compactly written as the following linear system

$$\boldsymbol{H}\boldsymbol{\beta} = \boldsymbol{T} \tag{3}$$

where $\boldsymbol{H}(\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{\tilde{N}}, b_1, b_2, \ldots, b_{\tilde{N}}, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N)$

$$= \begin{bmatrix} g(\boldsymbol{w}_1^T \boldsymbol{x}_1 + b_1) & \ldots & g(\boldsymbol{w}_{\tilde{N}}^T \boldsymbol{x}_1 + b_{\tilde{N}}) \\ \vdots & \ldots & \vdots \\ g(\boldsymbol{w}_1^T \boldsymbol{x}_N + b_1) & \ldots & g(\boldsymbol{w}_{\tilde{N}}^T \boldsymbol{x}_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}} \tag{4}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \ldots \\ \beta_{\tilde{N}} \end{bmatrix}_{\tilde{N} \times 1} \qquad \boldsymbol{T} = \begin{bmatrix} t_1 \\ t_2 \\ \ldots \\ t_N \end{bmatrix}_{N \times 1} \tag{5}$$

where $\boldsymbol{H}$ is called the hidden layer output matrix of SLFN, whose ith column is the ith hidden node output with respect to the inputs $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$.

According to Theorem 2.1 and Theorem 2.2 in [5], for the linear system in Eq. (4), its unique solution, i.e., the smallest norm least squares solution $\hat{\boldsymbol{\beta}}$ can be computed as follows:

$$\hat{\boldsymbol{\beta}} = \boldsymbol{H}^{\dagger}\boldsymbol{T} \tag{6}$$

where $\boldsymbol{H}^{\dagger}$ is the Moore-penrose generalized inverse of the matrix $\boldsymbol{H}$. Accordingly, Huang et al. proposed the following extreme learning machine ELM [5].

Extreme learning machine ELM.

Given the sample set $D = \{(\boldsymbol{x}_j, t_j) | \boldsymbol{x}_j \in \boldsymbol{R}^n, t_j \in \boldsymbol{R}, j = 1, 2, \ldots, N\}$, the infinitely differential activation function $g(\boldsymbol{x})$ and the hidden node number $\tilde{N}$ of SLFN with a single output.

Step1: Randomly assign the weight vector and the bias $\boldsymbol{w}_i, b_i, i = 1, 2, \ldots \tilde{N}$

Step2: Compute the hidden layer output matrix $\boldsymbol{H}$

Step3: Compute the output weight vector of SLFN, i.e., $\hat{\boldsymbol{\beta}} = \boldsymbol{H}^{\dagger}\boldsymbol{T}$, where $\boldsymbol{T} = [t_1, t_2, \ldots t_N]^T$.