Neural Networks 76 (2016) 29–38

Contents lists available at ScienceDirect

Neural Networks

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

# A Fast Reduced Kernel Extreme Learning Machine

Wan-Yu Deng[a,b], Yew-Soon Ong[b,*], Qing-Hua Zheng[c]

[a] School of Computer, Xian University of Posts & Telecommunications, Shaanxi, China
[b] Rolls-Royce@NTU Corporate Lab c/o, School of Computer Engineering, Nanyang Technological University, Singapore
[c] Department of Computer Science and Technology, Xi'an Jiaotong University, China

ARTICLE INFO

*Article history:*
Received 7 March 2015
Received in revised form 10 July 2015
Accepted 15 October 2015
Available online 6 January 2016

*Keywords:*
Extreme learning machine
Kernel method
Support vector machine
RBF network

ABSTRACT

In this paper, we present a fast and accurate kernel-based supervised algorithm referred to as the Reduced Kernel Extreme Learning Machine (RKELM). In contrast to the work on Support Vector Machine (SVM) or Least Square SVM (LS-SVM), which identifies the support vectors or weight vectors iteratively, the proposed RKELM randomly selects a subset of the available data samples as support vectors (or mapping samples). By avoiding the iterative steps of SVM, significant cost savings in the training process can be readily attained, especially on Big datasets. RKELM is established based on the rigorous proof of universal learning involving reduced kernel-based SLFN. In particular, we prove that RKELM can approximate any nonlinear functions accurately under the condition of support vectors sufficiency. Experimental results on a wide variety of real world small instance size and large instance size applications in the context of binary classification, multi-class problem and regression are then reported to show that RKELM can perform at competitive level of generalized performance as the SVM/LS-SVM at only a fraction of the computational effort incurred.

## 1. Introduction

Kernel based learning methods have been extensively used for solving classification and regression problem due to their high generalization performance and the mathematical rigor of the field (Neuvial, 2013). To date, a plethora of kernel based learning methods like the support vector machine (SVM) (Vapnik, 1995) and its variants including LS-SVM (Suykens & Vandewalle, 1999), RSVM (Lee & Huang, 2007) and LLSVM (Zhang, Lan, Wang, & Moerchen, 2012) have been proposed for data analysis. The classical SVM involves a mapping of the training data into a high dimensional feature space through some nonlinear feature mapping function. A standard optimization method then follows so as to arrive at appropriate solution that maximizes the margin of separation between the different classes in the nonlinear feature space, while minimizing training errors.

A least square version of the SVM classifier was subsequently proposed in Suykens and Vandewalle (1999). In contrast to the inequality constraints adopted in a classical SVM, equality constraints are considered in the LS-SVM (Suykens & Vandewalle, 1999). Instead of quadratic programming, one can thus implement the least square approach with ease by solving a set of linear equations. Notably, LS-SVM has been reported to exhibit superior generalization performance and low computational requirements in many applications. Recently, a unified learning framework for regression and classification, which is termed as the Kernel Extreme Learning Machine (KELM) was also proposed as an extension of the Extreme Learning Machine (ELM) learning theory and the classification capability theorem. In the KELM unified learning framework, the bias term in the optimization constraints of SVM, LS-SVM, and PSVM can be eliminated. This gives rise to learning algorithm with mild optimization constraints that offers improved generalization performance and low computational complexity. Nonetheless, when dealing with large scale problems, the conventional SVM and LS-SVM as well as the KELM do not scale well with the big datasets in general.

In the past decades, many large scale learning algorithms have been proposed. Lee and Huang (2007) proposed the reduced support vector machines (RSVM) which restrict the number of candidate support vectors. The main characteristic of this method is to reduce the kernel matrix from $N \times N$ to $N \times \tilde{N}$, where $N$ is the number of training instances and $\tilde{N}$ denotes the size of a randomly selected subset of training data that serve as candidate support vectors. Wang, Crammer, and Vucetic (2012) proposed a budgeted stochastic gradient descent approach for training SVMs

* Corresponding author.
*E-mail address:* ASYSOng@ntu.edu.sg (Y.-S. Ong).

http://dx.doi.org/10.1016/j.neunet.2015.10.006
0893-6080/© 2015 Elsevier Ltd. All rights reserved.

(BSGD-SVM). The approach bounds the number of support vectors during training through several budget maintenance strategies including removal, projection and merging. Chang, Guo, Lin, and Lu (2010) proposed an approach that employs a decision tree to decompose the given data space as a first step before training SVMs on the decomposed regions. Results reported indicated notable speed up in training time at competitive test accuracy. LASVM (Bordes, Ertekin, Weston, & Bottou, 2005), on the other hand, is a one-pass online SVM that involves iterations of sequential minimal optimization (SMO) during each model update so as to remove data samples that are deemed as unlikely to serve as suitable SVs from the training set. Tsang, Kwok, and Cheung (2005) scale up the kernel SVM by reformulating the quadratic programming used in SVM as a minimum enclosing ball problem and then use an efficient approximation algorithm to attain a near-optimal solution. In spite of the extensive efforts in the area to cope with the increasing data instance size and dimensionality more elegantly, existing works have mainly focused on developing effective strategies for ***identifying the optimal set of support vectors***. The computational process of identifying support vectors, however can become very intensive, especially when dealing with large scale data.

In what follows, the core objectives and contributions of the present work are outlined: (1) we propose a fast non-iterative kernel machine, which is referred to as the fast Reduced Kernel Extreme Learning Machine (RKELM), based on kernel-based learning and extreme learning machine. A key characteristics of the present work is that ***support vectors are randomly chosen from the training set*** as opposed to some sophisticated process which is often compute intensive. (2) We prove that RKELM can approximate any nonlinear functions with zero error only if the kernel is strictly positive definite and all training data are chosen as support vectors. (3) We analyze the relation of the present work to other related works, such as KELM, ELM and RSVM, and reveal the effects of hidden nodes size and the regularization parameter on generalization performances.

The rest of the paper is organized as follows: Section 2 gives a brief overview of the classic ELM (Huang, Zhu, & Siew, 2006) and the KELM (Huang et al., 2006). Section 3 presents the mathematical derivation of RKELM. The relation of RKELM to other relevant state-of-the-art algorithms is then discussed in Section 4. The performance evaluation and validation of RKELM is subsequently presented in Section 5, using commonly used well established datasets from the UCI repository. Last but not least, the brief conclusions and future works are given in Section 6.

## 2. A brief review of Extreme Learning Machine and its kernel extension

In this section, an overview of the ELM and its kernel extension (Huang et al., 2006) is presented. This serves to provide the necessary background for the development of the RKELM in Section 3.

### 2.1. ELM

The Extreme Learning Machine (ELM) was proposed as a fast learning method for **S**ingle-hidden-**L**ayer **F**eedforward **N**eural network (SLFN) in Feng, Ong, and Lim (2013); Feng, Ong, Lim, and Tsang (2015), Huang, Chen, and Siew (2006); Huang, Zhu, and Siew (2004) and Rong, Ong, Tan, and Zhu (2008), where the hidden layer can be any form of piecewise continuous computational functions including Sigmoid, Radial Basis, trigonometric, threshold, ridge polynomial, fully complex, fuzzy inference, high-order, wavelet, etc. (Huang & Chen, 2007). In ELM, the number of hidden nodes poses as a structural parameter that needs to be predefined, while parametric settings of the hidden nodes, for example, the impact factors of the RBF nodes, the biases and/or input weights of the additive nodes are randomly assigned.

Given training samples $\{(\mathbf{x}_i, \mathbf{t}_i)|\mathbf{x}_i \in \mathbf{R}^d, \mathbf{t}_i \in \mathbf{R}^m\}_{i=1}^N$, where $N$ is the number of instances, $d$ is the dimension, $m$ is the number of output nodes. For regression problems $m = 1$, while for classification problems $m$ is the number of categories, classes or labels. The output function of ELM for SLFNs is given by

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

where $L$ is the number of hidden nodes, $\boldsymbol{\beta} = [\beta_i, \dots, \beta_L]^T$ is the vector of output weights, $\mathbf{a}_i$ is the center of RBF nodes or input weights of additive nodes, $b_i$ is the impact factor of RBF nodes or bias of additive nodes, and $h(\cdot)$ is the activation function which is but not limited to Sigmoid, Sine and hardlim functions. The ELM can be solved as a constrained optimization problem (Huang, 2014; Huang, Zhou, Ding, & Zhang, 2012; Huang et al., 2006):

$$\text{Minimize}_{\boldsymbol{\beta}} : \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|_p^{\alpha_1} + \frac{C}{2}\|\boldsymbol{\beta}\|_q^{\alpha_2} \tag{2}$$

where $\alpha_1 > 0, \alpha_2 > 0, p, q = 0, \frac{1}{2}, 1, 2, \dots, F, +\infty$ and $C$ is control parameter for a tradeoff between structural risk and empirical risk, $\mathbf{H}$ is the hidden-layer output matrix

$$\mathbf{H} = \begin{bmatrix} h(\mathbf{w}_1, \mathbf{x}_1, b_1) & \cdots & h(\mathbf{w}_L, \mathbf{x}_1, b_L) \\ \vdots & \ddots & \vdots \\ h(\mathbf{w}_1, \mathbf{x}_N, b_1) & \cdots & h(\mathbf{w}_L, \mathbf{x}_N, b_L) \end{bmatrix} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}. \tag{3}$$

A number of efficient methods may be used to determine the output weights $\boldsymbol{\beta}$ such as the orthogonal projection method, iterative methods (Luo, Vong, & Wong, 2014), eigenvalue decomposition method (Golub & VanLoan, 1996) and others. When $p, q = F$ and $\alpha_1, \alpha = 2$, a popular and efficient closed-form solution (Huang et al., 2012) is:

$$\boldsymbol{\beta} = \begin{cases} \mathbf{H}^\top \left(C\mathbf{I} + \mathbf{H}\mathbf{H}^\top\right)^{-1}\mathbf{T} & N \geq L \\ \left(C\mathbf{I} + \mathbf{H}^\top\mathbf{H}\right)^{-1}\mathbf{H}^\top\mathbf{T} & N \leq L. \end{cases} \tag{4}$$

### 2.2. Kernel Extreme Learning Machine

As proposed in Huang et al. (2012), if $\boldsymbol{h}(\cdot)$ is unknown, i.e., an implicit function, one can apply the Mercer's conditions on ELM, and define a kernel matrix for ELM that takes the form:

$$\mathbf{K}_{ELM} = \mathbf{H}\mathbf{H}^T : \mathbf{K}_{ELM\,i,j} = \boldsymbol{h}(\mathbf{x}_i) \cdot \boldsymbol{h}(\mathbf{x}_j) = \kappa(\mathbf{x}_i, \mathbf{x}_j). \tag{5}$$

Then, substituting (5) and (4) into (1), we can obtain the kernel form of the output function as follows,

$$\boldsymbol{f}(\mathbf{x}) = \begin{bmatrix} \kappa(\mathbf{x}, \mathbf{x}_1) \\ \vdots \\ \kappa(\mathbf{x}, \mathbf{x}_N) \end{bmatrix}^T (C\mathbf{I} + \mathbf{K}_{ELM})^{-1}\mathbf{T}. \tag{6}$$

Similar to the SVM (Vapnik, 1995) and LS-SVM (Suykens & Vandewalle, 1999), $\boldsymbol{h}(\mathbf{x})$ need not be known; instead, its kernel $\kappa(\mathbf{u}, \mathbf{v})$ (e.g., Gaussian kernel $\kappa(\mathbf{u}, \mathbf{v}) = \exp(\|\mathbf{u} - \mathbf{v}\|^2/\sigma)$) can be provided. $L$ need not be available beforehand either. The experimental and theoretical analysis of Huang et al. (2012) showed that KELM produces improved generalization performance over the SVM/LS-SVM. The work, however was established only on small datasets. When dealing with Big data, however, the training time of $\mathcal{O}(N^3)$ and kernel matrix size of $\mathcal{O}(N^2)$ become a significant concern (Zhai, Ong, & Tsang, 2014).