Neurocomputing 174 (2016) 1147–1155

Contents lists available at ScienceDirect

Neurocomputing

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

Cholesky factorization based online regularized and kernelized extreme learning machines with forgetting mechanism

Xin-Ran Zhou, Chun-Sheng Wang *

School of Information Science and Engineering, Central South University, Changsha, China

ARTICLE INFO

Article history:
Received 25 June 2015
Received in revised form
21 September 2015
Accepted 11 October 2015
Communicated by G.-B. Huang
Available online 23 October 2015

Keywords:
Extreme learning machine
Online sequential learning algorithms
Forgetting mechanism
Cholesky decomposition

ABSTRACT

In this paper, we propose two alternative schemes of fast online sequential extreme learning machine (ELM) for training the single hidden-layer feedforward neural networks (SLFN), termed as Cholesky factorization based online regularized ELM with forgetting mechanism (CF-FORELM) and Cholesky factorization based online kernelized ELM with forgetting mechanism (CF-FOKELM). First, the solutions of regularized ELM (RELM) and kernelized ELM (KELM) using the matrix Cholesky factorization are introduced; then the recursive method for calculating Cholesky factor of involved matrix in RELM and KELM is designed when RELM and KELM are applied to train SLFN online; consequently, the CF-FORELM and CF-FOKELM are obtained. The numerical simulation results show CF-FORELM demands less computational burden than Dynamic Regression ELM (DR-ELM), and CF-FOKELM also owns higher computational efficiency than both FOKELM and online sequential ELM with kernels (OS-ELMK), and CF-FORELM is less sensitive to model parameters than CF-FOKELM.

1. Introduction

Lots of research work has shown that the single hidden-layer feedforward neural networks (SLFN) can approximate any function and form decision boundaries with arbitrary shapes if the activation function is chosen properly [1–3]. To train SLFN fast, Huang et al. proposed a learning algorithm called extreme learning machine (ELM), which randomly assigns the hidden nodes parameters (the input weights and hidden layer biases of additive networks or the centers and impact factors of RBF networks) and then determines the output weights by the Moore–Penrose generalized inverse [4,5]. The original ELM is batch learning algorithm. Over the past decade, owing to its high efficiency, ELM has gained increasing interest from various research fields and its variants have been designed for different purposes under general or specific conditions [6–10].

In order to train SLFN to achieve better generalization ability, Regularized ELM (RELM) [11–13], which is equivalent to the constrained optimization based ELM [14,15] mathmatically, was presented by researchers. Besides, calculating inverse of invertible matrix in RELM is less complex than calculating Moore–Penrose generalized inverse in ELM; the RELM can greatly reduce the randomness effect in ELM [16].

In order to train SLFN to capture late characteristics of identified systems, especially nonstationary time series or time-varying systems where training data have timeliness, Zhang and Wang, using the block matrix inverse formula [17], proposed Dynamic Regression ELM (DR-ELM) [18], that is, Local ELM (LELM) [19]. Recently, Zhou et al. presented online RELM with forgetting mechanism (FORELM) [20] invoking Sherman–Morrison formula [17]. Both DR-ELM and FORELM study a given number of samples, i.e., those samples during the sliding time window.

If the feature mapping in SLFN is unknown, the kernelized ELM (KELM) can be constructed [14,15]. In order to train SLFN to identify time-varying or nonstationary systems, Zhou et al. presented online kernelized ELM with forgetting mechanism (FOKELM) [20], and Wang and Han proposed an online sequential extreme learning machine with kernels (OS-ELMK) [21]. Both FOKELM and OS-ELMK also study samples during the sliding time window only. We do not need to select the number of hidden nodes for these KELMs.

The time efficiency of learning algorithm is an important index in general. In realtime applications, such as stock forecast, modelling for controlled objets, signal processing, the computational efficiency of online training algorithm for SLFN is a crucial factor. Hence, in order to accelerate DR-ELM, FOKELM and OS-ELMK, we design Cholesky factorization based online regularized ELM with forgetting mechanism (CF-FORELM) and Cholesky factorization based online kernelized ELM with forgetting mechanism (CF-FOKELM). The numerical experiments show CF-FORELM runs faster than DR-ELM, and CF-FOKELM also owns higher computational

* Corresponding author.
E-mail address: wangcsh2015@126.com (C.-S. Wang).

http://dx.doi.org/10.1016/j.neucom.2015.10.033
0925-2312/

efficiency than both FOKELM and OS-ELMK. Furthermore, we compare parameters sensitivity between CF-FORELM with the sigmoidal additive function and CF-FOKELM with Gaussian kernel function, contrast their speed, and offer suggestions to choose CF-FORELM or CF-FOKELM.

It should be noticed that herein CF-FORELM, like FORELM, tunes the output weights of SLFN due to addition and deletion of the samples one by one, viz., learns and forgets samples sequentially, and network architecture is fixed. It is completely different from those offline incremental RELM [22–24] seeking optimal network architecture by adding hidden nodes one by one, and learning the data in batch mode.

The rest of this paper is organized as follows. Section 2 gives a brief review of the basic concepts and related work of RELM and KELM. Section 3 introduces solutions of RELM and KELM by using the Cholesky factorization. Section 4 proposes two new online learning algorithms, viz., CF-FORELM and CF-FOKELM. Performance evaluation is conducted in Section 5. Finally, conclusions and discussion are given in Section 6.

2. Brief review of the RELM and KELM

For simplicity, ELM based learning algorithm for SLFN with multiple input single output is discussed.

The output of a SLFN with L hidden nodes (additive or RBF nodes) can be represented by

$$f(x)=\sum_{i=1}^{L}\beta_i G(\mathbf{a}_i,b_i,\boldsymbol{x})=\boldsymbol{h}(x)\boldsymbol{\beta}\quad \boldsymbol{x}\in \mathrm{R}^n,\quad \mathbf{a}_i\in \mathrm{R}^n, \tag{1}$$

where $\mathbf{a}_i$ and b_i are the learning parameters of hidden nodes, and $\boldsymbol{\beta}=[\beta_1,\beta_2,\ldots,\beta_L]^T$ is the vector of the output weights, and $G(\mathbf{a}_i,b_i,\boldsymbol{x})$ denotes the output of the i-th hidden node with respect to the input $\boldsymbol{x}$, i.e., activation function. $\boldsymbol{h}(x)=[G(\mathbf{a}_i,b_1,\boldsymbol{x}),G(\mathbf{a}_i,b_2,\boldsymbol{x}),\ldots,G(\mathbf{a}_i,b_L,\boldsymbol{x})]$ is a feature mapping from the n-dimensional input space to the L-dimensional hidden-layer feature space. In ELM, $\mathbf{a}_i$ and b_i are randomly determined firstly.

For a given set of distinct training data $\{(\boldsymbol{x}_i,y_i)\}_i^N\subset \mathrm{R}^n\times R$, where $\boldsymbol{x}_i$ is an n dimension input vector and y_i is the corresponding scalar observation, The RELM, i.e., constrained optimization based ELM, can be formulated as

$$\begin{aligned}&\text{Minimize}: \quad L_{P_{ELM}}=\frac{1}{2}\|\boldsymbol{\beta}\|^2+c\frac{1}{2}\|\boldsymbol{\xi}\|^2\\ &\text{Subject to}: \quad \boldsymbol{H\beta}=\boldsymbol{Y}-\boldsymbol{\xi}\end{aligned} \tag{2}$$

where $\boldsymbol{\xi}=[\xi_1,\xi_2,\ldots,\xi_N]^T$ denotes the training error. $\boldsymbol{Y}=[y_1,y_3,\ldots,y_N]^T$ indicates the target value of all the samples. $\boldsymbol{H}_{N\times L}=[\boldsymbol{h}(x_1)^T,\boldsymbol{h}(x_2)^T,\ldots,\boldsymbol{h}(x_N)^T]^T$ is the mapping matrix for the inputs of all the samples. c is the regularization parameter (a positive constant).

Based on the KKT theorem, the constrained optimization of (2) can be transferred to the following dual optimization problem:

$$L_{D_{ELM}}=\frac{1}{2}\|\boldsymbol{\beta}\|^2+c\frac{1}{2}\|\boldsymbol{\xi}\|^2-\boldsymbol{\alpha}^T(\boldsymbol{H\beta}-\boldsymbol{Y}+\boldsymbol{\xi}) \tag{3}$$

where $\boldsymbol{\alpha}=[\alpha_1,\alpha_2,\ldots,\alpha_N]^T$ is the Lagrange multipliers vector. Using KKT optimality conditions, the following equations can be gotten:

$$\frac{\partial L_{D_{ELM}}}{\partial\boldsymbol{\beta}}=\mathbf{0}\rightarrow\boldsymbol{\beta}=\boldsymbol{H}^T\boldsymbol{\alpha} \tag{4}$$

$$\frac{\partial L_{D_{ELM}}}{\partial\boldsymbol{\xi}}=\mathbf{0}\rightarrow\boldsymbol{\alpha}=c\boldsymbol{\xi} \tag{5}$$

$$\frac{\partial L_{D_{ELM}}}{\partial\boldsymbol{\alpha}}=\mathbf{0}\rightarrow\boldsymbol{H\beta}-\boldsymbol{Y}+\boldsymbol{\xi}=\mathbf{0} \tag{6}$$

Ultimately, $\boldsymbol{\beta}$ can be obtained as follows [11,14,15]:

$$\boldsymbol{\beta}=(c^{-1}\mathbf{I}+\boldsymbol{H}^T\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{Y} \tag{7}$$

or

$$\boldsymbol{\beta}=\boldsymbol{H}^T(c^{-1}\mathbf{I}+\boldsymbol{H}\boldsymbol{H}^T)^{-1}\boldsymbol{Y} \tag{8}$$

In order to reduce computational costs, when $N>L$, one may prefer to apply solution (7), and when $N<L$, one may prefer to apply solution (8).

Invoking Eq. (8), the output of SLFN can be obtained as

$$\begin{aligned}f(x)&=\boldsymbol{h}(x)\boldsymbol{\beta}=\boldsymbol{h}(x)\boldsymbol{H}^T(c^{-1}\mathbf{I}+\boldsymbol{H}\boldsymbol{H}^T)^{-1}\boldsymbol{Y}=\boldsymbol{h}(x)[\boldsymbol{h}(\boldsymbol{x}_1)^T,\boldsymbol{h}(\boldsymbol{x}_2)^T,\ldots,\boldsymbol{h}(\boldsymbol{x}_N)^T]\\ &\quad(c^{-1}\mathbf{I}+\boldsymbol{H}\boldsymbol{H}^T)^{-1}\boldsymbol{Y}\\ &=[\boldsymbol{h}(\boldsymbol{x})\boldsymbol{h}(\boldsymbol{x}_1)^T,\boldsymbol{h}(\boldsymbol{x})\boldsymbol{h}(\boldsymbol{x}_2)^T,\ldots,\boldsymbol{h}(\boldsymbol{x})\boldsymbol{h}(\boldsymbol{x}_N)^T]\boldsymbol{g}\\ &=\sum_{i=1}^{N}g_i\boldsymbol{h}(\boldsymbol{x})\boldsymbol{h}(\boldsymbol{x}_i)^T\end{aligned} \tag{9}$$

where $\boldsymbol{g}=[g_1,g_2,\ldots,g_N]^T$. Obviously, $\boldsymbol{g}$ can be calculated by solving the following equation:

$$(c^{-1}\mathbf{I}+\boldsymbol{H}\boldsymbol{H}^T)\boldsymbol{g}=\boldsymbol{Y} \tag{10}$$

If the feature mapping $\boldsymbol{h}(\boldsymbol{x})$ is unknown, one can apply Mercer's conditions on RELM. The kernel matrix is defined as $\boldsymbol{\Omega}=\boldsymbol{H}\boldsymbol{H}^T$ and $\boldsymbol{\Omega}_{ij}=\boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{h}(\boldsymbol{x}_j)^T=K(\boldsymbol{x}_i,\boldsymbol{x}_j)$. Then, the output of SLFN by kernel based RELM can be given as

$$\begin{aligned}f(x)&=\boldsymbol{h}(x)\boldsymbol{\beta}=\boldsymbol{h}(x)\boldsymbol{H}^T(c^{-1}\mathbf{I}+\boldsymbol{H}\boldsymbol{H}^T)^{-1}\boldsymbol{Y}\\ &=[K(\boldsymbol{x},\boldsymbol{x}_1),K(\boldsymbol{x},\boldsymbol{x}_2),\ldots,K(\boldsymbol{x},\boldsymbol{x}_N)](c^{-1}\mathbf{I}+\boldsymbol{\Omega})^{-1}\boldsymbol{Y}\\ &=[K(\boldsymbol{x},\boldsymbol{x}_1),K(\boldsymbol{x},\boldsymbol{x}_2),\ldots,K(\boldsymbol{x},\boldsymbol{x}_N)]\boldsymbol{g}\\ &=\sum_{i=1}^{N}g_iK(\boldsymbol{x},\boldsymbol{x}_i)\end{aligned} \tag{11}$$

here $\boldsymbol{g}$ can be calculated by solving the following equation:

$$(c^{-1}\mathbf{I}+\boldsymbol{\Omega})\boldsymbol{g}=\boldsymbol{Y} \tag{12}$$

3. The solutions of RELM and KELM by the Cholesky factorization

Theorem 1. *The matrix $c^{-1}\mathbf{I}+\boldsymbol{H}\boldsymbol{H}^T$ is a symmetrical positive definition matrix.*

Proof. Symmetry. Apparenty,

$$(c^{-1}\mathbf{I}+\boldsymbol{H}\boldsymbol{H}^T)^T=c^{-1}\mathbf{I}+(\boldsymbol{H}\boldsymbol{H}^T)^T=c^{-1}\mathbf{I}+(\boldsymbol{H}\boldsymbol{H}^T) \tag{13}$$

Positive definition property. For any $\boldsymbol{\zeta}=[\zeta_1,\zeta_2,\ldots,\zeta_N]\neq\mathbf{0}$, it holds that

$$\boldsymbol{\zeta}c^{-1}\mathbf{I}\boldsymbol{\zeta}^T=c^{-1}\sum_{i=1}^{N}\zeta_i^2>0. \tag{14}$$

□

Additionally,

$$\begin{aligned}\boldsymbol{\zeta}\boldsymbol{H}\boldsymbol{H}^T\boldsymbol{\zeta}^T=\boldsymbol{\zeta}\boldsymbol{H}(\boldsymbol{\zeta}\boldsymbol{H})^T&=\left(\sum_{i=1}^{N}\zeta_iG(\mathbf{a}_1,b_1,\boldsymbol{x}_i)\right)^2+\left(\sum_{i=1}^{N}\zeta_iG(\mathbf{a}_2,b_2,\boldsymbol{x}_i)\right)\\ &\quad 2+\cdots+\left(\sum_{i=1}^{N}\zeta_iG(\mathbf{a}_L,b_L,\boldsymbol{x}_i)\right)^2\geq 0\end{aligned} \tag{15}$$

Now that $c^{-1}\mathbf{I}+\boldsymbol{H}\boldsymbol{H}^T$ is symmetric positive definite matrix. Denote $\boldsymbol{B}=c^{-1}\mathbf{I}+\boldsymbol{H}\boldsymbol{H}^T$, then, $\boldsymbol{B}$ can be factorized in Cholesky form, i.e., $\boldsymbol{B}=\boldsymbol{U}^T\boldsymbol{U}$, where $\boldsymbol{U}$ is an upper triangular matrix. Then $\boldsymbol{U}$ can be calculated as following formulas:

$$u_{ii}=\left(b_{ii}-\sum_{d=1}^{i-1}u_{di}^2\right)^{1/2},\quad i=1,\ldots,N \tag{16}$$