Neural Networks 61 (2015) 32–48

Contents lists available at ScienceDirect

Neural Networks

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

Review

Trends in extreme learning machines: A review

Gao Huang[a], Guang-Bin Huang[b,*], Shiji Song[a,*], Keyou You[a]

[a] *Department of Automation, Tsinghua University, Beijing 100084, China*

[b] *School of Electrical and Electronic Engineering, Nanyang Technological University, Nanyang Avenue, 639798, Singapore*

ARTICLE INFO

Article history:
Received 20 March 2014
Received in revised form 25 August 2014
Accepted 2 October 2014
Available online 16 October 2014

Keywords:
Extreme learning machine
Classification
Clustering
Feature learning
Regression

ABSTRACT

Extreme learning machine (ELM) has gained increasing interest from various research fields recently. In this review, we aim to report the current state of the theoretical research and practical advances on this subject. We first give an overview of ELM from the theoretical perspective, including the interpolation theory, universal approximation capability, and generalization ability. Then we focus on the various improvements made to ELM which further improve its stability, sparsity and accuracy under general or specific conditions. Apart from classification and regression, ELM has recently been extended for clustering, feature selection, representational learning and many other learning tasks. These newly emerging algorithms greatly expand the applications of ELM. From implementation aspect, hardware implementation and parallel computation techniques have substantially sped up the training of ELM, making it feasible for big data processing and real-time reasoning. Due to its remarkable efficiency, simplicity, and impressive generalization performance, ELM have been applied in a variety of domains, such as biomedical engineering, computer vision, system identification, and control and robotics. In this review, we try to provide a comprehensive view of these advances in ELM together with its future perspectives.

Contents

[*] Corresponding authors.
E-mail addresses: huang-g09@mails.tsinghua.edu.cn (G. Huang), egbhuang@ntu.edu.sg (G.-B. Huang), shijis@mail.tsinghua.edu.cn (S. Song), youky@mail.tsinghua.edu.cn (K. You).

http://dx.doi.org/10.1016/j.neunet.2014.10.001
0893-6080/© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Feedforward neural networks (FNN) have been well studied and widely used since the introduction of the well-known back-propagation (BP) algorithm (Rumelhart, Hinton, & Williams, 1986). Traditional BP algorithm is essentially a first order gradient method for parameter optimization, which suffers from slow convergence and local minimum problem. Researchers have proposed various ways to improve the efficiency or optimality in training FNN, such as second order optimization methods (Hagan & Menhaj, 1994; Wilamowski & Yu, 2010), subset selection methods (Chen, Cowan, & Grant, 1991; Li, Peng, & Irwin, 2005) or global optimization methods (Branke, 1995; Yao, 1993). Though leading to faster training speed or better generalization performance compared to the BP algorithm, most of these methods still cannot guarantee a global optimal solution.

Recently, extreme learning machine (ELM) has been proposed for training single hidden layer feedforward neural networks (SLFNs). In ELM, the hidden nodes are randomly initiated and then fixed without iteratively tuning. Actually, the hidden nodes in ELM are even not required to be neuron alike. The only free parameters need to be learned are the connections (or weights) between the hidden layer and the output layer. In this way, ELM is formulated as a *linear-in-the-parameter* model which boils down to solving a linear system. Compared to traditional FNN learning methods, ELM is remarkably efficient and tends to reach a global optimum. Theoretical studies have shown that even with randomly generated hidden nodes, ELM maintains the universal approximation capability of SLFNs (Huang & Chen, 2007, 2008; Huang, Chen, & Siew, 2006). With commonly used activation functions, ELM can attain the almost optimal generalization bound of traditional FNN in which all the parameters are learned (Lin, Liu, Fang, & Xu, 2014; Liu, Lin, Fang, & Xu, 2014). The advantages of ELM in efficiency and generalization performance over traditional FNN algorithms have been demonstrated on a wide range of problems from different fields (Huang, Zhou, Ding, & Zhang, 2012; Huang, Zhu, & Siew, 2006). It is worth noting that ELM is generally much more efficient than support vector machines (SVMs) (Cortes & Vapnik, 1995), least square support vector machines (LS-SVMs) (Suykens & Vandewalle, 1999) and other state-of-the-art algorithms. Empirical studies have shown that the generalization ability of ELM is comparable or even better than that of SVMs and its variants (Fernández-Delgado, Cernadas, Barro, Ribeiro, & Neves, 2014; Huang, Song, Gupta, & Wu, 2014; Huang, Zhou, et al., 2012; Huang, Zhu, et al., 2006). Detailed comparisons of ELM and SVM can be found in Huang (2014) and Huang, Zhou, et al. (2012).

During the past decade, theories and applications of ELM have been extensively studied. From learning efficiency point of view, the original design objects of ELM have three-folds: least human invention, high learning accuracy and fast learning speed (as shown in Fig. 1). Various extensions have been made to the original ELM model to make it more efficient and suitable for specific applications. A literature survey on ELM theories and applications was given by Huang, Wang, and Lan (2011). Since then, research on ELM has become even more active. From theoretical aspect, the universal approximation capability of ELM has been further studied in Huang, Zhou, et al. (2012). The generalization ability of ELM has been investigated in the framework of statistical learning theory (Lin et al., 2014; Liu, Gao, & Li, 2012; Liu et al., 2014) and the initial localized generalization error model (LGEM) (Wang, Shao, Miao, & Zhai, 2013). Many variants of ELM have been proposed to meet particular application requirements. For example, in *cost sensitive* learning, the test time should be minimized, which requires a compact network to meet test time budget. To this end, ELM has been successfully adapted to achieve high compactness in network size (Bai, Huang, Wang, Wang, & Westover, 2014; Deng, Li, & Irwin, 2011; Du, Li, Irwin, & Deng, 2013; He, Du, Wang, Zhuang, & Shi, 2011; Lahoz, Lacruz, & Mateo, 2013; Li, Li & Rong, 2013; Martinez-Martinez et al., 2011; Wang, Er, & Han, 2014a; Yang, Wang, & Yuan, 2013, 2012; Yu & Deng, 2012). We also witness the extensions of ELM for online sequential data (Lan, Soh, & Huang, 2009; Liang, Huang, Saratchandran, & Sundararajan, 2006; Rong, Huang, Sundararajan, & Saratchandran, 2009; Ye, Squartini, & Piazza, 2013; Zhao, Wang, & Park, 2012), noisy/missing data (Horata, Chiewchanwattana, & Sunat, 2013; Man, Lee, Wang, Cao, & Miao, 2011; Miche et al., 2010; Yu, Miche, et al., 2013), imbalanced data (Horata et al., 2013; Huang et al., 2014; Zong, Huang, & Chen, 2013), etc. Additionally, apart from being used for traditional classification and regression tasks, ELM has recently been extended for clustering, feature selection and representational learning (Benoit, van Heeswijk, Miche, Verleysen, & Lendasse, 2013; Huang et al., 2014; Kasun, Zhou, & Huang, 2013). In this review, we provide a snapshot assessment of these new developments in the ELM theories and applications.

It is worth noting that the ELM learning frameworks' randomized strategies for nonlinear feature construction have drawn great interests in the computational intelligence and machine learning community (Le, Sarlos, & Smola, 2013; Rahimi & Recht, 2007, 2008a, 2008b; Saxe et al., 2011; Widrow, Greenblatt, Kim, & Park,

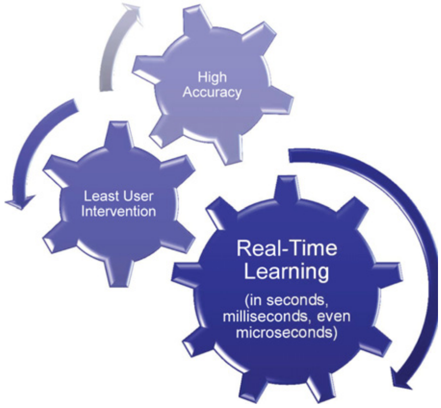

Fig. 1. Learning targets of ELM framework.

2013). These approaches are closely related to ELM and some can be seen as its special cases such that they share many common properties. For example, the random kitchen sinks (RKS) introduced in Rahimi and Recht (2008b) is a special case of ELM that restricts its hidden layer to be constructed of Fourier basis. The No-Prop algorithm in Widrow et al. (2013) has similar spirit of ELM but trains its output weights using the least mean square (LMS) method.

The rest of this review is organized as follows. In Section 2, we introduce the formulations of classical ELM. In Section 3, we review the recent theoretical advances related to ELM. We then introduce the extensions and improvements of ELM for classification and regression, unsupervised learning (e.g., clustering and representational learning) and feature selection in Section 4, Section 5, Section 6, respectively. In Section 7, we review the researches on parallel and hardware implementation issues. The applications of ELM is summarized in Section 8, and Section 9 concludes the paper.

2. Classical ELM

In this section, we introduce the classical ELM model and its basic variants for supervised classification and regression (Huang et al., 2011; Huang, Zhou, et al., 2012; Huang, Zhu, et al., 2006; Huang, Zhu, & Siew, 2004).

2.1. ELM hidden nodes, feature mappings and feature space

ELM was proposed for "generalized" single-hidden layer feedforward networks where the hidden layer need not be neuron alike (Huang & Chen, 2007, 2008; Huang, Zhou, et al., 2012). The output function of ELM for generalized SLFNs is

$$f_L(\mathbf{x}) = \sum_{i=1}^{L} \boldsymbol{\beta}_i h_i(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} \tag{1}$$

where $\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_L]^T$ is the output weight vector between the hidden layer of L nodes to the $m \geq 1$ output nodes, and $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \ldots, h_L(\mathbf{x})]$ is ELM *nonlinear feature mapping* (Fig. 2), e.g., the output (row) vector of the hidden layer with respect to the input $\mathbf{x}$. $h_i(\mathbf{x})$ is the output of the ith hidden node output. The output functions of hidden nodes may not be unique. *Different output functions may be used in different hidden neurons.* In particular, in real applications $h_i(\mathbf{x})$ can be

$$h_i(\mathbf{x}) = G(\mathbf{a}_i, b_i, \mathbf{x}), \quad \mathbf{a}_i \in \mathbf{R}^d, b_i \in R \tag{2}$$

where $G(\mathbf{a}, b, \mathbf{x})$ (with hidden node parameters $(\mathbf{a}, b)$) is a nonlinear piecewise continuous function satisfying ELM universal approximation capability theorems (Huang & Chen, 2007, 2008; Huang, Chen, et al., 2006).

Table 1
Commonly used mapping functions in ELM.

Sigmoid function	$G(\mathbf{a}, b, \mathbf{x}) = \frac{1}{1+\exp(-(\mathbf{a}\cdot\mathbf{x}+b))}$
Hyperbolic tangent function	$G(\mathbf{a}, b, \mathbf{x}) = \frac{1-\exp(-(\mathbf{a}\cdot\mathbf{x}+b))}{1+\exp(-(\mathbf{a}\cdot\mathbf{x}+b))}$
Gaussian function	$G(\mathbf{a}, b, \mathbf{x}) = \exp(-b\Vert\mathbf{x}-\mathbf{a}\Vert)$
Multiquadric function	$G(\mathbf{a}, b, \mathbf{x}) = (\Vert\mathbf{x}-\mathbf{a}\Vert + b^2)^{1/2}$
Hard limit function	$G(\mathbf{a}, b, \mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{a}\cdot\mathbf{x}+b \leq 0 \\ 0, & \text{otherwise} \end{cases}$
Cosine function/Fourier basis	$G(\mathbf{a}, b, \mathbf{x}) = \cos(\mathbf{a}\cdot\mathbf{x}+b)$

Basically, ELM trains an SLFN in two main stages: (1) random feature mapping and (2) linear parameters solving. In the first stage, ELM randomly initializes the hidden layer to map the input data into a feature space (called ELM feature space[1]) by some nonlinear mapping functions (Fig. 3). The random feature mapping stage differs ELM from many existing learning algorithms such as SVM, which uses kernel functions for feature mapping, or deep neural networks (Bengio, 2009), which use Restricted Boltzmann machines (RBM) or Auto-Encoders/Auto-Decoders for feature learning. The nonlinear mapping functions in ELM can be any nonlinear piecewise continuous functions, and Table 1 lists some often used ones.

In ELM, the hidden node parameters $(\mathbf{a}, b)$ are randomly generated (independent of the training data) according to any continuous probability distribution instead of being explicitly trained, leading to remarkable efficiency compared to traditional BP neural networks.

Apart from the activation functions summarized in Table 1, there are other special mapping functions used in ELM and its variants, such as those used in fuzzy ELM (Daliri, 2012; Qu, Shang, Wu, & Shen, 2011; Zhang & Ji, 2013) and wavelet ELM (Avci & Coteli, 2012; Cao, Lin, & Huang, 2010; Malathi, Marimuthu, & Baskar, 2010; Malathi, Marimuthu, Baskar, & Ramar, 2011).

2.2. Basic ELM

In the second stage of ELM learning, the weights connecting the hidden layer and the output layer, denoted by $\boldsymbol{\beta}$, are solved by minimizing the approximation error in the squared error sense:

$$\min_{\boldsymbol{\beta}\in\mathbf{R}^{L\times m}} \Vert\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\Vert^2, \tag{3}$$

where $\mathbf{H}$ is the hidden layer output matrix (*randomized matrix*):

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) & \cdots & h_L(\mathbf{x}_1) \\ \vdots & \vdots & \vdots \\ h_1(\mathbf{x}_N) & \vdots & h_L(\mathbf{x}_N) \end{bmatrix} \tag{4}$$

and $\mathbf{T}$ is the training data target matrix:

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix} = \begin{bmatrix} t_{11} & \cdots & t_{1m} \\ \vdots & \vdots & \vdots \\ t_{N1} & \cdots & t_{Nm} \end{bmatrix} \tag{5}$$

where $\Vert\cdot\Vert$ denotes the Frobenius norm.

[1] Different from the conventional neural network learning tenet which believes that hidden neurons need to be tuned, ELM theories show that learning can be made without iteratively tuning hidden neurons. Although there may have different type of "non-iteratively learning" implementation of hidden layers which need not be randomness based (Deng, Zheng, & Wang, 2013; Huang, Zhou, et al., 2012), this paper focuses on ELM's random feature mappings related works.

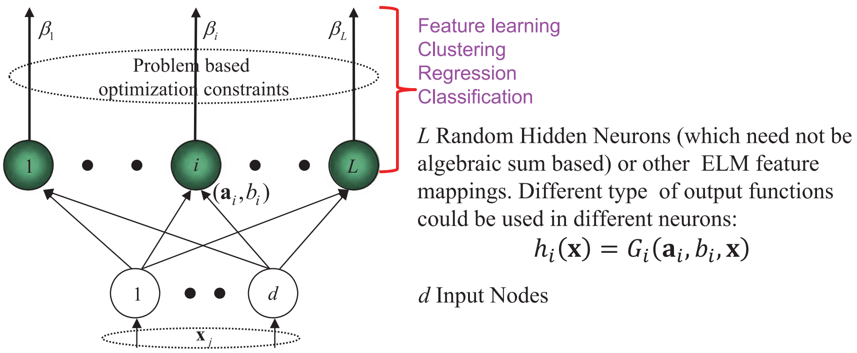
β_1
β_i
β_L
Problem based optimization constraints
Feature learning
Clustering
Regression
Classification
1
i
L
$(\mathbf{a}_i, b_i)$
L Random Hidden Neurons (which need not be algebraic sum based) or other ELM feature mappings. Different type of output functions could be used in different neurons:
$h_i(\mathbf{x}) = G_i(\mathbf{a}_i, b_i, \mathbf{x})$
1
d
d Input Nodes
$\mathbf{x}_j$

Fig. 2. ELM architecture. The hidden nodes in ELM can be combinatorial nodes each consisting of different type of computational nodes (Huang & Chen, 2007).

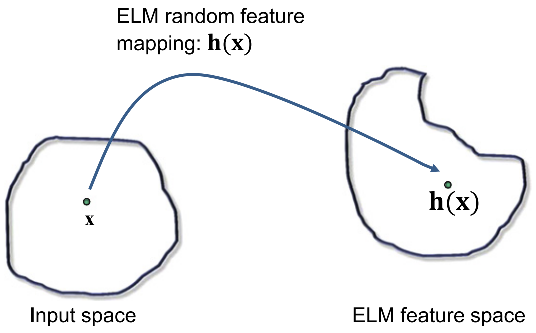

Fig. 3. ELM feature mappings and feature space: (1) Different from the conventional random projection which is a linear mapping, ELM random feature mapping may be formed by different type of nonlinear piecewise continuous neurons: $\mathbf{h}(\mathbf{x}) = [h_1(\mathbf{x}), \ldots, h_L(\mathbf{x})]$ where $h_i(\mathbf{x}) = G_i(\mathbf{a}_i, b_i, \mathbf{x})$ (Huang & Chen, 2007); (2) Different from the conventional neural networks learning tenet that hidden node parameters $(\mathbf{a}_i, b_i)$ need to be adjusted during training, ELM theories (Huang & Chen, 2007, 2008; Huang, Chen, et al., 2006) rigorously prove that hidden node parameters $(\mathbf{a}_i, b_i)$ can be randomly generated according to any continuous probability distribution and the learning capabilities preserved; (3) Different from the feature mapping in SVM which is usually unknown to users, ELM random feature mappings are known to users due to their randomness.

The optimal solution to (3) is given by

$$\boldsymbol{\beta}^* = \mathbf{H}^\dagger \mathbf{T}, \tag{6}$$

where $\mathbf{H}^\dagger$ denotes the Moore–Penrose generalized inverse of matrix $\mathbf{H}$.

There are many efficient methods to solve the above problem, such as Gaussian elimination, orthogonal projection method, iterative method, and single value decomposition (SVD) (Rao & Mitra, 1971).

2.3. Essences of ELM

From the learning point of view, unlike traditional learning algorithms (Block, 1962; Block, Knight, & Rosenblatt, 1962; Rosenblatt, 1958, 1962; Rumelhart et al., 1986; Schmidt, Kraaijveld, & Duin, 1992; White, 1989, 2006), in essence the original aim of ELM is to simultaneously satisfy several salient targets (Huang, Ding, & Zhou, 2010; Huang, Zhou, et al., 2012; Huang, Zhu, et al., 2006; Huang et al., 2004):

(1) Generalization performance: Most algorithms proposed for feedforward neural networks do not consider the generalization performance when they are proposed first time. ELM aims to reach better generalization performance by reaching both the smallest training error and the smallest norm of output weights:

$$\text{Minimize: } \|\boldsymbol{\beta}\|_p^{\sigma_1} + C\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|_q^{\sigma_2} \tag{7}$$

where $\sigma_1 > 0, \sigma_2 > 0, p, q = 0, \frac{1}{2}, 1, 2, \ldots, +\infty$. The first term in the objective function is a regularization term which controls the complexity of the learned model.

(2) Universal approximation capability: Although feedforward neural network architectures themselves satisfy universal approximation capability, most popular learning algorithms designed to train feedforward neural networks do not satisfy the universal approximation capability. In most cases, network architectures and their corresponding learning algorithms are inconsistent in universal approximation capability. However, ELM learning algorithms satisfy universal approximation capability.

(3) Learning without "iteratively tuning" hidden nodes: ELM theories believe that hidden nodes are important and critical to learning, however, hidden nodes need not be tuned and can be independent of training data. Learning can be done without iteratively tuning hidden nodes.

(4) Unified learning theory: There should exist a unified learning algorithm for "generalized" networks with additive/RBF hidden nodes, multiplicative nodes, fuzzy rules, fully complex nodes, hinging functions, high-order nodes, ridge polynomials, wavelets, and Fourier series, etc. (Huang & Chen, 2007).

2.4. Regularized ELM and ELM kernels

Huang, Zhou, et al. (2012) especially studied the stability and generalization performance of ELM with $\sigma_1 = \sigma_2 = p = q = 2$:

$$\begin{aligned} &\min_{\boldsymbol{\beta} \in \mathbf{R}^{L\times m}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\sum_{i=1}^{N}\|\mathbf{e}_i\|^2 \\ &\text{s.t.} \quad \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} = \mathbf{t}_i^T - \mathbf{e}_i^T, \quad i = 1, \ldots, N. \end{aligned} \tag{8}$$

By substituting the constraints of (8) into its objective function, we obtain the following equivalent unconstrained optimization problem:

$$\min_{\boldsymbol{\beta} \in \mathbf{R}^{L\times m}} \quad L_{\mathrm{ELM}} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{2}\|\mathbf{T} - \mathbf{H}\boldsymbol{\beta}\|^2. \tag{9}$$

The above problem is widely known as the ridge regression or regularized least squares. By setting the gradient of L_{ELM} with respect to $\boldsymbol{\beta}$ to zero, we have

$$L_{\mathrm{ELM}} = \boldsymbol{\beta}^* - C\mathbf{H}^T(\mathbf{T} - \mathbf{H}\boldsymbol{\beta}^*) = \mathbf{0}. \tag{10}$$

If $\mathbf{H}$ has more rows than columns ($N > L$), which is usually the case where the number of training patterns is larger than the number of the hidden neurons, we have the following closed form solution for $\boldsymbol{\beta}$:

$$\boldsymbol{\beta}^* = \left(\mathbf{H}^T\mathbf{H} + \frac{\mathbf{I}}{C}\right)^{-1}\mathbf{H}^T\mathbf{T}, \tag{11}$$

where $\mathbf{I}$ is an identity matrix of dimension L.

Note that in practice, rather than explicitly inverting the $L \times L$ matrix in the above expression, we can instead solve a set of linear equations in a more efficient and numerically stable manner.

If the number of training patterns is less than the number of hidden neurons ($N < L$), then $\mathbf{H}$ will have more columns than rows, which usually gives an under-determined least squares problem. Moreover, it is less efficient to invert a $L \times L$ matrix in this case. To handle this problem, we restrict $\boldsymbol{\beta}$ to be a linear combination of the rows in $\mathbf{H}$: $\boldsymbol{\beta} = \mathbf{H}^T\boldsymbol{\alpha}$ ($\boldsymbol{\alpha} \in \mathbf{R}^{N\times m}$). Notice that when $N < L$ and $\mathbf{H}$ is of full row rank, then $\mathbf{HH}^T$ is invertible. Substituting $\boldsymbol{\beta} = \mathbf{H}^T\boldsymbol{\alpha}$ into (10), and multiplying both sides by $(\mathbf{HH}^T)^{-1}\mathbf{H}$, we get

$$\boldsymbol{\alpha}^* - C\left(\mathbf{T} - \mathbf{HH}^T\boldsymbol{\alpha}^*\right) = 0. \tag{12}$$

This yields

$$\boldsymbol{\beta}^* = \mathbf{H}^T\boldsymbol{\alpha}^* = \mathbf{H}^T\left(\mathbf{HH}^T + \frac{\mathbf{I}}{C}\right)^{-1}\mathbf{T}, \tag{13}$$

where $\mathbf{I}$ is an identity matrix of dimension N.

Therefore, in the case where training patterns are plentiful compared to hidden neurons, we use (11) to compute the output weights, otherwise we use (13).

$\mathbf{H}^T\mathbf{H}$ in (11) and $\mathbf{HH}^T$ in (13) are also called "ELM kernel matrix" in which $\mathbf{h}(\mathbf{x}_i) \cdot \mathbf{h}(\mathbf{x}_j)$ is ELM kernel (Frenay & Verleysen, 2011; Huang, Zhou, et al., 2012).

2.5. Learning principles of ELM

Huang (2014) analyzes the three learning principles of ELM.

From learning capability (e.g., universal approximation capability, classification capability) point of view, ELM theories are suitable to almost all types of hidden neurons used in real applications and possibly in biological learning mechanism.

> **Learning Principle I:** (Huang, 2014) Hidden neurons of SLFNs with almost any nonlinear piecewise continuous activation functions or their linear combinations can be randomly generated according to any continuous sampling distribution probability, and such hidden neurons can be independent of training samples and also its learning environment.

ELM learning framework also considers learning stability and generalization performance which have been omitted by most conventional learning algorithms when they were first time proposed.

> **Learning Principle II:** (Huang, 2014) For the sake of system stability and generalization performance, the norm of the output weights of generalized SLFNs need to be smaller with some optimization constraints.

> **Learning Principle III:** (Huang, 2014) *From optimization point of view*, the output nodes of SLFNs should have no biases (or set bias zero).

Learning Principle III of ELM is different from the common understanding among neural network community (e.g., Schmidt et al., 1992; White, 1989, 2006) that the output hidden nodes need biases. From optimization point of view, to have the biases in the output nodes will result in suboptimal solutions. Details of analysis can be referred to Huang (2014).

3. ELM theories

This section reviews the theoretical works related to ELM, including the interpolation theory, universal approximation capability, and generalization bound of ELM.

3.1. Interpolation theory

The learning capability of ELM can be interpreted from the interpolation point of view, as stated by the following two theorems.

Theorem 1 (*Huang, Zhu, et al., 2006*)**.** *Given any small positive value $\epsilon > 0$, any activation function which is infinitely differentiable in any interval, and N arbitrary distinct samples $(\mathbf{x}_i, \mathbf{t}_i) \in \mathbf{R}^d \times \mathbf{R}^m$, there exists $L < N$ such that for any $\{\mathbf{a}_i, \mathbf{b}_i\}_{i=1}^L$ randomly generated from any interval of $\mathbf{R}^d \times R$, according to any continuous probability distribution, with probability one, $\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\| < \epsilon$. Furthermore, if $L = N$, then with probability one, $\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\| = 0$.*

Theorem 1 proves that for any given training set, there exists an ELM network which gives sufficient small training error in squared error sense with probability one, and the number of hidden neurons is no larger than the number of distinct training samples. Actually, if the number of hidden neurons is equal to the number of distinct training samples, then with probability one, the training error decreases to zero. Therefore, from the interpolation perspective, ELM can fit perfectly to any training set provided the number of hidden neurons is large enough.

3.2. Universal approximation capability

The universal approximation capability of SLFNs has been well studied in the past decades (Poggio & Girosi, 1990; White, 1992). However, it is usually assumed that the activation function of the hidden neurons is continuous and differentiable, and the parameters of hidden neurons need to be adjusted during training. In the ELM learning paradigm, the hidden layer parameters are randomly generated instead of being trained. Interestingly, Huang and Chen (2007, 2008), Huang, Chen, et al. (2006) proved that even with randomly generated hidden layer, ELM is still a universal learner.

Theorem 2 (*Huang & Chen, 2007, 2008; Huang, Chen, et al., 2006*)**.** *Given any nonconstant piecewise continuous function $G : \mathbf{R}^d \to R$, if span $\{G(\mathbf{a}, b, \mathbf{x}) : (\mathbf{a}, b) \in \mathbf{R}^d \times R\}$ is dense in L^2, for any continuous target function f and any function sequence $\{G(\mathbf{a}_i, b_i, \mathbf{x})\}_{i=1}^L$ randomly generated according to any continuous sampling distribution, $\lim_{L\to\infty} \|f - f_L\| = 0$ holds with probability one if the output weights $\boldsymbol{\beta}_i$ are determined by ordinary least square to minimize $\|f(\mathbf{x}) - \sum_{i=1}^L \boldsymbol{\beta}_i G(\mathbf{a}_i, b_i, \mathbf{x})\|$.*

From the above theorem, we see that most commonly used activation functions satisfy the conditions that guarantee the universal approximation capability of ELM. Note that Theorem 2 does not require the activation functions to be continuous or differentiable, thus the threshold function and many other activation function can be used in ELM.

Following the property that ELM can approximate any continuous target function, it is not difficult to show that ELM can learn arbitrary decision hyperplane for classification tasks, as stated by the following theorem.

Theorem 3 (*Huang, Zhou, et al., 2012*)**.** *Given any feature mapping $\mathbf{h}(\mathbf{x})$, if $\mathbf{h}(\mathbf{x})$ is dense in $C(\mathbf{R}^d)$ or in $C(\boldsymbol{\Omega})$, where $\boldsymbol{\Omega}$ is a compact set of $\mathbf{R}^d$, then a generalized SLFN with such a random hidden layer mapping $\mathbf{h}(\mathbf{x})$ can separate arbitrary disjoint regions of any shapes in $\mathbf{R}^d$ or $\boldsymbol{\Omega}$.*

Basically, the above theorem states that ELM can approximate any complex decision boundary in classification provided the number of hidden nodes is large enough.

3.3. Generalization bound

Section 3.1 introduces the interpolation capability of ELM on the training data sets, and Section 3.2 investigates the universal capability of ELM. In practice, to achieve better generalization performance on testing set is usually the main pursuit. There are several approaches for studying the generalization ability of a learning machine, e.g., Bayesian framework, statistical learning theory (SLT) and cross validation. Among them, Vapnik–Chervonenkis dimension theory in SLT is one of the most widely used frameworks in generalization bound analysis.

The VC dimension, originally defined by Vladimir Vapnik and Alexey Chervonenkis, is a measure of the capacity of a statistical classification algorithm, defined as the cardinality of the largest set of points that the algorithm can shatter (Vapnik, 2000).

Define the expectation of the test error as

$$R(\boldsymbol{\alpha}) = \int \ell(y, f(\mathbf{x}, \boldsymbol{\theta}))\mathrm{d}p(\mathbf{x}, y), \tag{14}$$

where f is the prediction function, $\boldsymbol{\theta}$ is its parameters, ℓ is a loss function, and $p(\mathbf{x}, y)$ is the cumulative probability distribution that generates training and test samplings.

$R(\boldsymbol{\theta})$ is also known as the *actual risk*, which is difficult to compute in practice. We further define the *empirical risk* $R_{emp}(\boldsymbol{\theta})$ as

$$R_{emp}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} \ell(y_i, f(\mathbf{x}_i, \boldsymbol{\theta})). \tag{15}$$

Then with probability $1-\eta$ $(0 \le \eta \le 1)$, we have the following inequality holds (Vapnik, 2000):

$$R(\boldsymbol{\theta}) \le R_{emp}(\boldsymbol{\theta}) + \sqrt{\frac{\upsilon(\log(2N/\upsilon)+1)-\log(\eta/4)}{N}}, \tag{16}$$

where υ is the VC dimension of a learning machine.

The inequality (16) gives an upper bound ('VC bound') of the actual risk. Minimizing the VC bound is known as the *structure risk minimization* (SRM). From the SRM perspective, to ensure better generalization performance on test set, an algorithm should not only achieve low training error on training set, but also should have a lower VC dimension. Most of the learning algorithms, such as BP neural networks and SVMs, can yield satisfactory performance on training set. However, the VC dimension of these algorithms is difficult to compute or extremely high. For example, the VC dimension for SVM with RBF kernels is infinite.

Recently, Liu, Gao, et al. (2012) showed that the VC dimension of an ELM is equal to its number of hidden neurons with probability one. Thus we are provided a convenient way to compute the generalization bound of ELM.

Theorem 4 (*Liu, Gao, et al., 2012*)**.** *The VC dimension of ELM with L hidden nodes which are infinitely differentiable in any interval is equal to L with probability one.*

From Theorem 1, we know that ELM can achieve low approximation error on training set. Meanwhile, Theorem 4 states that ELM has a relatively low VC dimension. These two theorems together lead to the conclusion that ELM is an ideal classification model under the SRM framework.

The aforementioned work gives an analysis on the generalization performance of ELM on classification tasks. With respect to regression tasks, the generalization ability of ELM has recently been comprehensively studied in Lin et al. (2014) and Liu et al. (2014). It is shown in this two-part papers that with some suitable activation functions, such as polynomials, sigmoid and Nadaraya–Watson functions, ELM can achieve the same optimal generalization bound as a SLFN in which all parameters are tunable. Lin et al. (2014) also find that certain activation functions, such as Gaussian-type functions, may degrade the generalization performance of ELM. However, this problem can be circumvented by training multiple ELM models or introducing regularization techniques (Huang, Zhou, et al., 2012; Lin et al., 2014).

Wang, Shao, et al. (2013) investigated the generalization ability of ELM using the initial localized error model (LGEM). It is shown that the generalization error consists of three parts: training error, stochastic sensitivity measure, and a constant. Based on the derived results, the authors proposed an algorithm for automatic model selection (choosing the number of hidden neurons) method for ELM, which yields satisfactory results on real data sets.

Chen, Peng, Zhou, Li, and Pan (2014) established the generalization bound of an ELM variant used for "learning-to-rank" tasks. In their paper, covering number was introduced to measure the capacity of the hypothesis space.

Rahimi and Recht (2007, 2008a) provided a theoretical analysis on random Fourier features—one of ELM random feature mappings, and show that the inner product of transformed data can uniformly approximate those of some popular kernel functions in the feature space. Later, they gave a generalization bound of testing error for SLFNs with random hidden layer (Rahimi & Recht, 2008b). It is proved that the empirical risk of random SLFNs converges to the lowest true risk attained by a family of functions with a convergence rate of $O(1/\sqrt{N} + 1/\sqrt{L})$.

4. ELM for classification and regression

This section reviews various improvements and extensions of classical ELM for classification and regression problems.

4.1. Improving the stability of ELM

Solving the output weights is a key step in ELM. Essentially, this step is equal to solving a (regularized) least squares problem or ridge regression, where a $N \times N$ or $L \times L$ matrix should be inverted. Mathematically, by setting a proper ridge magnitude (the value of C), one can always ensure positive definiteness of the matrix to be inverted. However, overly increasing the regularization term will sacrifice the prediction accuracy of ELM. One approach for improving the stability of ELM in solving the output weights is to find high quality feature mappings in the first stage. Wang, Cao, and Yuan (2011) proved that for certain activation functions, such as the RBF function, there always exist input weights such that the mapping matrix $\mathbf{H}$ is of full column rank or of full row rank. Then, an efficient input weights selection algorithm is proposed to replace the random feature mapping in ELM, which improves the stability in solving the output weights. Later, this algorithm is extended for ELM with sigmoid activation functions (Chen, Zhu, & Wang, 2013). In Yuan, Wang, and Cao (2011), the output weights are solved in different ways based on the condition of $\mathbf{H}$: column full rank, row full rank, neither column nor row full rank. In this way, the output weights can be computed in a more stable manner than traditional ELM.

4.2. Improving the compactness of ELM

Due to the fact that ELM generates hidden layer randomly, it usually requires more hidden neurons than that of conventional neural networks to achieve matched performance. Large network size results in longer running time in the testing phase of ELM, which may hinder its efficient deployment in some test time sensitive scenarios. Thus, the topic on improving the compactness of ELM has attracted great interest.

One idea is to train ELM in a dynamic way, that is growing, pruning or replacing hidden neurons during the training process. In incremental ELM (I-ELM) (Huang & Chen, 2007, 2008; Huang, Chen, et al., 2006), the newly added hidden neuron(s) can be selected from a candidate pool, and only appropriate neurons are added into the network. Thus, the obtained network can be more compact by getting rid of insignificant neurons. For example, a fast incremental ELM called bidirectional ELM (B-ELM) was proposed to reduce the network size of classical ELM (Yang et al., 2012). In pruning ELM (P-ELM) (Miche et al., 2010; Rong, Ong, Tan, & Zhu, 2008), an initial network is built using traditional ELM, then those hidden neurons with less contribution to the training performance will be removed. In adaptive ELM (A-ELM) (Zhang, Lan, Huang, & Xu, 2012), the size of hidden layer may increase, decrease or stay the same at any step of the training process. The two-stage ELM algorithm proposed by Deng et al. (2011) integrates ELM and leave-one-out (LOO) cross validation with a stepwise construction procedure, which can automatically decide the size of the network and improves the compactness of the model constructed by the ELM. Later, Du et al. (2013) extended the two-stage ELM to a locally regularized framework which also achieves high compactness. The parsimonious ELM proposed by Wang, Er, and Han (2014b) adopts recursive orthogonal least squares to perform forward selection and backward elimination of hidden neurons in ELM, thus leading to a parsimonious structure. Luo, Vong, and Wong (2014) presented a sparse Bayesian approach for learning a compact ELM model. Instead of focusing on explicitly adding or deleting hidden neurons in most existing sparse ELMs, the approach in Luo, et al. (2014) automatically tunes most of the output weights to zeros with an assumed prior distribution, leading to improved sparsity as well as high generalization performance.

Yu and Deng (2012) proposed to reduce the size of ELM network by performing gradient descend on $\boldsymbol{\theta}$, i.e., the parameters of the hidden layer. Note that in Yu and Deng (2012), the activation function is assumed to be continuous and differentiable with respect to the parameters $\boldsymbol{\theta}$. In this algorithm, the output weights $\boldsymbol{\beta}$ are expressed as a function of $\boldsymbol{\theta}$, as shown in (11) or (13). Thus the parameters $\boldsymbol{\theta}$ can be updated in the direction along which the overall square error is reduced the most without solving $\boldsymbol{\beta}$ explicitly. The authors demonstrate by experiments that the algorithm requires only about 1/16 of the model size and thus approximately 1/16 of test time compared with traditional ELM.

Bai et al. (2014) proposed a sparse ELM (S-ELM) by replacing the equality constraints in traditional ELM model by inequality constraints, which greatly reduces the storage space and testing time. Similar to SVMs, the S-ELM with inequality constraints leads to a quadratic programming problem. However, since there is no bias term involved, S-ELM is more efficient in training than SVMs. Furthermore, a fast iterative approach was developed in Bai et al. (2014) for training S-ELM, making it even more efficient than traditional ELM on large data set. In Gastaldo, Zunino, Cambria, and Decherchi (2013), ELM is combined with random projection (RP) to shrink the number of hidden neurons without affecting the generalization performance.

4.3. ELM for online sequential data

Although hidden neurons in ELM can be randomly generated before training data are presented, traditional ELM assumes that all the training data are ready before the training process. However, in certain applications, the training data are received sequentially. To extend ELM for online sequential data, Liang et al. (2006) proposed the online sequential ELM (OS-ELM), which can learn the data one-by-one or chunk-by-chunk with fixed or varying chunk sizes. The OS-ELM is summarized as follows:

Algorithm 1 The OS-ELM algorithm (Liang et al., 2006)

```
Input:
  A training set $\{\mathbf{X}, \mathbf{T}\} = \{\mathbf{x}_i, \mathbf{t}_i\}_{i=1}^N$
Output:
  An trained ELM model.
  Initialization phase:
    - Let $k = 0$. Calculate the hidden layer output matrix $\mathbf{H}_0$ using initial training data, and estimate the initial output weight $\boldsymbol{\beta}_0$ as in classical ELM. Let $\mathbf{P}_0 = (\mathbf{H}_0^T\mathbf{H}_0)^{-1}$.
  Online sequential learning phase:
    - When the $(k+1)$th chunk of new data $\{\mathbf{X}_{k+1}, \mathbf{T}_{k+1}\}$ arrived, update the hidden layer output matrix as $\mathbf{H}_{k+1} = [\mathbf{H}_k^T, \Delta\mathbf{H}_{k+1}^T]^T$, where $\Delta\mathbf{H}_{k+1}$ is the hidden layer output matrix corresponds to the newly arrived data.
    - Update the output weights as $\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \mathbf{P}_{k+1}\mathbf{H}_{k+1}^T(\mathbf{T}_{k+1} - \mathbf{H}_{k+1}\boldsymbol{\beta}_k)$, where $\mathbf{P}_{k+1} = \mathbf{P}_k - \mathbf{P}_k\mathbf{H}_{k+1}^T(\mathbf{I} + \mathbf{H}_{k+1}\mathbf{P}\mathbf{H}_{k+1}^T)^{-1}\mathbf{H}_{k+1}\mathbf{P}_k$
    - Set $k = k + 1$
```

Zhao, Wang, et al. (2012) introduced the forgetting mechanism to OS-ELM (FOS-ELM) to reflect the timeliness of training data in short-term predictions. The algorithm is suitable for practical applications where training data are not only presented one by one or chuck by chuck, but also have the property of timeliness, i.e., training data have a period of validity. For example, in short-term prediction for stock price, the outdated training data with less effectiveness should be down-weighted during training. Rong et al. (2009) proposed an OS-Fuzzy-ELM for the TSK fuzzy model with any bounded nonconstant piecewise continuous membership functions. Ye et al. (2013) extended the OS-ELM to train time-varying data under nonstationary conditions.

4.4. ELM for imbalanced data

In some applications, the training data may be imbalanced. For example, the number of training samples in some classes are much larger than that of other classes. There are also cases where each data point is associated with a unique cost or an importance coefficient. Zong et al. (2013) proposed the weighted ELM (W-ELM) to handle the imbalanced data problem. Different from the classical ELM formulation (8) which treats all training data point equally, W-ELM re-weights them by adding different penalty coefficients to the training errors corresponding to different inputs. Specifically, (8) is reformulated as

$$\begin{aligned} &\min_{\boldsymbol{\beta}\in\mathbf{R}^{L\times m}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}\sum_{i=1}^{N} C_i\|\mathbf{e}_i\|^2 \\ &\text{s.t.} \quad \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} = \mathbf{t}_i^T - \mathbf{e}_i^T, \quad i = 1, \ldots, N, \end{aligned} \tag{17}$$

where C_i is the penalty coefficient corresponding to the ith training point. For imbalance data set, we can set larger value of C_i to points from minority classes and smaller value of C_i to points from majority classes, thus reducing the risk of overfitting to the majority classes.

Substituting the constraints to the objective function yields

$$\min_{\boldsymbol{\beta}\in\mathbf{R}^{L\times m}} \quad L_{\mathrm{ELM}} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}(\mathbf{T} - \mathbf{H}\boldsymbol{\beta})^T\mathbf{C}(\mathbf{T} - \mathbf{H}\boldsymbol{\beta}), \tag{18}$$

where $\mathbf{C}$ is a diagonal matrix whose diagonal elements are $C_1, \ldots, C_N$.

By setting the gradient to zero, we obtain the closed form solution to $\boldsymbol{\beta}$

$$\boldsymbol{\beta} = \begin{cases} \left(\mathbf{H}^T\mathbf{C}\mathbf{H} + \mathbf{I}\right)^{-1} \mathbf{H}^T\mathbf{C}\mathbf{T}, & \text{if } L \leq N \\ \mathbf{H}^T \left(\mathbf{C}\mathbf{H}\mathbf{H}^T + \mathbf{I}\right)^{-1} \mathbf{C}\mathbf{T}, & \text{if } N \geq L. \end{cases} \tag{19}$$

Note that (19) reduces to (11) and (13) respectively if all C_i's are equal. Thus the W-ELM maintains the good performance on well balanced data as classical ELM, and tends to lead to better results on imbalanced data by re-weighting training data to strengthen the impact of minority class while weakening the impact of majority class.

In Horata et al. (2013), an iteratively re-weighted ELM is proposed to handle outliers in the training set. Huang et al. (2014) extended the W-ELM to semi-supervised learning problems.

4.5. *ELM for noisy/missing data*

In Man et al. (2011), the sensitivity of the output weights $\boldsymbol{\beta}$ with respect to the hidden output matrix $\mathbf{H}$ is analyzed. The unregularized ELM is expressed as $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$. Denote by $\Delta\mathbf{H}$ the change in hidden output matrix caused by input data noise or disturbance, and $\Delta\boldsymbol{\beta}$ the corresponding change of the output weights. Then in the case with input data noise, we have

$$(\mathbf{H} + \Delta\mathbf{H})(\boldsymbol{\beta} + \Delta\boldsymbol{\beta}) = \mathbf{T}. \tag{20}$$

Using the fact that $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$, (20) can be expressed as

$$\boldsymbol{\beta} = \mathbf{H}^{\dagger}(\Delta\mathbf{H}(\boldsymbol{\beta} + \Delta\boldsymbol{\beta})), \tag{21}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose inverse of $\mathbf{H}$.

From (21), we have the following inequality:

$$\|\Delta\boldsymbol{\beta}\| \leq \|\mathbf{H}^{\dagger}\|\|\Delta\mathbf{H}\|\|\boldsymbol{\beta} + \Delta\boldsymbol{\beta}\|. \tag{22}$$

The sensitivity of the output weights $\boldsymbol{\beta}$ can be obtained as

$$\frac{\|\Delta\boldsymbol{\beta}\|}{\|\boldsymbol{\beta}\|} \approx \frac{\|\Delta\boldsymbol{\beta}\|}{\|\boldsymbol{\beta} + \Delta\boldsymbol{\beta}\|} \leq \|\mathbf{H}^{\dagger}\|\|\Delta\mathbf{H}\| = \bar{\kappa}(\mathbf{H})\frac{\|\Delta\mathbf{H}\|}{\|\mathbf{H}\|} \tag{23}$$

where $\bar{\kappa}(\mathbf{H}) = \|\mathbf{H}^{\dagger}\|\|\mathbf{H}\|$, which is referred to the *generalized condition number* of the hidden output matrix $\mathbf{H}$.

From (23), it is clear that if the disturbance in $\mathbf{H}$ can be reduced by selecting input weights properly, then the change in output weights will be decreased. In Man et al. (2011), a FIR-ELM is proposed to improve the performance of ELM on noise data, where the input weights are assigned based on the finite impulse response filter (FIR), such that the hidden layer performs as a preprocessor to improve the robustness of the model. In Man, Lee, Wang, Cao, and Khoo (2012), a DFT-ELM is proposed based on discrete Fourier transform (DFT) technique. Similar to FIR-ELM, DFT-ELM also improves the robustness of ELM by designing the input weights to remove the effects of the disturbance from the noisy input data.

Horata et al. (2013) proposed three improved ELM algorithms to solve the outlier robustness problem: (1) ELM based on iteratively reweighted least squares (IRWLS-ELM); (2) ELM based on the multivariate least rimmed squares (MLTS-ELM); and (3) ELM based on the one-step reweighted MLTS (RMLTS-ELM). In IRWLS, the training data is reweighted iteratively via a M-estimate function, e.g., the Huber function, then the weighted ELM (17) is solved at each loop. Therefore, the influence of outliers can be reduced gradually by down-weighing them. In MLTS-ELM and RMLTS-ELM, the multivariate least-trimmed squares estimator (MLTS) based on the minimum covariance determinant (MCD) estimator is introduced to improve the robustness of ELM against outliers.

Yu, Miche, et al. (2013) studied the problem of ELM regression with missing data, and proposed a Tikhonov regularized optimally pruned ELM (TROP-ELM) to handle missing data problem. In TROP-ELM, the missing values are first replaced by their respective conditional means. The authors suggested to use Gaussian function as the active function, whose centers are randomly selected from the inputted data, and the pairwise squared distances are estimated from observed data. When the distance matrix is ready, the output matrix of hidden layer can be computed. In order to improve the compactness of the network, TROP-ELM follows Miche et al. (2010) to rank the hidden neurons using least angle regression (LARS). Then a Tikhonov-regularized version of PRESS is used to select the optimal number of neurons based on leave-one-out (LOO) error.

4.6. *Incremental ELM*

Incremental ELM (I-ELM) is an important variant of ELM. Indeed, the universal approximation capability of ELM is proved from the incremental learning framework (Huang & Chen, 2007, 2008; Huang, Chen, et al., 2006). Compared to other incremental learning algorithms proposed in the literature (Chen et al., 1991; Huang, Song, & Wu, 2012; Kwok & Yeung, 1997; Platt, 1991), I-ELM has the notable feature that it can work with a wide type of activation functions as long as they are nonconstant piecewise continuous functions and are dense in L^2 (Huang & Chen, 2008). The original I-ELM constructs SLFNs incrementally by adding random hidden neurons to the existing network one by one (Huang & Chen, 2007; Huang, Chen, et al., 2006). In the enhanced I-ELM (EI-ELM) algorithm proposed in Huang and Chen (2008), at each step k random hidden neurons are generated and only the one which mostly reduces the training error is selected. It is shown that EI-ELM tends to improve the compactness of the network compared to original I-ELM (Huang & Chen, 2008). Later, Feng, Huang, Lin, and Gay (2009) proposed an efficient incremental ELM called error minimized ELM (EM-ELM) which can add hidden neurons one by one or group by group (with varying group sizes). The EM-ELM is described in Algorithm 2.

By setting a proper training error criteria, the EM-ELM algorithm automatically determines the number of hidden neurons. With the output weights updating strategy introduced in Algorithm 2, EM-ELM is computationally more efficient than classical ELM and other constructive neural networks.

Inspired by the empirical studies, the random strategy in EM-ELM could be further improved by taking into account the underlying distribution of the input data. This issue has been addressed in the *sample selection methods* (Plutowski, Cottrell, & White, 1996), yet it still deserves further investigation when applied to ELM.

4.7. *ELM ensembles*

It is well known that combining a number of learning machines can reduce the risk of overfitting and lead to better generalization performance. Ensemble learning has also been studied in the ELM literature (Cao, Lin, Huang, & Liu, 2012; Lan et al., 2009; Landa-Torres, et al., 2012; Liu & Wang, 2010; Liu, Xu, & Wang, 2009; Tian & Mao, 2010; You, Lei, Zhu, Xia, & Wang, 2013; Zhai, Xu, & Wang, 2012). In Lan et al. (2009), an ensemble of online sequential ELM (EOS-ELM) is proposed which takes the averaged prediction of several independently trained OS-ELM as the final prediction. EOS-ELM further improves the prediction accuracy of OS-ELM. In Liu and Wang (2010), the ensemble learning technique and cross-validation method are embedded into the training phase so as to alleviate the overtraining problem and enhance the predictive stability of ELM. The voting based ELM (V-ELM) proposed by Cao,

Algorithm 2 The EM-ELM algorithm (Feng et al., 2009)

```
Input:
  A training set $\{\mathbf{X}, \mathbf{T}\} = \{\mathbf{x}_i, \mathbf{t}_i\}_{i=1}^N$;
  The maximum number of hidden neurons $N_{\max}$;
  The expected training error $\epsilon$.
Output:
  An trained ELM model.
  Initialization phase:
    - Initialize the SLFN with $N_0$ random hidden neurons
    - Calculate the hidden layer output matrix $\mathbf{H}_0$
    - Calculate the corresponding output error
  Recursively growing phase:
    - Let $k = 0, N_k = N_0$
    - while $N_k < N_{\max}$ and $E(\mathbf{H}_k) > \epsilon$ do
      - Let $k = k + 1$
      - Randomly add $\delta N_k$ hidden neurons to the existing network, and calculate the corresponding output matrix $\delta\mathbf{H}_k$
      - Let $N_k = N_{k-1} + \delta N_k$, and $\mathbf{H}_k = [\mathbf{H}_{k-1}, \delta\mathbf{H}_k]$
      - Update the output weights by $\boldsymbol{\beta}_k = \begin{bmatrix} \mathbf{U}_k \\ \mathbf{D}_k \end{bmatrix} \mathbf{T}$, where $\mathbf{D}_k = \left(\left(\mathbf{I} - \mathbf{H}_{k-1}\mathbf{H}_{k-1}^{\dagger}\right)\delta\mathbf{H}_k\right)^{\dagger}$, and $\mathbf{U}_k = \mathbf{H}_{k-1}^{\dagger}\left(\mathbf{I} - \delta\mathbf{H}_k\mathbf{D}_k\right)$
      - Update the training error $E(\mathbf{H}_k)$
    - end while
```

Lin, Huang, and Liu (2012) can also be seen as an ELM ensemble. We summarize the V-ELM in Algorithm 3.

Algorithm 3 The V-ELM algorithm (Cao, Lin, Huang and Liu (2012))

```
Input:
  A training set $\{\mathbf{X}, \mathbf{T}\} = \{\mathbf{x}_i, \mathbf{t}_i\}_{i=1}^N$, and a test set $\{\mathbf{X}^{\text{test}}, \mathbf{T}^{\text{test}}\} = \{\mathbf{x}_i^{\text{test}}, \mathbf{t}_i^{\text{test}}\}_{i=1}^{N^{\text{test}}}$;
  Number of independent ELMs: $K$;
Output:
  The prediction of the testing set.
  Training phase:
    - Set $k = 1$
    - while $(k \le K)$ do
      - Train the $k$th ELM independently using $\{\mathbf{X}, \mathbf{T}\}$
      - $k = k + 1$
    - end while
  Testing phase:
    - for any testing sample $\mathbf{x}^{\text{test}}$
      - Initialize a zero valued label vector $S_t \in \mathbf{R}^m$
      - Set $k = 1$
      - while $(k \le K)$ do
        - Using the $k$th ELM to predict the label of $\mathbf{x}^{\text{test}}$, say, as $i$ where $i \in [1, 2, \ldots, C]$
        - Let $S_t(i) = S_t(i) + 1$
        - $k = k + 1$
      - end while
      - The final class label of $\mathbf{x}^{\text{test}}$ is given by $\hat{t}^{\text{test}} = \arg\max_{i \in [1,\ldots,C]} \{S_t(i)\}$
    - end for
```

4.8. ELM for ranking

Machine learning based ranking, which is also known as "learning-to-rank", is an active research topic in information retrieval, natural language processing and data mining. In such ranking systems, the ranking problem is usually casted into a supervised learning problem, such as regression or classification, and a prediction model is trained from the data. Recently, ELM is effectively extended to learning-to-rank tasks in Chen et al. (2014) and Zong and Huang (2014).

Zong and Huang (2014) proposed two types of ranking algorithms based on ELM, namely, pointwise RankELM and pairwise RankELM. In pointwise ranking model, each training input $\mathbf{x}_i$ consists of a query and a document, and the training output t_i indicates the relevance degree between them. It is obvious that such training data can be learned with a classification model or regression model, and ELM can be adopted for this task naturally (Zong & Huang, 2014). In pairwise ranking model, the objective is to model the relative relevance of query–document pairs. Roughly, if a query–document pair $\mathbf{x}_i$ has a higher relevance score than another query–document pair $\mathbf{x}_j$, then we should have $t_i > t_j$. In the pairwise RankELM (Zong & Huang, 2014), this problem is formulated as

$$\begin{aligned} &\min_{\boldsymbol{\beta}, \mathbf{x}i_{i,j}} \quad \|\boldsymbol{\beta}\|^2 + C\sum_{i,j}\xi_{i,j}^2 \\ &\text{s.t.} \quad \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} - \mathbf{h}(\mathbf{x}_j)\boldsymbol{\beta} = t_i - t_j - \xi_{i,j}, \quad i, j = 1, \ldots, N. \end{aligned} \tag{24}$$

It can be shown that the above formulation has a closed form solution:

$$\boldsymbol{\beta} = (\mathbf{I}/C + \mathbf{H}^T\mathbf{L}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{L}\mathbf{T}, \tag{25}$$

where $\mathbf{L}$ is the Laplacian matrix whose definition can be found in Zong and Huang (2014).

In Chen et al. (2014), a theoretical analysis on the generalization bound of ELM based ranking is established, which demonstrates that ELM based ranking can attain satisfactory learning rates under mild conditions.

4.9. ELM for semi-supervised learning

In many practical problems, such as text classification, spam filtering and natural language processing, obtaining training labels is nontrivial, while unlabeled data is available in large quantity and easy to collect. Semi-supervised learning is a technique that can utilize both labeled and unlabeled data to achieve higher prediction accuracy than pure supervised learning or unsupervised learning. Recently, ELM has been extended for semi-supervised learning (Li, Liu, & Ouyang, 2012; Li, Zhang, Xu, Luo, & Li, 2013; Huang et al., 2014; Liu, Chen, Liu, & Zhao, 2011; Tang & Han, 2010). In Li, Zhang, et al. (2013), a co-training approach have been proposed to train ELM. In this approach, an ELM is trained at each iteration, and the most confidently labeled data with their corresponding predicted labels are added to the labeled training set. In this way, the pseudo-labeled training set is augmented gradually and prediction accuracy can be improved. Huang et al. (2014) proposed a semi-supervised ELM (SS-ELM) based on manifold regularization. The SS-ELM inherits the advantages of classical ELM, such as it maintains the learning power and computational efficiency of ELM, and naturally handles the multi-class problem.

We denote the labeled data in the training set as $\{\mathbf{X}_l, \mathbf{T}_l\} = \{\mathbf{x}_i, \mathbf{t}_i\}_{i=1}^l$, and unlabeled data as $\mathbf{X}_u = \{\mathbf{x}_i\}_{i=1}^u$, where l and u are the number of labeled and unlabeled data, respectively. The formulation of SS-ELM is given by

$$\begin{aligned}&\min_{\boldsymbol{\beta}\in\mathbf{R}^{L\times m}} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{1}{2}\sum_{i=1}^{l} C_i\|\mathbf{e}_i\|^2 + \frac{\lambda}{2} Tr\left(\mathbf{F}^T\mathbf{L}\mathbf{F}\right)\\ &\text{s.t.}\quad \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta} = \mathbf{t}_i^T - \mathbf{e}_i^T, \quad i = 1,\ldots,l,\\ &\mathbf{f}_i(\mathbf{x}_i) = \mathbf{h}(\mathbf{x}_i)\boldsymbol{\beta}, \quad i = 1,\ldots,l+u,\end{aligned} \tag{26}$$

where $\mathbf{L} \in \mathbf{R}^{(l+u)\times(l+u)}$ is the *graph Laplacian* (Belkin, Niyogi, & Sindhwani, 2006; Huang et al., 2014) built from both labeled and unlabeled data, $\mathbf{F} \in \mathbf{R}^{(l+u)\times m}$ is the ELM output with its ith row equal to $\mathbf{f}(\mathbf{x}_i)$, and $Tr(\cdot)$ denotes the trace of a matrix. Here λ is a trade-off parameter associated with the manifold regularization term, and C_i is the penalty coefficient associated with the prediction error of the ith labeled sample. Similar to the weighted ELM discussed in Section 4.4, different C_i's are usually assigned to samples from different classes to prevent the algorithm overfitting to some dominant classes.

The optimization of (26) is convex, and the optimal solution to the output weights can be expressed in closed form:

$$\boldsymbol{\beta} = \begin{cases}\left(\mathbf{I}_L + \mathbf{H}^T\mathbf{C}\mathbf{H} + \lambda\mathbf{H}^T\mathbf{L}\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{C}\widetilde{\mathbf{T}}, & \text{if } l > L\\ \mathbf{H}^T\left(\mathbf{I}_{l+u} + \mathbf{C}\mathbf{H}\mathbf{H}^T + \lambda\mathbf{L}\mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{C}\widetilde{\mathbf{T}}, & \text{if } l \le L\end{cases} \tag{27}$$

where $\widetilde{\mathbf{T}} \in \mathbf{R}^{(l+u)\times L}$ is the augmented training target with its first l rows equal to $\mathbf{T}_l$ and the rest equal to 0.

4.10. Other variants of ELM

Due to its flexibility, ELM has many variants developed for special applications. We summarize some important variants of ELM as follows:

- Bayesian ELM
  Luo, et al. (2014) and Soria-Olivas et al. (2011)
- Fuzzy ELM
  Daliri (2012), Qu et al. (2011), Zhang and Ji (2013)
- Wavelet ELM
  Avci and Coteli (2012), Cao et al. (2010), Malathi et al. (2010, 2011)
- Complex ELM
  Huang, Li, Chen, and Siew (2008) and Savitha, Suresh, and Sundararajan (2012).
- Genetic/Evolutionary ELM
  Avci (2013), Cao, Lin and Huang (2012), Feng, Qian, and Zhang (2012), Li, Li, Zhai, and Shiu (2012), Sanchez-Monedero, Gutierrez, Fernandez-Navarro, and Hervas-Martinez (2011), Saraswathi, Sundaram, Sundararajan, Zimmermann, and Nilsen-Hamilton (2011), Sanchez-Monedero et al. (2010), Wang, Li, and Cao (2012).

5. ELM for unsupervised learning

5.1. ELM for embedding and clustering

ELM was primarily proposed for supervised learning tasks, e.g., regression and classification, while relatively few works consider ELM for unsupervised learning problems. Recently, Huang et al. (2014) proposed an unsupervised ELM (US-ELM) for unsupervised learning such as clustering or embedding, thus greatly expanding the applicability of ELM. The US-ELM is based on the manifold regularization framework, and its formulation is given by:

$$\begin{aligned}&\min_{\boldsymbol{\beta}\in\mathbf{R}^{L\times m}} \quad \|\boldsymbol{\beta}\|^2 + \lambda\, Tr\left(\boldsymbol{\beta}^T\mathbf{H}^T\mathbf{L}\mathbf{H}\boldsymbol{\beta}\right)\\ &\text{s.t}\quad \boldsymbol{\beta}^T\mathbf{H}^T\mathbf{H}\boldsymbol{\beta} = \mathbf{I}_m,\end{aligned} \tag{28}$$

where $\mathbf{L} \in \mathbf{R}^{N\times N}$ is the graph Laplacian built from the unlabeled data, $\mathbf{H} \in \mathbf{R}^{N\times L}$ is the hidden layer output matrix, and $\boldsymbol{\beta} \in \mathbf{R}^{L\times m}$ is the output weight matrix. The US-ELM maps the input data into a m-dimensional space in which the data is well clustered. Therefore, US-ELM can be used for nonlinear dimension reduction or embedding. If the learning task is clustering, then k-means algorithm is applied to cluster the embedded data in the new space.

It is shown in Huang et al. (2014) that the above optimization equals solving the following generalized eigenvalue problem (GEP):

$$\left(\mathbf{I}_L + \lambda\mathbf{H}^T\mathbf{L}\mathbf{H}\right)\mathbf{v} = \gamma\mathbf{H}^T\mathbf{H}\mathbf{v}. \tag{29}$$

In US-ELM, we first solve the above GEP to find the first $m+1$ generalized eigenvectors corresponding to the $m+1$ smallest eigenvalues. As in the algorithm of Laplacian eigenmaps (Belkin & Niyogi, 2003), the first eigenvector is discarded while the second through the $m+1$ eigenvectors is used to compute the output weights of US-ELM:

$$\boldsymbol{\beta} = [\widetilde{\mathbf{v}}_2, \widetilde{\mathbf{v}}_3, \ldots, \widetilde{\mathbf{v}}_{m+1}], \tag{30}$$

where $\widetilde{\mathbf{v}}_i = \mathbf{v}_i/\|\mathbf{H}\mathbf{v}_i\|$, $i = 2, \ldots, m+1$.

If the number of labeled data is fewer than the number of hidden neurons, problem (29) is underdetermined. In this case, the following alternative formulation is used:

$$\left(\mathbf{I}_u + \lambda\mathbf{L}\mathbf{H}\mathbf{H}^T\right)\mathbf{u} = \gamma\mathbf{H}\mathbf{H}^T\mathbf{u}, \tag{31}$$

and the output weights are computed by

$$\boldsymbol{\beta} = \mathbf{H}^T[\widetilde{\mathbf{u}}_2, \widetilde{\mathbf{u}}_3, \ldots, \widetilde{\mathbf{u}}_{m+1}], \tag{32}$$

where $\widetilde{\mathbf{u}}_i = \mathbf{u}_i/\|\mathbf{H}\mathbf{H}^T\mathbf{u}_i\|$, $i = 2, \ldots, m+1$.

The US-ELM is summarized in Algorithm 4.

Algorithm 4 The US-ELM algorithm (Huang et al., 2014)

```
Input:
  The training data: $\mathbf{X} \in \mathbf{R}^{N\times d}$;
Output:
  - For embedding task:
      The embedding in a $m$-dimensional space: $\mathbf{E} \in \mathbf{R}^{N\times m}$;
  - For clustering task:
      The label vector of cluster index: $\mathbf{t} \in \mathbf{N}_+^{N\times 1}$.

  Step 1: Construct the graph Laplacian $\mathbf{L}$ from $\mathbf{X}$.
  Step 2: Initiate an ELM network of $L$ hidden neurons with random input weights, and calculate the output matrix of the hidden neurons $\mathbf{H} \in \mathbf{R}^{N\times L}$.
  Step 3:
    - If $L \le N$
        Find the generalized eigenvectors $\mathbf{v}_2, \mathbf{v}_3, \ldots, \mathbf{v}_{m+1}$ of (29) corresponding to the second through the $m+1$ smallest eigenvalues. Compute output weights $\boldsymbol{\beta}$ from (30);
    - Else
        Find the generalized eigenvectors $\mathbf{u}_2, \mathbf{u}_3, \ldots, \mathbf{u}_{m+1}$ of (31) corresponding to the second through the $m+1$ smallest eigenvalues. Compute output weights $\boldsymbol{\beta}$ from (32);

  Step 4: Calculate the embedding matrix: $\mathbf{E} = \mathbf{H}\boldsymbol{\beta}$.
  Step 5 (For clustering only): Treat $\mathbf{E}$ as the new data matrix, and cluster them into $K$ clusters using the $k$-means algorithm. Let $\mathbf{t}$ be cluster assignment vector.
  return $\mathbf{E}$ (for embedding task) or $\mathbf{t}$ (for clustering task);
```

5.2. ELM for representational learning

Representational learning, e.g., stacked autoencoder (SAE) and stacked autodecoder (SDA), is effective in learning useful features for achieving high generalization performance, especially for handling big data. Though the popular deep neural networks equipped with unsupervised representational learning have yielded state-of-the-art performance on many difficult learning tasks, they are generally slow in training.

Recently, Kasun et al. (2013) proposed an ELM-based autoencoder for learning representations, which performs layer-wise representational learning using autoencoders learned by ELM, resulting in a multi-layer feedforward network. According to the experimental results on the MNIST OCR data set, this approach is several orders of magnitude faster than deep belief networks and deep Boltzmann machines, and the achieved accuracy is highly competitive with that of deep learning algorithms.

6. ELM for feature selection

In Benoit et al. (2013), a supervised feature selection method based on extreme learning machine (FS-ELM) was proposed. In the algorithm, ELM with leave-one-out error (LOO) is used for fast evaluation of generalization error to guide the search for optimal feature subsets.

Give the training data set $\{\mathbf{X}, \mathbf{T}\}$, the supervised feature selection problem can be expressed by:

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad \frac{1}{N}\sum_{i=1}^{N}(\mathbf{T}_i - f(\boldsymbol{\alpha}^T\mathbf{x}_i;\boldsymbol{\beta}))^2 \qquad (33)$$
$$\text{s.t} \quad \|\boldsymbol{\alpha}\|_0 = d_s \leq d, \quad \boldsymbol{\alpha} \in \{0, 1\}^d,$$

where f is a mapping function such as an ELM network, $\boldsymbol{\beta}$ are its parameters (output weights), d_s is the size of the feature subset, and $\|\cdot\|_0$ denotes the L_0-norm. Each binary variable α_i indicates whether the ith feature is selected or not.

Since the L_0-norm is non-continuous, the above optimization is difficult to solve. Thus in Benoit et al. (2013), the following relaxed L_1-norm problem is considered:

$$\min_{\tilde{\boldsymbol{\alpha}},\boldsymbol{\beta}} \quad \frac{1}{N}\sum_{i=1}^{N}(\mathbf{T}_i - f(\tilde{\boldsymbol{\alpha}}^T\mathbf{x}_i;\boldsymbol{\beta}))^2 + C_1\|\tilde{\boldsymbol{\alpha}}\|_1, \qquad (34)$$

where $C_1 \in \mathbf{R}^+$ is regularization coefficient, and $\|\cdot\|_1$ denotes the L_1-norm. Note that here $\tilde{\boldsymbol{\alpha}}$ is no longer a binary vector, but can take any real values. If the jth entry of $\tilde{\boldsymbol{\alpha}}$ is not strictly zero, then the corresponding jth feature is selected. To improve the sparsity, Benoit et al. (2013) proposed to discretize $\tilde{\boldsymbol{\alpha}}$ by limiting it to take values on the hypergrid $\{0, 1/k, 2/k, \ldots, 1\}^d$ with k non-zero values in each dimension. Then the gradient of the regularized training error of ELM is used to guide the search for optimal $\tilde{\boldsymbol{\alpha}}$. At each step, the search only considers the *direct neighbor* pointed to by the gradient. Here, a direct neighbor of $\tilde{\boldsymbol{\alpha}}$ is feature scaling $\tilde{\boldsymbol{\alpha}}'$ satisfying $\max_{j=1,\ldots,d}|\tilde{\alpha}_j - \tilde{\alpha}'_j| \leq 1/k$. The algorithm starts with a randomly initialized $\tilde{\boldsymbol{\alpha}}$, and gradually eliminates features by increasing C_1 in (34). Since the optimization of (34) is non-convex, multiple runs with differently initiated parameters are used to alleviate the local minimum problem. FS-ELM is summarized in Algorithm 5.

7. Implementation of ELM

7.1. Parallel implementation of ELM

ELM is well known for its computational efficiency, making it well-suited for large data processing. However, it is still worth speeding up its implementation in real applications. As pointed out in van Heeswijk, Miche, Oja, and Lendasse (2011), the running time of ELM is still significant when it is applied to large data sets, and the model selection process for optimal structure searching is time consuming. Parallel computing technique is one of the most popular approaches for fast implementation of learning algorithms. Several works in the literature have made efforts to implement ELM using parallel techniques such as MapReduce and GPU implementation (He et al., 2011; He, Shang, Zhuang, & Shi, 2013; van Heeswijk et al., 2011). Cloud computing techniques have also been applied in training ELM (Lin, Yin, Cai, Liu, & Li, 2013).

Algorithm 5 The FS-ELM algorithm (Benoit et al., 2013)

```
Input:
  The training data: $\mathbf{X} \in \mathbf{R}^{N\times d}$, $\mathbf{T} \in \mathbf{R}^{N\times m}$;
Output:
  The feature selection path (FSP) curve and the sparsity-error
  trade-off (SET) curve.
  for all restarts do:
    - Let $C_1 \leftarrow 0$, and randomly initiate $\tilde{\boldsymbol{\alpha}}$ on the hypergrid
      $\{0, 1/k, 2/k, \ldots, 1\}^d$.
    - Solve the parameters $\boldsymbol{\beta}$ in ELM and compute the regularized
      training error.
  end for
  while $\|\tilde{\boldsymbol{\alpha}}\|_0 > 0$
    - Estimate the generalization error using current $\tilde{\boldsymbol{\alpha}}$ and $\boldsymbol{\beta}$
    - Convert $\tilde{\boldsymbol{\alpha}}$ to binary vector $\boldsymbol{\alpha}$
    - Update the FSP and SET curve
    - Compute the gradient of regularized training error
    - Find the direct neighbor $\tilde{\boldsymbol{\alpha}}_{\text{new}}$ pointed by the gradient
    - Solve the parameters $\boldsymbol{\beta}_{\text{new}}$ in ELM corresponding $\tilde{\boldsymbol{\alpha}}_{\text{new}}$, and
      compute the regularized training error.
  end while
```

7.2. Hardware implementation of ELM

Hardware implementation of ELM has been realized in several recent works (Basu, Shuo, Zhou, Lim, & Huang, 2013; Decherchi, Gastaldo, Leoncini, & Zunino, 2012). Decherchi et al. (2012) addressed the implementation of ELM on two types of reconfigurable digital hardware, i.e., field-programmable gate array devices (FPGAs) and complex programmable logic devices (CPLDs). In their approach, a modified ELM model with L_1 regularization and hinge loss function is proposed to improve the sparsity of ELM. Thus the complexity of the trained ELM model to be implemented can be reduced. The circuit implementation procedure is summarized below. For detailed description, we refer readers to the reference Decherchi et al. (2012).

1. **Stage 1 (Input)**: The *input* stage generates an $(d + 1)$-dimensional vector, which feeds the subsequent stage sequentially through a Multiplexer.
2. **Stage 2a (Neuron)**: This stage finalizes the quantity of the inner product of input data and input weights.
3. **Stage 2b (Neuron)**: This stage applies the nonlinear activation function to the inner product obtained above.
4. **Stage 3 (Output)**: The *output* stage operates in pipeline with Stage 2 and hosts a MAC circuit to work out the ELM output.

Basu et al. (2013) proposed a silicon implementation of ELM using spiking neural circuits. The major building blocks include a silicon neuron for converting input current to a frequency of spikes, synapse circuit for converting incoming spike to current and Address Event Representation (AER) circuits for implementing

Table 2
Performance comparison of ELM-AE with state-of-the-art deep networks on MNIST data set (Kasun et al., 2013).

Algorithms	Testing accuracy (%)	Training time
ELM-AE	99.03	444.655 s
Deep belief network (DBN)	98.87	20,580 s
Deep Boltzmann machine (DBM)	99.05	68,246 s
Stacked auto-encoder (SAE)	98.6	>17 h
Stacked denoising auto-encoder (SDAE)	98.72	>17 h

large scale reconfigurable connectivity between neurons using time multiplexing of a single connection. In the same paper, details of possible architectures for implementing the ELM hidden neurons and output layer were given.

8. Empirical study of ELM and applications

Extensive empirical studies have been conducted on the performance of ELM and its variants. Comparisons with other state-of-the-art learning algorithms have also been made during the past years. Generally, ELM is fast and stable in training, easy in implementation, and accurate in modeling and prediction.

8.1. Comparison with SVM and its variant

A comprehensive empirical study on the training efficiency and generalization performance of ELM classification and regression was given in Huang, Zhou, et al. (2012). Comparisons was made with classical SVM, LS-SVM on more than forty data sets. Moreover, different types of activation functions in ELM were also studied. As verified by the experimental results, ELM achieved similar or better generalization performance for regression and binary class classification, and significantly better generalization performance on multiclass classification data sets. Meanwhile, ELM had a better scalability and much faster learning speed (up to several orders of magnitude).

8.2. Comparison with deep learning

Kasun et al. (2013) proposed an ELM-based autoencoder (ELM-AE) for representational learning and classification. In their experiments, a 784-700- 700-15000-10 multi-layer ELM network was tested on the well-known MNIST data set, which contains 60,000 images of handwritten digits for training and 10,000 images for testing. The results demonstrated that the multi-layer ELM-AE yielded matchable precision compared several state-of-the-art deep learning approaches, while it was much faster in training (see Table 2). Kasun et al. (2013) also studied how ELM autoencoder learns feature representations. They created 10 mini data sets containing digits 0–9 from the MNIST data set, and sent each mini data set through an ELM-AE (network structure: 784-20-784). It was found that the output weights $\boldsymbol{\beta}$ of the ELM-AE actually captures useful information from the original images. Huang et al. (2014) also showed that the unsupervised ELM outperforms deep auto-encoder on embedding and clustering tasks.

8.3. ELM for medical/biomedical application

Medical or biomedical data are usually with high dimensional features or has a large volume of samples. Thus advanced machine learning techniques such as SVM are usually used in medical or biomedical data analysis. As ELM has many advantages over other learning algorithms, it is interesting to see its application in this area. Indeed, during the past years, there have been many encouraging results on applying ELM for predicting protein–protein interactions (You et al., 2013), epileptic EEG patterns recognition (Song, Crowcroft, & Zhang, 2012; Song & Zhang, 2013; Yuan, Zhou, Li, & Cai, 2011a), EEG-based vigilance estimation (Shi & Lu, 2013), transmembrane beta-barrel chains detection (Savojardo, Fariselli, & Casadio, 2011), thyroid disease diagnosis (Li, Ouyang, Chen, & Liu, 2012), etc.

8.4. ELM for computer vision

ELM has been successfully applied to various computer vision tasks, such as face recognition (Baradarani, Wu, & Ahmadi, 2013; Choi, Toh, & Byun, 2012; He et al., 2014; Marques & Grana, 2013; Mohammed, Minhas, Wu, & Sid-Ahmed, 2011; Zong & Huang, 2011), human action recognition (Minhas, Baradarani, Seifzadeh, & Wu, 2010; Minhas, Mohammed, & Wu, 2012), terrain-based navigation (Kan, Lim, Ong, Tan, & Yeo, 2013), and fingerprint matching (Yang, Xie, et al., 2013).

8.5. ELM for image processing

ELM is also an attractive approach for image processing. For example, An and Bhanu (2012) proposed an efficient image super-resolution approach based on ELM which aims at generating high resolution images from low-resolution inputs. During the training process, image features were extracted as the input, and high-frequency components from the original high-resolution images were used as the target values. ELM then learns a model that maps the interpolated image to the high-frequency components. After training, the learned model are able to predict the high-frequency components from low-resolution images. In Li, Wang, and Chai (2013), ELM was used for burning state recognition of rotary kiln. In Chang, Han, Yao, Chen, and Xu (2010), ELM was used for change detection of land use and land cover.

8.6. ELM for system modeling and prediction

As traditional neural networks have been widely used in system modeling and perdition, ELM has great potential for developing efficient and accurate model for these applications. Xu, Dai, Dong, Zhang, and Meng (2013) developed an ELM-based predictor for real-time frequency stability assessment (FSA) of power systems. The inputs of the predictor are power system operational parameters, and the output is the frequency stability margin that measures the stability degree of the power system subject to a contingency. By off-line training with a frequency stability database, the predictor can be online applied for real-time FSA. The predictor was tested on New England 10-generator 39-bus test system, and the simulation results show that it can exactly accurately and efficiently predict the frequency stability. ELM has also been used for electricity price forecasting (Chen, Dong, et al., 2012), Temperature prediction of molten steel (Tian & Mao, 2010), sales forecasting (Chen & Ou, 2011; Wong & Guo, 2010), drying system modeling (Balbay, Kaya, & Sahin, 2012), security assessment of wind power system (Xu, Dong, Xu, Meng, & Wong, 2012; Xu, Dong, Zhao, Zhang, & Wong, 2012), etc.

Owing to its remarkable advantages, ELM has been widely adopted in many other applications. From the literature in the past

Table 3
Some recent applications of ELM.

Medical/Biomedical
Adamos et al. (2010), Barea et al. (2012), Boquete et al. (2012), Gao et al. (2013), Huang, Tan, et al. (2012), Karpagachelvi et al. (2012), Kaya and Uyar (2013), Kim et al. (2009), Li, Ouyang, et al. (2012), Osman et al. (2012), Pan et al. (2012), Rasheed and Rangwala (2012), Saraswathi et al. (2012), Savojardo et al. (2011), Shi et al. (2013), Yuan et al. (2011b, 2012), You et al. (2013),
Computer vision
Baradarani et al. (2013), Chen, Zhao, et al. (2012), Choi et al. (2012), He et al. (2014), Kan et al. (2013), Lemme et al. (2013), Malar et al. (2012), Marques and Grana (2013), Minhas, Baradarani, et al. (2010), Minhas et al. (2012), Nian et al. (2013), Yang, Xie, et al. (2013), Yu, Chen, et al. (2013), Zong and Huang (2011),
Image/video understanding and processing
An and Bhanu (2012), Bazi et al. (2014), Cao et al. (2013), Chang et al. (2010), Decherchi et al. (2013), Li, Wang, et al. (2013), Lu et al. (2013), Pan et al. (2012), Suresh et al. (2009), Wang, Huang, et al. (2011), Zhou et al. (2013),
Text classification and understanding
Cambria et al. (in press), Poria et al. (2014), Yang and Mao (2013), Zhao et al. (2011), Zheng et al. (2013),
System modeling and prediction
Balbay, Kaya, et al. (2012), Balbay, Avci, et al. (2012), Chen, Dong, et al. (2012), Chen and Ou (2011), Feng, Wang, et al. (2012), Lin et al. (2013), Li, Niu, et al. (2012), Minhas, Mohammed, et al. (2010), Saavedra-Moreno et al. (2013), Tan et al. (2012), Tian and Mao (2010), Wang and Do (2012), Wang, Qian, et al. (2013), Wefky et al. (2012), Wong and Guo (2010), Wong et al. (2013), Xia et al. (2012), Xu et al. (2013), Xu, Dong, Zhao, et al. (2012), Yan and Wang (2014), Yang et al. (2012), Yu et al. (2011), Zhang et al. (2013), Zhao et al. (2013), Zhao, Qi, et al. (2012),
Control and robotics Decherchi et al. (2011), Kosmatopoulos and Kouvelas (2009a, 2009b), Rong et al. (2011), Yu et al. (2012), Zhang and Slaughter (2011),
Chemical process
Liu, Jiang, et al. (2012), Wang, Xu, et al. (2012), Wang, Jin, et al. (2013), Zhang and Zhang (2011),
Fault detection and diagnosis
Chen, Ding, et al. (2012), Cheng et al. (2012), Creech and Jiang (2012), Kong et al. (2013), Martínez-Rego et al. (2010), Muhammad et al. (2013), Yang et al. (2011),
Time series analysis
Butcher et al. (2013), Sovilj et al. (2010), Wang and Han (2012), Zhang and Wang (2011a, 2011b),
Remote sensing
Pal et al. (2013), Pal (2009), Tang et al. (2014)

five years, we have witnessed its successful applications in text analysis, control system design, chemical process monitor, etc. Due to limited space, we are not able to cover all these applications. We summarize some of the recent applications of ELM in Table 3.

Considering the recent important extensions of ELM, such as ELM for clustering, feature selection, ranking, representational learning, there will be more potential fields that ELM can be applied to.

9. Conclusions

In this review, we reported the recent developments in theoretical studies and applications of ELM. Apparently, there is a growing interest in this research topic, and many significant theory justifications and interesting algorithms have been presented. Empirical results on a wide range of fields have demonstrated that ELM and its variants are efficient, accurate and easy to implement. Comparisons have shown the advantages of ELM over other state-of-the-art learning algorithms such as SVMs and deep learning algorithms. The following problems may be interesting for further research in the future:

- Improving ELM for handling high dimensional data. High dimensional data analysis is a challenging problem for traditional learning algorithms. Designing improved ELM algorithms such as introducing sparse coding techniques to ELM to efficiently handling high dimensional data will be an interesting topic.
- Justifying the random mechanism in ELM from a theoretical perspective, and studying the relationship between ELM and other algorithms. The random feature mapping is the key idea in ELM, which ensures its universal approximation capability and makes it very efficient in training. Meanwhile, random feature mapping also leads to better generalization performances and alleviates the problem of over-fitting. How to justify the effectiveness of this random mechanism needs further investigation. It will also be interesting to study the connection between ELM and other related algorithms which also adopt random mechanisms (including random sampling), such as random forest, and Adaboost.
- Investigating the impact of distribution form for generating the hidden layer parameters. From the universal approximation perspective, the hidden layer parameters can be generated according to any continuous distribution without sacrificing the universal approximation capability. However, the probability distribution function for generating these parameters may have an impact on the generalization performances in real applications. It is worth looking further into this problem and studying data-dependent generalization error bound for ELM.
- As shown in Table 4 and analyzed in Huang (2014) it seems that ELM is filling the gap between machine learning and biological learning mechanism. It is worth having detail study along this direction.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China under Grant Nos. 61273233 and 41427806, the Research Fund for the Doctoral Program of Higher Education under Grant Nos. 20120002110035 and 20130002130010, the National Key Technology R&D Program under Grant 2012BAF01B03, the Project of China Ocean Association under Grant DY125-25-02, and Tsinghua University Initiative Scientific Research Program under Grant 20131089300.

Table 4
Most conventional learning methods are contradictory to biological learning, while ELM resembles biological learning in many aspects.

Biological learning	Conventional learning methods (e.g., BP)	ELMs
Stable in a wide range of neurons (tens to thousands of neurons in each module)	Very sensitive to network size	Stable in a wide range of neurons (tens to thousands of neurons in each module)
Parallel implementation	Difficult for parallel implementation	Easy in parallel implementation
"Biological" implementation	Difficult for hardware implementation	Much easier in hardware implementation
Free of user specified parameters	Very sensitive to user specified parameters	Least human intervention
One module possibly works well for several types of applications	Different network types required for different type of applications	One network type works for different applications
Fast in micro learning point	Time consuming in each learning point	Fast in micro learning point
Nature in online sequential learning	Difficult for online sequential learning	Easy for online sequential learning
Fast speed and high accuracy	"Greedy" in best accuracy	Fast speed and high accuracy
"Biological Brains" are born before applications	"Artificial Brains (devised by conventional learning methods)" are chosen after applications are present	"Artificial Brains (devised by ELM)" can be generated before applications are present

References

Adamos, D. A., Laskaris, N. A., Kosmidis, E. K., & Theophilidis, G. (2010). Nass: An empirical approach to spike sorting with overlap resolution based on a hybrid noise-assisted methodology. *Journal of Neuroscience Methods*, *190*(1), 129–142.

An, L., & Bhanu, B. (2012). Image super-resolution by extreme learning machine. In *2012 19th IEEE international conference on image processing* (pp. 2209–2212). IEEE.

Avci, E. (2013). A new method for expert target recognition system: Genetic wavelet extreme learning machine (GAWELM). *Expert Systems with Applications*, *40*(10), 3984–3993.

Avci, E., & Coteli, R. (2012). A new automatic target recognition system based on wavelet extreme learning machine. *Expert Systems with Applications*, *39*(16), 12340–12348.

Bai, Z., Huang, G.-B., Wang, D., Wang, H., & Westover, M. B. (2014). Sparse extreme learning machine for classification. *IEEE Transactions on Cybernetics*.

Balbay, A., Avci, E., Sahin, O., & Coteli, R. (2012). Modeling of drying process of bittim nuts (*Pistacia terebinthus*) in a fixed bed dryer system by using extreme learning machine. *International Journal of Food Engineering*, *8*(4).

Balbay, A., Kaya, Y., & Sahin, O. (2012). Drying of black cumin (*Nigella sativa*) in a microwave assisted drying system and modeling using extreme learning machine. *Energy*, *44*(1), 352–357.

Baradarani, A., Wu, Q. M. J., & Ahmadi, M. (2013). An efficient illumination invariant face recognition framework via illumination enhancement and dd-dtcwt filtering. *Pattern Recognition*, *46*(1), 57–72.

Barea, R., Boquete, L., Ortega, S., Lopez, E., & Rodriguez-Ascariz, J. M. (2012). Eog-based eye movements codification for human computer interaction. *Expert Systems with Applications*, *39*(3), 2677–2683.

Basu, A., Shuo, S., Zhou, H. M., Lim, M. H., & Huang, G. B. (2013). Silicon spiking neurons for hardware implementation of extreme learning machines. *Neurocomputing*, *102*, 125–134.

Bazi, Y., Alajlan, N., Melgani, F., AlHichri, H., Malek, S., & Yager, R. R. (2014). Differential evolution extreme learning machine for the classification of hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, *11*(6), 1066–1070.

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, *15*(6), 1373–1396.

Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, *7*, 2399–2434.

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, *2*(1), 1–127.

Benoit, F., van Heeswijk, M., Miche, Y., Verleysen, M., & Lendasse, A. (2013). Feature selection for nonlinear models with extreme learning machines. *Neurocomputing*, *102*, 111–124.

Block, H. D. (1962). The perceptron: A model for brain function. I. *Reviewers of Modern Physics*, *34*(1), 123–135.

Block, H. D., Knight, J. B. W., & Rosenblatt, F. (1962). Analysis of a four-layer series-coupled perceptron. II. *Reviewers of Modern Physics*, *34*(1), 135–142.

Boquete, L., Miguel-Jimenez, J. M., Ortega, S., Rodriguez-Ascariz, J. M., Perez-Rico, C., & Blanco, R. (2012). Multifocal electroretinogram diagnosis of glaucoma applying neural networks and structural pattern analysis. *Expert Systems with Applications*, *39*(1), 234–238.

Branke, J. (1995). Evolutionary algorithms for neural network design and training. In *Proceedings of the first nordic workshop on genetic algorithms and its applications*.

Butcher, J. B., Verstraeten, D., Schrauwen, B., Day, C. R., & Haycock, P. W. (2013). Reservoir computing and extreme learning machines for non-linear time-series data analysis. *Neural Networks*, *38*, 76–89.

Cambria, E., Gastaldo, P., Bisio, F., & Zunino, R. (2014). An ELM-based model for affective analogical reasoning. *Neurocomputing*, in press.

Cao, J. W., Lin, Z. P., & Huang, G. B. (2010). Composite function wavelet neural networks with extreme learning machine. *Neurocomputing*, *73*(7-9), 1405–1416.

Cao, J. W., Lin, Z. P., & Huang, G. B. (2012). Self-adaptive evolutionary extreme learning machine. *Neural Processing Letters*, *36*(3), 285–305.

Cao, J. W., Lin, Z. P., Huang, G. B., & Liu, N. (2012). Voting based extreme learning machine. *Information Sciences*, *185*(1), 66–77.

Cao, F. L., Liu, B., & Park, D. S. (2013). Image classification based on effective extreme learning machine. *Neurocomputing*, *102*, 90–97.

Chang, N. B., Han, M., Yao, W., Chen, L. C., & Xu, S. G. (2010). Change detection of land use and land cover in an urban region with spot-5 images and partial lanczos extreme learning machine. *Journal of Applied Remote Sensing*, *4*.

Chen, S., Cowan, C., & Grant, P. (1991). Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, *2*(2), 302–309.

Chen, Q. S., Ding, J., Cai, J. R., & Zhao, J. W. (2012). Rapid measurement of total acid content (TAC) in vinegar using near infrared spectroscopy based on efficient variables selection algorithm and nonlinear regression tools. *Food Chemistry*, *135*(2), 590–595.

Chen, X., Dong, Z. Y., Meng, K., Ku, Y., Wong, K. P., & Ngan, H. W. (2012). Electricity price forecasting with extreme learning machine and bootstrapping. *IEEE Transactions on Power Systems*, *27*(4), 2055–2062.

Chen, F. L., & Ou, T. Y. (2011). Sales forecasting system based on gray extreme learning machine with taguchi method in retail industry. *Expert Systems with Applications*, *38*(3), 1336–1345.

Chen, H., Peng, J., Zhou, Y., Li, L., & Pan, Z. (2014). Extreme learning machine for ranking: Generalization analysis and applications. *Neural Networks*, *53*, 119–126.

Cheng, C., Tay, W. P., & Huang, G.-B. (2012). Extreme learning machines for intrusion detection. In *The 2012 International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). IEEE.

Chen, Y. Q., Zhao, Z. T., Wang, S. Q., & Chen, Z. Y. (2012). Extreme learning machine-based device displacement free activity recognition model. *Soft Computing*, *16*(9), 1617–1625.

Chen, Z. X. X., Zhu, H. Y. Y., & Wang, Y. G. G. (2013). A modified extreme learning machine with sigmoidal activation functions. *Neural Computing & Applications*, *22*(3–4), 541–550.

Choi, K., Toh, K. A., & Byun, H. (2012). Incremental face recognition for large-scale social network services. *Pattern Recognition*, *45*(8), 2868–2883.

Cortes, C., & Vapnik, V. (1995). Support vector machine. *Machine learning*, *20*(3), 273–297.

Creech, G., & Jiang, F. (2012). The application of extreme learning machines to the network intrusion detection problem. In *International conference of numerical analysis and applied mathematics*: *Vol. 1479* (pp. 1506–1511). AIP Publishing.

Daliri, M. R. (2012). A hybrid automatic system for the diagnosis of lung cancer based on genetic algorithm and fuzzy extreme learning machines. *Journal of Medical Systems*, *36*(2), 1001–1005.

Decherchi, S., Gastaldo, P., Dahiya, R. S., Valle, M., & Zunino, R. (2011). Tactile-data classification of contact materials using computational intelligence. *IEEE Transactions on Robotics*, *27*(3), 635–639.

Decherchi, S., Gastaldo, P., Leoncini, A., & Zunino, R. (2012). Efficient digital implementation of extreme learning machines for classification. *IEEE Transactions on Circuits and Systems II-Express Briefs*, *59*(8), 496–500.

Decherchi, S., Gastaldo, P., Zunino, R., Cambria, E., & Redi, J. (2013). Circular-ELM for the reduced-reference assessment of perceived image quality. *Neurocomputing*, *102*, 78–89.

Deng, J., Li, K., & Irwin, G. W. (2011). Fast automatic two-stage nonlinear model identification based on the extreme learning machine. *Neurocomputing*, *74*(16), 2422–2429.

Deng, W.-Y., Zheng, Q.-H., & Wang, Z.-M. (2013). Projection vector machine. *Neurocomputing*, *120*, 490–498.

Du, D. J., Li, K., Irwin, G. W., & Deng, J. (2013). A novel automatic two-stage locally regularized classifier construction method using the extreme learning machine. *Neurocomputing*, *102*, 10–22.

Feng, G. R., Huang, G. B., Lin, Q. P., & Gay, R. (2009). Error minimized extreme learning machine with growth of hidden nodes and incremental learning. *IEEE Transactions on Neural Networks*, *20*(8), 1352–1357.

Feng, G. R., Qian, Z. X., & Zhang, X. P. (2012). Evolutionary selection extreme learning machine optimization for regression. *Soft Computing*, *16*(9), 1485–1491.

Feng, Y., Wang, Y. N., & Yang, Y. M. (2012). Inverse kinematics solution for robot manipulator based on neural network under joint subspace. *International Journal of Computers Communications & Control*, *7*(3), 459–472.

Fernández-Delgado, M., Cernadas, E., Barro, S., Ribeiro, J., & Neves, J. (2014). Direct kernel perceptron (DKP): Ultra-fast kernel ELM-based classification with non-iterative closed-form weight calculation. *Neural Networks*, *50*, 60–71.

Frenay, B., & Verleysen, M. (2011). Parameter-insensitive kernel in extreme learning for non-linear support vector regression. *Neurocomputing*, *74*(16), 2526–2531.

Gao, J. F., Wang, Z., Yang, Y., Zhang, W. J., Tao, C. Y., Guan, J. A., et al. (2013). A novel approach for lie detection based on f-score and extreme learning machine. *Plos One*, *8*(6).

Gastaldo, P., Zunino, R., Cambria, E., & Decherchi, S. (2013). Combining ELM with random projections. *IEEE Intelligent Systems*, *28*(5), 18–20.

Hagan, M. T., & Menhaj, M. B. (1994). Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, *5*(6), 989–993.

He, Q., Du, C. Y., Wang, Q., Zhuang, F. Z., & Shi, Z. Z. (2011). A parallel incremental extreme svm classifier. *Neurocomputing*, *74*(16), 2532–2540.

He, Q., Shang, T. F., Zhuang, F. Z., & Shi, Z. Z. (2013). Parallel extreme learning machine for regression based on mapreduce. *Neurocomputing*, *102*, 52–58.

He, B., Xu, D., Nian, R., van Heeswijk, M., Yu, Q., Miche, Y., et al. (2014). Fast face recognition via sparse coding and extreme learning machine. *Cognitive Computation*, *6*, 264–277.

Horata, P., Chiewchanwattana, S., & Sunat, K. (2013). Robust extreme learning machine. *Neurocomputing*, *102*, 31–44.

Huang, G.-B. (2014). An insight to extreme learning machines: Random neurons, random features and kernels. Cognitive Computation (Online). http://dx.doi.org/10.1007/s12559-014-9255-2.

Huang, G.-B., & Chen, L. (2007). Convex incremental extreme learning machine. *Neurocomputing*, *70*(16), 3056–3062.

Huang, G.-B., & Chen, L. (2008). Enhanced random search based incremental extreme learning machine. *Neurocomputing*, *71*(16), 3460–3468.

Huang, G.-B., Chen, L., & Siew, C.-K. (2006). Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transactions on Neural Networks*, *17*(4), 879–892.

Huang, G.-B., Ding, X. J., & Zhou, H. M. (2010). Optimization method based extreme learning machine for classification. *Neurocomputing*, *74*(1-3), 155–163.

Huang, G.-B., Li, M.-B., Chen, L., & Siew, C.-K. (2008). Incremental extreme learning machine with fully complex hidden nodes. *Neurocomputing*, *71*(4), 576–583.

Huang, G., Song, S., Gupta, J., & Wu, C. (2014). Semi-supervised and unsupervised extreme learning machines. *IEEE Transactions on Cybernetics*,.

Huang, G., Song, S., & Wu, C. (2012). Orthogonal least squares algorithm for training cascade neural networks. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *59*(11), 2629–2637.

Huang, W., Tan, Z. M., Lin, Z., Huang, G. B., Zhou, J., Chui, C. K., et al. (2012). *A semi-automatic approach to the segmentation of liver parenchyma from 3D CT images with extreme learning machine*. IEEE.

Huang, G.-B., Wang, D. H., & Lan, Y. (2011). Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, *2*(2), 107–122.

Huang, G.-B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *42*(2), 513–529.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: theory and applications. *Neurocomputing*, *70*(1), 489–501.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. In: *IEEE International Joint Conference on Neural Networks, 2004*, vol. 2, pp. 985–990.

Kan, E. M., Lim, M. H., Ong, Y. S., Tan, A. H., & Yeo, S. P. (2013). Extreme learning machine terrain-based navigation for unmanned aerial vehicles. *Neural Computing & Applications*, *22*(3–4), 469–477.

Karpagachelvi, S., Arthanari, M., & Sivakumar, M. (2012). Classification of electrocardiogram signals with support vector machines and extreme learning machine. *Neural Computing & Applications*, *21*(6), 1331–1339.

Kasun, L. L. C., Zhou, H., & Huang, G.-B. (2013). Representational learning with ELMs for big data. *IEEE Intelligent Systems*, *28*(5), 31–34.

Kaya, Y., & Uyar, M. (2013). A hybrid decision support system based on rough set and extreme learning machine for diagnosis of hepatitis disease. *Applied Soft Computing*, *13*(8), 3429–3438.

Kim, J., Shin, H. S., Shin, K., & Lee, M. (2009). Robust algorithm for arrhythmia classification in ecg using extreme learning machine. *Biomedical Engineering Online*, *8*.

Kong, W. W., Zhang, C., Liu, F., Gong, A. P., & He, Y. (2013). Irradiation dose detection of irradiated milk powder using visible and near-infrared spectroscopy and chemometrics. *Journal of Dairy Science*, *96*(8), 4921–4927.

Kosmatopoulos, E. B., & Kouvelas, A. (2009a). Large scale nonlinear control system fine-tuning through learning. *IEEE Transactions on Neural Networks*, *20*(6), 1009–1023.

Kosmatopoulos, E. B., & Kouvelas, A. (2009b). Large scale nonlinear control system fine-tuning through learning. *IEEE Transactions on Neural Networks*, *20*(6), 1009–1023.

Kwok, T.-Y., & Yeung, D.-Y. (1997). Constructive algorithms for structure learning in feedforward neural networks for regression problems. *IEEE Transactions on Neural Networks*, *8*(3), 630–645.

Lahoz, D., Lacruz, B., & Mateo, P. M. (2013). A multi-objective micro genetic ELM algorithm. *Neurocomputing*, *111*, 90–103.

Lan, Y., Soh, Y. C., & Huang, G. B. (2009). Ensemble of online sequential extreme learning machine. *Neurocomputing*, *72*(13–15), 3391–3395.

Landa-Torres, I., Ortiz-Garcia, E. G., Salcedo-Sanz, S., Segovia-Vargas, M. J., Gil-Lopez, S., Miranda, M., et al. (2012). Evaluating the internationalization success of companies through a hybrid grouping harmony search-extreme learning machine approach. *IEEE Journal of Selected Topics in Signal Processing*, *6*(4), 388–398.

Lemme, A., Freire, A., Barreto, G., & Steil, J. (2013). Kinesthetic teaching of visuomotor coordination for pointing by the humanoid robot icub. *Neurocomputing*, *112*, 179–188.

Le, Q., Sarlos, T., & Smola, A. (2013). Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*.

Liang, N.-Y., Huang, G.-B., Saratchandran, P., & Sundararajan, N. (2006). A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Transactions on Neural Networks*, *17*(6), 1411–1423.

Li, B., Li, Y. B., & Rong, X. W. (2013). The extreme learning machine learning algorithm with tunable activation function. *Neural Computing & Applications*, *22*(3–4), 531–539.

Li, L., Liu, D., & Ouyang, J. (2012). A new regularization classification method based on extreme learning machine in network data. *Journal of Information & Computational Science*, *9*(12), 3351–3363.

Li, Y. J., Li, Y., Zhai, J. H., & Shiu, S. (2012). Rts game strategy evaluation using extreme learning machine. *Soft Computing*, *16*(9), 1627–1637.

Lin, S. J., Chang, C. H., & Hsu, M. F. (2013). Multiple extreme learning machines for a two-class imbalance corporate life cycle prediction. *Knowledge-Based Systems*, *39*, 214–223.

Li, G. Q., Niu, P. F., Liu, C., & Zhang, W. P. (2012). Enhanced combination modeling method for combustion efficiency in coal-fired boilers. *Applied Soft Computing*, *12*(10), 3132–3140.

Lin, S., Liu, X., Fang, J., & Xu, Z. (2014). Is extreme learning machine feasible? a theoretical assessment (part II). *IEEE Transactions on Neural Networks nd Learning Systems*.

Lin, J., Yin, J., Cai, Z., Liu, Q., & Li, K. (2013). A secure and practical mechanism of outsourcing extreme learning machine in cloud computing. *IEEE Intelligent Systems*, *28*(5), 35–38.

Li, L. N., Ouyang, J. H., Chen, H. L., & Liu, D. Y. (2012). A computer aided diagnosis system for thyroid disease using extreme learning machine. *Journal of Medical Systems*, *36*(5), 3327–3337.

Li, K., Peng, J.-X., & Irwin, G. W. (2005). A fast nonlinear model identification method. *IEEE Transactions on Automatic Control*, *50*(8), 1211–1216.

Liu, J. F., Chen, Y. Q., Liu, M. J., & Zhao, Z. T. (2011). SELM: Semi-supervised ELM with application in sparse calibrated location estimation. *Neurocomputing*, *74*(16), 2566–2572.

Liu, X. Y., Gao, C. H., & Li, P. (2012). A comparative analysis of support vector machines and extreme learning machines. *Neural Networks*, *33*, 58–66.

Liu, G. H., Jiang, H., Xiao, X. H., Zhang, D. J., Mei, C. L., & Ding, Y. H. (2012). Determination of process variable ph in solid-state fermentation by ft-nir spectroscopy and extreme learning machine (ELM). *Spectroscopy and Spectral Analysis*, *32*(4), 970–973.

Liu, X., Lin, S., Fang, J., & Xu, Z. (2014). Is extreme learning machine feasible? a theoretical assessment (part I). *IEEE Transactions on Neural Networks nd Learning Systems*.

Liu, N., & Wang, H. (2010). Ensemble based extreme learning machine. *IEEE Signal Processing Letters*, *17*(8), 754–757.

Liu, Y., Xu, X.J., & Wang, C.Y. (2009). Simple ensemble of extreme learning machine. In *Proceedings of the 2009 2nd international congress on image and signal processing*, Vols. 1–9.

Li, W. T., Wang, D. H., & Chai, T. Y. (2013). Burning state recognition of rotary kiln using ELMs with heterogeneous features. *Neurocomputing*, *102*, 144–153.

Li, K., Zhang, J., Xu, H., Luo, S., & Li, H. (2013). A semi-supervised extreme learning machine method based on co-training. *Journal of Computational Information Systems*, *9*(1), 207–214.

Luo, J., Vong, C.-M., & Wong, P.-K. (2014). Sparse bayesian extreme learning machine for multi-classification. *IEEE Transactions on Neural Networks nd Learning Systems*, *25*(4), 836–843.

Lu, B., Wang, G. R., Yuan, Y., & Han, D. (2013). Semantic concept detection for video based on extreme learning machine. *Neurocomputing*, *102*, 176–183.

Malar, E., Kandaswamy, A., Chakravarthy, D., & Dharan, A. G. (2012). A novel approach for detection and classification of mammographic microcalcifications using wavelet analysis and extreme learning machine. *Computers in Biology and Medicine*, *42*(9), 898–905.

Malathi, V., Marimuthu, N. S., & Baskar, S. (2010). Intelligent approaches using support vector machine and extreme learning machine for transmission line protection. *Neurocomputing*, *73*(10–12), 2160–2167.

Malathi, V., Marimuthu, N. S., Baskar, S., & Ramar, K. (2011). Application of extreme learning machine for series compensated transmission line protection. *Engineering Applications of Artificial Intelligence*, *24*(5), 880–887.

Man, Z. H., Lee, K., Wang, D. H., Cao, Z. W., & Khoo, S. Y. (2012). Robust single-hidden layer feedforward network-based pattern classifier. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(12), 1974–1986.

Man, Z. H., Lee, K., Wang, D. H., Cao, Z. W., & Miao, C. Y. (2011). A new robust training algorithm for a class of single-hidden layer feedforward neural networks. *Neurocomputing*, *74*(16), 2491–2501.

Marques, I., & Grana, M. (2013). Fusion of lattice independent and linear features improving face identification. *Neurocomputing*, *114*, 80–85.

Martinez-Martinez, J. M., Escandell-Montero, P., Soria-Olivas, E., Martin-Guerrero, J. D., Magdalena-Benedito, R., & Gomez-Sanchis, J. (2011). Regularized extreme learning machine for regression problems. *Neurocomputing*, *74*(17), 3716–3721.

Miche, Y., Sorjamaa, A., Bas, P., Simula, O., Jutten, C., & Lendasse, A. (2010). OP-ELM: Optimally pruned extreme learning machine. *IEEE Transactions on Neural Networks*, *21*(1), 158–162.

Minhas, R., Baradarani, A., Seifzadeh, S., & Wu, Q. M. J. (2010). Human action recognition using extreme learning machine based on visual vocabularies. *Neurocomputing*, *73*(10–12), 1906–1917.

Minhas, R., Mohammed, A. A., & Wu, Q. M. J. (2010). A fast recognition framework based on extreme learning machine using hybrid object information. *Neurocomputing*, *73*(10–12), 1831–1839.

Minhas, R., Mohammed, A. A., & Wu, Q. M. J. (2012). Incremental learning in human action recognition based on snippets. *IEEE Transactions on Circuits and Systems for Video Technology*, *22*(11), 1529–1541.

Mohammed, A. A., Minhas, R., Wu, Q. M. J., & Sid-Ahmed, M. A. (2011). Human face recognition based on multidimensional pca and extreme learning machine. *Pattern Recognition*, *44*(10–11), 2588–2597.

Muhammad, I. G., Tepe, K. E., & Abdel-Raheem, E. (2013). QAM equalization and symbol detection in OFDM systems using extreme learning machine. *Neural Computing & Applications*, *22*(3–4), 491–500.

Martínez-Rego, D., Fontenla-Romero, O., Pérez-Sánchez, B., & Alonso-Betanzos, A. (2010). Fault prognosis of mechanical components using on-line learning neural networks. In *Artificial Neural Networks–ICANN 2010* (pp. 60–66). Springer.

Nian, R., He, B., & Lendasse, A. (2013). 3d object recognition based on a geometrical topology model and extreme learning machine. *Neural Computing & Applications*, *22*(3–4), 427–433.

Osman, M. K., Mashor, M. Y., & Jaafar, H. (2012). Performance comparison of extreme learning machine algorithms for mycobacterium tuberculosis detection in tissue sections. *Journal of Medical Imaging and Health Informatics*, *2*(3), 307–312.

Pal, M. (2009). Extreme-learning-machine-based land cover classification. *International Journal of Remote Sensing*, *30*(14), 3835–3841.

Pal, M., Maxwell, A. E., & Warner, T. A. (2013). Kernel-based extreme learning machine for remote-sensing image classification. *Remote Sensing Letters*, *4*(9), 853–862.

Pan, C., Park, D. S., Lu, H. J., & Wu, X. P. (2012). Color image segmentation by fixation-based active learning with ELM. *Soft Computing*, *16*(9), 1569–1584.

Platt, J. (1991). A resource-allocating network for function interpolation. *Neural Computation*, *3*(2), 213–225.

Plutowski, M., Cottrell, G., & White, H. (1996). Experience with selecting exemplars from clean data. *Neural Networks*, *9*(2), 273–294.

Poggio, T., & Girosi, F. (1990). Networks for approximation and learning. *Proceedings of the IEEE*, *78*(9), 1481–1497.

Poria, S., Cambria, E., Winterstein, G., & Huang, G.-B. (2014). Sentic patterns: Dependency-based rules for concept-level sentiment analysis. (http://dx.doi.org/10.1016/j.knosys.2014.05.005) Knowledge-Based Systems.

Qu, Y. P., Shang, C. J., Wu, W., & Shen, Q. (2011). Evolutionary fuzzy extreme learning machine for mammographic risk analysis. *International Journal of Fuzzy Systems*, *13*(4), 282–291.

Rahimi, A., & Recht, B. (2007). Random features for large-scale kernel machines. In *Advances in neural information processing systems*, Vol. 3, p. 5.

Rahimi, A., & Recht, B. (2008a). Uniform approximation of functions with random bases. In *Communication, control, and computing, 2008 46th annual allerton conference on* (pp. 555–561). IEEE.

Rahimi, A., & Recht, B. (2008b). Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In: *Advances in neural information processing systems*, pp. 1313–1320.

Rao, C. R., & Mitra, S. K. (1971). *Generalized inverse of matrices and its applications, Vol. 7*. New York: Wiley.

Rasheed, Z., & Rangwala, H. (2012). Metagenomic taxonomic classification using extreme learning machines. *Journal of Bioinformatics and Computational Biology*, *10*(5).

Rong, H. J., Huang, G. B., Sundararajan, N., & Saratchandran, P. (2009). Online sequential fuzzy extreme learning machine for function approximation and classification problems. *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, *39*(4), 1067–1072.

Rong, H.-J., Ong, Y.-S., Tan, A.-H., & Zhu, Z. (2008). A fast pruned-extreme learning machine for classification problem. *Neurocomputing*, *72*(1), 359–366.

Rong, H. J., Suresh, S., & Zhao, G. S. (2011). Stable indirect adaptive neural controller for a class of nonlinear system. *Neurocomputing*, *74*(16), 2582–2590.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, *65*(6), 386–408.

Rosenblatt, F. (1962). *Principles of neurodynamics: perceptrons and the theory of brain mechanisms*. New York: Spartan Books.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(9), 533–536.

Saavedra-Moreno, B., Salcedo-Sanz, S., Carro-Calvo, L., Gascon-Moreno, J., Jimenez-Fernandez, S., & Prieto, L. (2013). Very fast training neural-computation techniques for real measure-correlate-predict wind operations in wind farms. *Journal of Wind Engineering and Industrial Aerodynamics*, *116*, 49–60.

Sanchez-Monedero, J., Gutierrez, P. A., Fernandez-Navarro, F., & Hervas-Martinez, C. (2011). Weighting efficient accuracy and minimum sensitivity for evolving multi-class classifiers. *Neural Processing Letters*, *34*(2), 101–116.

Sanchez-Monedero, J., Hervas-Martinez, C., Gutierrez, P. A., Ruz, M. C., Moreno, M. C. R., & Cruz-Ramirez, M. (2010). Evaluating the performance of evolutionary extreme learning machines by a combination of sensitivity and accuracy measures. *Neural Network World*, *20*(7), 899–912.

Saraswathi, S., Fernandez-Martinez, J. L., Kolinski, A., Jernigan, R. L., & Kloczkowski, A. (2012). Fast learning optimized prediction methodology (flopred) for protein secondary structure prediction. *Journal of Molecular Modeling*, *18*(9), 4275–4289.

Saraswathi, S., Sundaram, S., Sundararajan, N., Zimmermann, M., & Nilsen-Hamilton, M. (2011). ICGA-PSO-ELM approach for accurate multiclass cancer classification resulting in reduced gene sets in which genes encoding secreted proteins are highly represented. *IEEE-ACM Transactions on Computational Biology and Bioinformatics*, *8*(2), 452–463.

Savitha, R., Suresh, S., & Sundararajan, N. (2012). Fast learning circular complex-valued extreme learning machine (CC-ELM) for real-valued classification problems. *Information Sciences*, *187*, 277–290.

Savojardo, C., Fariselli, P., & Casadio, R. (2011). Improving the detection of transmembrane beta-barrel chains with n-to-1 extreme learning machines. *Bioinformatics*, *27*(22), 3123–3128.

Saxe, A., Koh, P. W., Chen, Z., Bhand, M., Suresh, B., & Ng, A. Y. (2011). On random weights and unsupervised feature learning. In: *Proceedings of the 28th international conference on machine learning* (pp. 1089–1096).

Schmidt, W. F., Kraaijveld, M. A., & Duin, R. P. (1992). Feed forward neural networks with random weights. In *Proceedings of 11th IAPR international conference on pattern recognition methodology and systems* (pp. 1–4). Netherlands: Hague.

Shi, J., Cai, Y., Zhu, J., Zhong, J., & Wang, F. (2013). Semg-based hand motion recognition using cumulative residual entropy and extreme learning machine. *Medical & Biological Engineering & Computing*, *51*(4), 417–427.

Shi, L. C., & Lu, B. L. (2013). Eeg-based vigilance estimation using extreme learning machines. *Neurocomputing*, *102*, 135–143.

Song, Y., Crowcroft, J., & Zhang, J. (2012). Automatic epileptic seizure detection in EEGs based on optimized sample entropy and extreme learning machine. *Journal of Neuroscience Methods*, *210*, 132–146.

Song, Y. D., & Zhang, J. X. (2013). Automatic recognition of epileptic EEG patterns via extreme learning machine and multiresolution feature extraction. *Expert Systems with Applications*, *40*(14), 5477–5489.

Soria-Olivas, E., Gomez-Sanchis, J., Jarman, I., Vila-Frances, J., Martinez, M., Magdalena, J. R., et al. (2011). Belm: Bayesian extreme learning machine. *IEEE Transactions on Neural Networks*, *22*(3), 505–509.

Sovilj, D., Sorjamaa, A., Yu, Q., Miche, Y., & Severin, E. (2010). OPELM and OPKNN in long-term prediction of time series using projected input data. *Neurocomputing*, *73*(10–12), 1976–1986.

Suresh, S., Babu, R. V., & Kim, H. J. (2009). No-reference image quality assessment using modified extreme learning machine classifier. *Applied Soft Computing*, *9*(2), 541–552.

Suykens, J. A., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing letters*, *9*(3), 293–300.

Tan, Y. H., Dong, R. L., Chen, H., & He, H. (2012). Neural network based identification of hysteresis in human meridian systems. *International Journal of Applied Mathematics and Computer Science*, *22*(3), 685–694.

Tang, J., Deng, C., Huang, G.-B., & Zhao, B. (2014). Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine. *IEEE Transactions on Geoscience and Remote Sensing*, http://dx.doi.org/10.1109/TGRS.2014.2335751.

Tang, X. L., & Han, M. (2010). Ternary reversible extreme learning machines: the incremental tri-training method for semi-supervised classification. *Knowledge and Information Systems*, *23*(3), 345–372.

Tian, H. X., & Mao, Z. Z. (2010). An ensemble ELM based on modified AdaBoost.RT algorithm for predicting the temperature of molten steel in ladle furnace. *IEEE Transactions on Automation Science and Engineering*, *7*(1), 73–80.

van Heeswijk, M., Miche, Y., Oja, E., & Lendasse, A. (2011). GPU-accelerated and parallelized ELM ensembles for large-scale regression. *Neurocomputing*, *74*(16), 2430–2437.

Vapnik, V. (2000). *The nature of statistical learning theory*. Springer.

Wang, Y. G., Cao, F. L., & Yuan, Y. B. (2011). A study on effectiveness of extreme learning machine. *Neurocomputing*, *74*(16), 2483–2490.

Wang, D. H., & Do, H. T. (2012). Computational localization of transcription factor binding sites using extreme learning machines. *Soft Computing*, *16*(9), 1595–1606.

Wang, N., Er, M.-J., & Han, M. (2014a). Parsimonious extreme learning machine using recursive orthogonal least squares. *IEEE Transactions on Neural Networks*, http://dx.doi.org/10.1109/TNNLS.2013.2296048.

Wang, N., Er, M. J., & Han, M. (2014b). Parsimonious extreme learning machine using recursive orthogonal least squares. *IEEE Transactions on Neural Networks nd Learning Systems*,.

Wang, X. Y., & Han, M. (2012). Multivariate chaotic time series prediction based on extreme learning machine. *Acta Physica Sinica*, *61*(8).

Wang, L., Huang, Y. P., Luo, X. Y., Wang, Z., & Luo, S. W. (2011). Image deblurring with filters learned by extreme learning machine. *Neurocomputing*, *74*(16), 2464–2474.

Wang, J. N., Jin, J. L., Geng, Y., Sun, S. L., Xu, H. L., Lu, Y. H., et al. (2013). An accurate and efficient method to predict the electronic excitation energies of bodipy fluorescent dyes. *Journal of Computational Chemistry*, *34*(7), 566–575.

Wang, G. T., Li, P., & Cao, J. T. (2012). Variable activation function extreme learning machine based on residual prediction compensation. *Soft Computing*, *16*(9), 1477–1484.

Wang, H., Qian, G., & Feng, X. Q. (2013). Predicting consumer sentiments using online sequential extreme learning machine and intuitionistic fuzzy sets. *Neural Computing & Applications*, *22*(3–4), 479–489.

Wang, X. Z., Shao, Q. Y., Miao, Q., & Zhai, J. H. (2013). Architecture selection for networks trained with extreme learning machine using localized generalization error model. *Neurocomputing*, *102*, 3–9.

Wang, J. N., Xu, H. L., Sun, S. L., Gao, T., Li, H. Z., Li, H., et al. (2012). An effective method for accurate prediction of the first hyperpolarizability of alkalides. *Journal of Computational Chemistry*, *33*(2), 231–236.

Wefky, A., Espinosa, F., de Santiago, L., Revenga, P., Lazaro, J. L., & Martinez, M. (2012). Electrical drive radiated emissions estimation in terms of input control using extreme learning machines. *Mathematical Problems in Engineering*.

White, H. (1989). An additional hidden unit test for neglected nonlinearity in multilayer feedforward networks. In: *Proceedings of the international conference on neural networks* (pp. 451–455).

White, H. (2006). Approxiate nonlinear forecasting methods. In G. Elliott, C. W. J. Granger, & A. Timmermann (Eds.), *Handbook of economics forecasting* (pp. 460–512). New York: Elsevier.

White, H. (1992). *Artificial neural networks: approximation and learning theory*. Blackwell Publishers, Inc.

Widrow, B., Greenblatt, A., Kim, Y., & Park, D. (2013). The No-Prop algorithm: A new learning algorithm for multilayer neural networks. *Neural Networks*, *37*, 182–188.

Wilamowski, B. M., & Yu, H. (2010). Neural network learning without backpropagation. *IEEE Transactions on Neural Networks*, *21*(11), 1793–1803.

Wong, W. K., & Guo, Z. X. (2010). A hybrid intelligent model for medium-term sales forecasting in fashion retail supply chains using extreme learning machine and harmony search algorithm. *International Journal of Production Economics*, *128*(2), 614–624.

Wong, K. I., Wong, P. K., Cheung, C. S., & Vong, C. M. (2013). Modeling and optimization of biodiesel engine performance using advanced machine learning methods. *Energy*, *55*, 519–528.

Xia, M., Zhang, Y. C., Weng, L. G., & Ye, X. L. (2012). Fashion retailing forecasting based on extreme learning machine with adaptive metrics of inputs. *Knowledge-Based Systems*, *36*, 253–259.

Xu, Y., Dai, Y. Y., Dong, Z. Y., Zhang, R., & Meng, K. (2013). Extreme learning machine-based predictor for real-time frequency stability assessment of electric power systems. *Neural Computing & Applications*, *22*(3–4), 501–508.

Xu, Y., Dong, Z. Y., Xu, Z., Meng, K., & Wong, K. P. (2012). An intelligent dynamic security assessment framework for power systems with wind power. *IEEE Transactions on Industrial Informatics*, *8*(4), 995–1003.

Xu, Y., Dong, Z. Y., Zhao, J. H., Zhang, P., & Wong, K. P. (2012). A reliable intelligent system for real-time dynamic security assessment of power systems. *IEEE Transactions on Power Systems*, *27*(3), 1253–1263.

Yang, X., & Mao, K. (2013). Reduced ELMs for causal relation extraction from unstructured text. *IEEE Intelligent Systems*, *28*(5), 48–52.

Yang, Y. M., Wang, Y. N., & Yuan, X. F. (2013). Parallel chaos search based incremental extreme learning machine. *Neural Processing Letters*, *37*(3), 277–301.

Yang, Y. M., Wang, Y. N., & Yuan, X. F. (2012). Bidirectional extreme learning machine for regression problem and its learning effectiveness. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(9), 1498–1505.

Yang, J. C., Xie, S., Yoon, S., Park, D., Fang, Z. J., & Yang, S. Y. (2013). Fingerprint matching based on extreme learning machine. *Neural Computing & Applications*, *22*(3-4), 435–445.

Yang, H. M., Xu, W. J., Zhao, J. H., Wang, D. H., & Dong, Z. Y. (2011). Predicting the probability of ice storm damages to electricity transmission facilities based on ELM and copula function. *Neurocomputing*, *74*(16), 2573–2581.

Yan, Z., & Wang, J. (2014). Robust model predictive control of nonlinear systems with unmodeled dynamics and bounded uncertainties based on neural networks. *IEEE Transactions on Neural Networks nd Learning Systems*, *25*(3), 457–469.

Yao, X. (1993). A review of evolutionary artificial neural networks. *International Journal of Intelligent Systems*, *8*(4), 539–567.

Ye, Y. B., Squartini, S., & Piazza, F. (2013). Online sequential extreme learning machine in nonstationary environments. *Neurocomputing*, *116*, 94–101.

You, Z. H., Lei, Y. K., Zhu, L., Xia, J. F., & Wang, B. (2013). Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis. *BMC Bioinformatics*, *14*.

Yuan, Y. B., Wang, Y. G., & Cao, F. L. (2011). Optimization approximation solution for regression problem based on extreme learning machine. *Neurocomputing*, *74*(16), 2475–2482.

Yuan, Q., Zhou, W., Li, S., & Cai, D. (2011a). Epileptic EEG classification based on extreme learning machine and nonlinear features. *Epilepsy Research*, *96*, 29–38.

Yuan, Q., Zhou, W. D., Li, S. F., & Cai, D. M. (2011b). Epileptic eeg classification based on extreme learning machine and nonlinear features. *Epilepsy Research*, *96*(1–2), 29–38.

Yuan, Q., Zhou, W. D., Zhang, J., Li, S. F., Cai, D. M., & Zeng, Y. J. (2012). EEG classification approach based on the extreme learning machine and wavelet transform. *Clinical EEG and Neuroscience*, *43*(2), 127–132.

Yu, H., Chen, Y., Liu, J., & Huang, G.-B. (2013). An adaptive and iterative online sequential ELM based multi-degree-of-freedom gesture recognition system. *IEEE Intelligent Systems*, *28*(6), 55–59.

Yu, Y., Choi, T. M., & Hui, C. L. (2011). An intelligent fast sales forecasting model for fashion products. *Expert Systems with Applications*, *38*(6), 7373–7379.

Yu, Y., Choi, T. M., & Hui, C. L. (2012). An intelligent quick prediction algorithm with applications in industrial control and loading problems. *IEEE Transactions on Automation Science and Engineering*, *9*(2), 276–287.

Yu, D., & Deng, L. (2012). Efficient and effective algorithms for training single-hidden-layer neural networks. *Pattern Recognition Letters*, *33*(5), 554–558.

Yu, Q., Miche, Y., Eirola, E., van Heeswijk, M., Severin, E., & Lendasse, A. (2013). Regularized extreme learning machine for regression with missing data. *Neurocomputing*, *102*, 45–51.

Zhai, J. H., Xu, H. Y., & Wang, X. Z. (2012). Dynamic ensemble extreme learning machine based on sample entropy. *Soft Computing*, *16*(9), 1493–1502.

Zhang, R., Dong, Z. Y., Xu, Y., Meng, K., & Wong, K. P. (2013). Short-term load forecasting of australian national electricity market by an ensemble model of extreme learning machine. *IET Generation Transmission & Distribution*, *7*(4), 391–397.

Zhang, W. B., & Ji, H. B. (2013). Fuzzy extreme learning machine for classification. *Electronics Letters*, *49*(7), 448–449.

Zhang, R., Lan, Y., Huang, G. B., & Xu, Z. B. (2012). Universal approximation of extreme learning machine with adaptive growth of hidden nodes. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(2), 365–371.

Zhang, Y., & Slaughter, D. C. (2011). Hyperspectral species mapping for automatic weed control in tomato under thermal environmental stress. *Computers and Electronics in Agriculture*, *77*(1), 95–104.

Zhang, X., & Wang, H. L. (2011a). Incremental regularized extreme learning machine based on cholesky factorization and its application to time series prediction. *Acta Physica Sinica*, *60*(11).

Zhang, X., & Wang, H. L. (2011b). Selective forgetting extreme learning machine and its application to time series prediction. *Acta Physica Sinica*, *60*(8).

Zhang, Y. W., & Zhang, P. C. (2011). Optimization of nonlinear process based on sequential extreme learning machine. *Chemical Engineering Science*, *66*(20), 4702–4710.

Zhao, Z. P., Li, P., & Xu, X. Z. (2013). Forecasting model of coal mine water inrush based on extreme learning machine. *Applied Mathematics & Information Sciences*, *7*(3), 1243–1250.

Zhao, L., Qi, J. Q., Wang, J., & Yao, P. J. (2012). The study of using an extreme learning machine for rapid concentration estimation in multi-component gas mixtures. *Measurement Science & Technology*, *23*(8).

Zhao, X. G., Wang, G. R., Bi, X., Gong, P. Z., & Zhao, Y. H. (2011). XML document classification based on ELM. *Neurocomputing*, *74*(16), 2444–2451.

Zhao, J. W., Wang, Z. H., & Park, D. S. (2012). Online sequential extreme learning machine with forgetting mechanism. *Neurocomputing*, *87*, 79–89.

Zheng, W. B., Qian, Y. T., & Lu, H. J. (2013). Text categorization based on regularization extreme learning machine. *Neural Computing & Applications*, *22*(3–4), 447–456.

Zhou, Z. H., Zhao, J. W., & Cao, F. L. (2013). Surface reconstruction based on extreme learning machine. *Neural Computing & Applications*, *23*(2), 283–292.

Zong, W. W., & Huang, G.-B. (2014). Learning to rank with extreme learning machine. *Neural processing letters*, *39*(2), 155–166.

Zong, W. W., & Huang, G.-B. (2011). Face recognition based on extreme learning machine. *Neurocomputing*, *74*(16), 2541–2551.

Zong, W. W., Huang, G.-B., & Chen, Y. Q. (2013). Weighted extreme learning machine for imbalance learning. *Neurocomputing*, *101*, 229–242.