Neural Process Lett
DOI 10.1007/s11063-016-9496-z

A Kind of Parameters Self-adjusting Extreme Learning Machine

Peifeng Niu[1,2] · Yunpeng Ma[1] · Mengning Li[1] · Shanshan Yan[3] · Guoqiang Li[1,2]

Abstract Extreme learning machine (ELM) is a kind of feed-forward single hidden layer neural network, whose input weights and thresholds of hidden layers are generated randomly. Because the output-weights of ELM are calculated by the least-square method, the ELM presents a high speed on training and testing. However, the random input-weights and thresholds of hidden layers are not the best parameters, which can not pledge the training goals of the ELM to achieve the global minimum. In order to obtain the optimal input-weights and bias of hidden layer, this paper proposes the self-adjusting extreme learning machine, called SA-ELM. Based on the idea of the ameliorated teaching learning based optimization, the input-weights and the bias of hidden layer of extreme learning machine are adjusted with "teaching phase" and "learning phase" to minimize the objective function values. The SA-ELM is applied to the eight benchmark functions to test its validity and feasibility. Compared with ELM and fast learning network, the SA-ELM owns good regression accuracy and generalization performance. Besides, the SA-ELM is applied to build the thermal efficiency model of a 300 MW pulverized coal furnace. The experiment results reveal that the proposed algorithm owns engineering practical application value.

Keywords Extreme learning machine · Fast learning network · Ameliorated teaching learning based optimization · Boiler thermal efficiency model

✉ Yunpeng Ma
18232362368@163.com

1 Key Laboratory of Industrial Computer Control Engineering of Hebei Province, Yanshan University, Qinhuangdao 066004, China

2 National Engineering Research Center for Equipment and Technology of Cold Strip Rolling, Qinhuangdao, China

3 Hydropower Station of Administration of Taolinkou Reservoir, Qinhuangdao, China

Published online: 30 January 2016

1 Introduction

1.1 Background

In recent years, machine learning method has drawn public attention, such as deep Boltzman machines [1], support vector machine [2], extreme learning machine (ELM) [3], artificial neural network [4]. All of them have been applied in many areas widely, such as classification problems [5–8], regression approximation problems [9–11], image processing problem [12–15], and intelligent control [16–18].

ELM proposed by Huang et al. in (2004) is a novel single feed-forward artificial neural network. Its input-weights and bias of hidden layer of ELM are generated randomly; the output-weights of ELM are determined analytically and unique. So the ELM takes a higher speed for training than traditional feed-forward neural networks and could avoid falling into local minimum. Undoubtedly, the random input-weights and thresholds of hidden layers are not the best parameters, which can not pledge the training goals of the ELM to achieve the global minimum. In order to solve the aforementioned problem, many scholars proposed numerous modified ELM, in [19], the adaptive differential evolution algorithm is combined with ELM to optimize the parameters of the hidden layer of ELM; Zhu et al. [20] proposed a hybrid ELM that using differential evolution to optimize the input-weights of ELM and the output-weights are determined analytically by M-P generalized inverse. Matias et al. [21] puts forward optimal ELM, which is applied to optimize feed-forward single hidden layer neural network structure and the structure parameter; Han et al. [22] used modified PSO to optimize the input-weights and the bias of hidden layer of extreme learning machine; Li et al. [23,24] proposed a hybrid method called tuning ELM improved by artificial bee colony (ABC), which can obtain the optimal input-weight and threshold and improve the generalization performance of ELM.

1.2 Motivations

In order to obtain the optimal input-weights and bias of hidden layer, this paper proposes a self-adjusting extreme learning machine, called SA-ELM. Based on the idea of the ameliorated teaching learning based optimization (ATLBO) [25], the input-weights and the bias of hidden layer of extreme learning machine are adjusted in "teaching phase" and "learning phase" to minimize the objective function values. The ATLBO is a novel high-effective meta-heuristic optimization algorithm, which is proposed by Li et al. During the research [25], the performance of ATLBO has been compared with ABC and Gravitational Search Algorithm. The ATLBO has a fast convergence speed and better performance accuracy. In addition, the ATLBO has already been applied to optimize the NOx emission model of coal-fire boiler. Therefore, the paper adopts the thought of ATLBO to find better input-weights and the bias of hidden layer of extreme learning machine. The SA-ELM is applied to the eight benchmark functions to test its validity. Compared with original ELM and fast learning network (FLN) [26], the SA-ELM possesses good regression accuracy and generalization performance. Experimental result shows the SA-ELM could search for the better network parameters. And after that, making use of what have get to model save the best parameters of extreme learning machine to model the pulverized coal furnace thermal efficiency.

Along with great energy consumption, especially the coal consumption in fire power plant, boiler combustion model problem of power plants attracted technical personnel. Considering the nonlinear, large lagging and strong coupling combustion process of the pulverized coal furnace, it is very difficult to build its thermal efficiency model bases theory. Here the artificial neural network can solve the nonlinear problem, which identifies system model based on data. ELM owns fast learning speed than traditional neural network and has overcome some issues in traditional neural network, such as local minima, the stop condition, number of iteration. Therefore, this paper adopts SA-ELM to build the pulverized coal furnace thermal efficiency model. Experimental results show that the proposed model is convenient and simple and has engineering practical application value.

1.3 Contributions

This paper proposes SA-ELM, which can find the best input-weights and thresholds automatically. The main contributions of this paper are enumerated as follows:

1. Based on the idea of ameliorated teaching learning based optimization (ATLBO), SA-ELM is proposed to avoid falling into local minima and search for the best network parameters of ELM.
2. The proposed SA-ELM is applied to model the pulverized coal furnace thermal efficiency. Theoretically, it is difficult to build the fire-coal boiler thermal efficiency model, so an artificial neural network is essential to solve the problem. The proposed SA-ELM is a new kind of feed-forward neural network, which could be used to build the thermal efficiency model. Based on the combustion model, an operating mode of boiler is optimized in the following work.

The paper is organized as follows: basic knowledge and related works is given in Sect. 2; the proposed SA-ELM machine is given in Sect. 3; Sect. 4 shows the performance evaluation of the SA-ELM; Sect. 5 proves the real-world design problem; Complexity analysis in Sect. 6; the conclusions of this paper in Sect. 7.

2 Basic Knowledge and Related Works

In this chapter, we introduce some contents about traditional ELM, FLN and ameliorated teaching-learning-based optimization algorithm briefly.

2.1 Extreme Learning Machine [3]

Suppose that there are N stochastic samples $(\boldsymbol{x}_i, \boldsymbol{t}_i)$, where $\boldsymbol{x}_i = [x_1^i, x_2^i, \ldots, x_n^i]^{\mathrm{T}}$ is the ith training sample of an n-dimensional vector, $\boldsymbol{t}_i = [t_1^i, t_2^i, \ldots, t_l^i]^{\mathrm{T}}$ is the target vector. Here, the input-weights $\boldsymbol{W}_{M\times n}$, the bias of hidden layer is $\boldsymbol{b}_{M\times 1}$, and the output-weights are $\boldsymbol{\beta}_{l\times M}$, where M delegates the number of hidden nodes. The matrix $\boldsymbol{W}_{M\times n}$ and $\boldsymbol{b}_{M\times 1}$ are generated randomly. The target output of the ELM can be calculated by the follow equation:

$$\boldsymbol{t}_k^i = \sum_{j=1}^{M} \beta_{kj} g_j(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{X}), \quad k = 1, 2, \ldots, l \tag{1}$$

where $g_j(\cdot)$ is the activation function of hidden layer.

Equation (1) could be rewritten as follows:

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

Where H is the output matrix of hidden layer and defined as:

$$\mathbf{H}(\mathbf{W}, \mathbf{b}, \mathbf{X}) = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\mathbf{w}_M \cdot \mathbf{x}_1 + b_M) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \cdots & g(\mathbf{w}_M \cdot \mathbf{x}_N + b_N) \end{bmatrix}_{N\times M}$$

$$\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_M]^{\mathrm{T}}_{l\times M} \text{ and } \mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_N]^{\mathrm{T}}_{l\times N}. \tag{3}$$

The output weight matrix $\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_M]^{\mathrm{T}}_{l\times M}$ can be determined analytically by the minimum norm least square solution:

$$\tilde{\boldsymbol{\beta}} = \arg\min_{\beta} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\| = \mathbf{H}^{+}\mathbf{T} \tag{4}$$

where $\mathbf{H}^{+}$ is the M-P generalized inverse of H. If the $\mathbf{H}$ is nonsingular, Eq. (4) can be rewritten as:

$$\tilde{\boldsymbol{\beta}} = (\mathbf{H}^{\mathrm{T}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{T} \tag{5}$$

The ELM could be summarized as follows:

(1) Randomly assign the input weights matrix $\boldsymbol{W}_{M\times n}$ and the bias matrix $\boldsymbol{b}_{M\times 1}$.
(2) Calculate the output matrix $\mathbf{H}$ of the hidden layer by Eq. (3).
(3) Calculate the output weight matrix $\boldsymbol{\beta}_{l\times M}$.

2.2 Fast Learning Network

Fast learning network [26], proposed by Li et al. in 2013, is a novel double parallel forward neural network. Its structure of the fast learning machine is shown in Fig. 1

Given N arbitrary distinct samples (x_i, y_i), where $x_i = [x_1^i, x_2^i, \ldots, x_n^i]^{\mathrm{T}}$ is the ith training sample of an n-dimensional vector quantity, $y_i = [y_1^i, y_2^i, \ldots, y_l^i]^{\mathrm{T}}$ is the target vector. The fast learning network has m hidden nodes, $\mathbf{W}^{in}$ is the $m \times n$ input weight matrix, $\mathbf{b} = [b_1, b_2, \ldots, b_m]^{\mathrm{T}}$ is the bias matrix of hidden layer, $\mathbf{W}^{oh}$ is the $l \times m$ matrix which connects hidden layer with output layer. $\mathbf{W}^{oi}$ is the connective weight matrix between the output layer and the input layer. $\mathbf{c} = [c_1, c_2, \ldots, c_l]^{\mathrm{T}}$ is the bias matrix of output layer. The

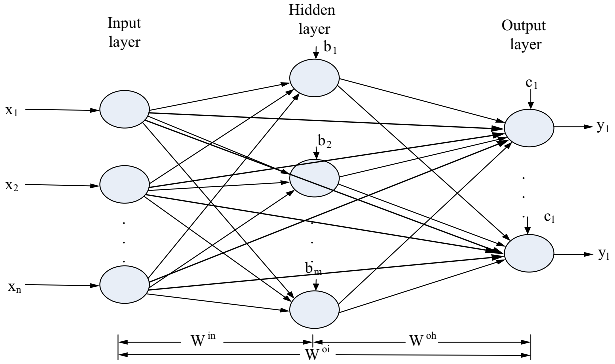

Fig. 1 Structure of the fast learning network

jth output target vector of the FLN could be written as follows:

$$\mathbf{y}_j = f\left(\mathbf{W}^{oi}\mathbf{x}_j + \mathbf{c} + \sum_{k=1}^{m}\mathbf{W}_k^{oh} g(\mathbf{W}_k^{in}\mathbf{x}_j + b_k)\right), \quad j = 1, 2, \ldots, N \tag{6}$$

where $f(\cdot)$ and $g(\cdot)$ are the active function of output layer and hidden layer, separately. Equation (6) could be rewritten as follows:

$$\begin{aligned}\mathbf{Y} &= f(\mathbf{W}^{oi}\mathbf{X} + \mathbf{W}^{oh}\mathbf{G} + \mathbf{c})\\ &= f\left([\mathbf{W}^{oi}\mathbf{W}^{oh}\mathbf{c}]\begin{bmatrix}\mathbf{X}\\ \mathbf{G}\\ \mathbf{I}\end{bmatrix}\right)\\ &= f\left(\mathbf{W}\begin{bmatrix}\mathbf{X}\\ \mathbf{G}\\ \mathbf{I}\end{bmatrix}\right)\end{aligned} \tag{7}$$

Where,

$$\begin{aligned}&\mathrm{G}(\mathbf{W}_1^{in}, \ldots, \mathbf{W}_m^{in}, b_1, \ldots, b_m, \mathrm{x}_1, \ldots, \mathrm{x}_N)\\ &= \begin{bmatrix} g(\mathbf{W}_1^{in}x_1 + b_1) & \cdots & g(\mathbf{W}_1^{in}x_N + b_1)\\ \vdots & \ddots & \vdots\\ g(\mathbf{W}_m^{in}x_1 + b_m) & \cdots & g(\mathbf{W}_m^{in}x_N + b_m)\end{bmatrix}_{m\times N}\end{aligned} \tag{8}$$

$$\mathbf{W} = [\mathbf{W}^{oi}\mathbf{W}^{oh}\mathbf{c}]_{l\times(n+m+1)} \tag{9}$$

$$\mathbf{I} = [11\cdots 1]_{1\times N} \tag{10}$$

Where the $\mathbf{W}$ is the output weight matrix, $\mathbf{G}$ is the hidden layer output matrix of the FLN. The FLN is proposed based on ELM, the output weights $\mathbf{W}$ could be determined analytically as follow:

$$\left\| f\left(\hat{\mathbf{W}}\begin{bmatrix}\mathbf{X}\\ \mathbf{G}\\ \mathbf{I}\end{bmatrix} - \mathbf{Y}\right)\right\| = \min_{\mathbf{W}}\left\| f\left(\mathbf{W}\begin{bmatrix}\mathbf{X}\\ \mathbf{G}\\ \mathbf{I}\end{bmatrix} - \mathbf{Y}\right)\right\| \tag{11}$$

For an invertible activation function $f(\cdot)$, the output weights are also analytically determined by Eq. (12).

$$\left\|\hat{\mathbf{W}}\begin{bmatrix}\mathbf{X}\\ \mathbf{G}\\ \mathbf{I}\end{bmatrix} - f^{-1}(\mathbf{Y})\right\| = \min_{\mathbf{W}}\left\|\mathbf{W}\begin{bmatrix}\mathbf{X}\\ \mathbf{G}\\ \mathbf{I}\end{bmatrix} - f^{-1}(\mathbf{Y})\right\| \tag{12}$$

where $f^{-1}(\cdot)$ is the invertible function of $f(\cdot)$.

According to the Moore–Penrose generalized inverse [27], the minimum norm least-squares solution of Eq. (12):

$$\hat{\mathbf{W}} = f^{-1}(\mathbf{Y})\begin{bmatrix}\mathbf{X}\\ \mathbf{G}\\ \mathbf{I}\end{bmatrix}^{+} = f^{-1}(\mathbf{Y})\mathbf{H}^{+} \tag{13}$$

where $\mathbf{H} = \left[\mathbf{X}\ \mathbf{G}\ \mathbf{I}\right]^{\mathrm{T}}$.

$$\begin{cases}\mathbf{W}^{oi} = \hat{\mathbf{W}}(1 : l, 1 : n)\\ \mathbf{W}^{oh} = \hat{\mathbf{W}}(1 : l, n+1 : (n+m))\\ \mathbf{c} = \hat{\mathbf{W}}(1 : l, n+m+1)\end{cases} \tag{14}$$

The learning algorithm of FLN can be summarized as follows:

(1) Randomly assign the input-weight matrix $\mathbf{W}^{in}$ and the bias matrix $\mathbf{b}$.
(2) Calculate the hidden output-weight $\mathbf{G}$ using Eq. (8).
(3) Calculate the connection matrix $\mathbf{W}$ using Eq. (13).
(4) Confirm FLN's model parameters by Eq. (14).

2.3 Ameliorated Teaching-Learning-Based Optimization

The teaching-learning-based optimization algorithm (TLBO) [28] is a novel swarm optimization algorithm proposed by Rao et al. This algorithm takes some advantages, such as a simple structure, few parameters and a high running speed. In order to improve the solution quality and to quicken the convergence speed of TLBO, Li et al. propose [25]. In ATLBO, there are three main improvements relate to TLBO as follows the elitist strategy replaces the greedy selection mechanism; an inertia weight function and an acceleration coefficient function are applied to quicken the "teaching phase" and "learning phase", separately.

In teaching phase of the ATLBO, the performance of individual updates as follows:

$$X_{new,i} = \omega_i X_{old,i} + \phi_i (M_{new} - T_F M_i) \tag{15}$$

$$\omega_i = 1/(1 + \exp(-\mathit{fit}(i)/ap)^{iter}) \tag{16}$$

$$\phi_i = 1/(1 + \exp(-\mathit{fit}(i)/ap) \times iter) \tag{17}$$

where ω_i is the inertia weight, which controls impact of the previous solution; ϕ_i is the acceleration coefficient that decides the maximum step size; $\mathit{fit}(i)$ is the fitness of the ith learner; *ap* is the maximum fitness in the first iteration; *iter* is the current iteration.

In learning phase, the learners enhance their knowledge by communicating with each other. A learner can update his mark by the following form:

$$X_{new,i} = \begin{cases} X_{old,i} + \varphi_i (X_j - X_i) & \text{if} \quad f(X_i) \le f(X_j) \\ X_{old,i} + \psi_i (X_{best} - X_i) & \text{if} \quad f(X_i) > f(X_j) \end{cases} \tag{18}$$

$$\varphi_i = 1 - \exp(\mathit{fit}(X_j) - \mathit{fit}(X_i)) \tag{19}$$

$$\psi_i = 1 - \exp(\mathit{fit}(X_{best}) - \mathit{fit}(X_i)) \tag{20}$$

where X_{best} is the best learner in a class; φ_i and ψ_i are the acceleration coefficients that decide the step size depending on the difference between two learner.

3 The Proposed Self-adjusting ELM

The input-weights and threshold of hidden layers of extreme learning machine are randomly assigned. And the weights matrix of the output layer is analytically calculated in one step, which is unique. However, the weights of input layer and thresholds of hidden layer are not optimal parameters. Therefore, in order to find the optimal weights of input layer and thresholds of hidden layer, this paper proposes a new algorithm called self- adjusting extreme learning machine.

The SA-ELM is proposed based on the idea of teaching-learning-based optimization algorithm (called TLBO), whose input-weight values and the bias of hidden nodes are adjusted via 'teaching phase' and 'learning phase' of TLBO. The SA-ELM is described as follows in detail.

In "teaching phase" of SA-ELM, the input-weight values and the thresholds of hidden nodes are determined randomly, which are regardd as learners' marks of all kinds of courses, namely $\theta = [w_{11}, w_{12}, \ldots, w_{1n}, w_{21}, w_{22}, \ldots, w_{2n}, \ldots, w_{m1}, \ldots, w_{mn}, b_1, \ldots, b_m]$, where w_{ij} is the weight value of the connection between the ith hidden node and the jth input node, $w_{ij} \in [-1, 1]$, b_i is the bias of the ith hidden node, $b_i \in [0, 1]$, n is the number of input nodes, m is the number of hidden nodes, $(n + 1) \times m$ is the dimension of learners' mark, that is $(n + 1) \times m$ parameters need to optimize. In this algorithm, the fitness function sets as follows:

$$f(\theta) = \sqrt{\frac{\sum_j^N \left\| \sum_k^m \rho_k g(w_k x_j + b_k) - y_j \right\|_2^2}{N}} \tag{21}$$

where ρ is the output-weight matrix, N is the number of training samples.

In this step, the target function fitness value of each other is calculated preferentially. And the learner who has the minimum fitness value is selected as the teacher. The new mark of a learner mainly depends on the previous mark $\theta_{old,i}$ and the difference $(\theta_{best} - \theta_{old,i})$ between the teacher and the previous mark. The updating mechanism of the structure parameters of the SA-ELM are established as follow:

$$\theta_{new,i} = \omega_i \theta_{old,i} + \phi_i(\theta_{best} - \theta_{old,i}) \tag{22}$$

$$\omega_i = 1/(1 + \exp(-f(i)/a)^{iter}) \tag{23}$$

$$\phi_i = 1/(1 + \exp(-f(i)/a) \times iter) \tag{24}$$

Where ω_i is the inertia weight, which controls the influence of the previous mark. ϕ_i is the acceleration coefficient, which determines the maximum step size. In Eqs. (23) and (24), a is the max target function fitness value in the first iteration. *iter* is the current iteration.

In "learning phase" of the SA-ELM, the learners increase their marks via communicating with the others. In this step, the elitist mechanism is adopted to update the structure parameters. In ith iteration, the ith learner updates his marks as follows:

$$\theta_{new,i} = \begin{cases} \theta_{old,i} + \alpha_i(\theta_j - \theta_i) & \text{if} \quad f(\theta_i) \le f(\theta_j) \\ \theta_{old,i} + \beta_i(\theta_{best} - \theta_i) & \text{if} \quad f(\theta_i) > f(\theta_j) \end{cases} \tag{25}$$

$$\alpha_i = 1 - \exp(f(\theta_j) - f(\theta_i)) \tag{26}$$

$$\beta_i = 1 - \exp(f(\theta_{best}) - f(\theta_i)) \tag{27}$$

In Eq. (25), θ_{best} is the best learner; α_i and β_i are acceleration coefficients, which decide the step size depending on the difference between two learners.

The learning algorithm of the SA-ELM as follows:

(1) Randomly generated the input-weights and the bias of hidden layer, and set the population number and target function.
(2) "Teaching phase". Calculate fitness value, updating structure parameters by Eq. (22).
(3) "Learning phase". Adopt elitist strategy to update parameters by Eq. (25).

According to the above explanations about the SA-ELM, the flowchart of the SA-ELM algorithm is shown in Fig. 2.

There are some special notations in weights restoration equations have shown in Table 1 clearly.

4 Experimental Study and Discussion

In order to testify the performance of the proposed SA-ELM algorithm, the SA-ELM is applied to the benchmark problems listed in Table 2. This paper adopts 8 regression

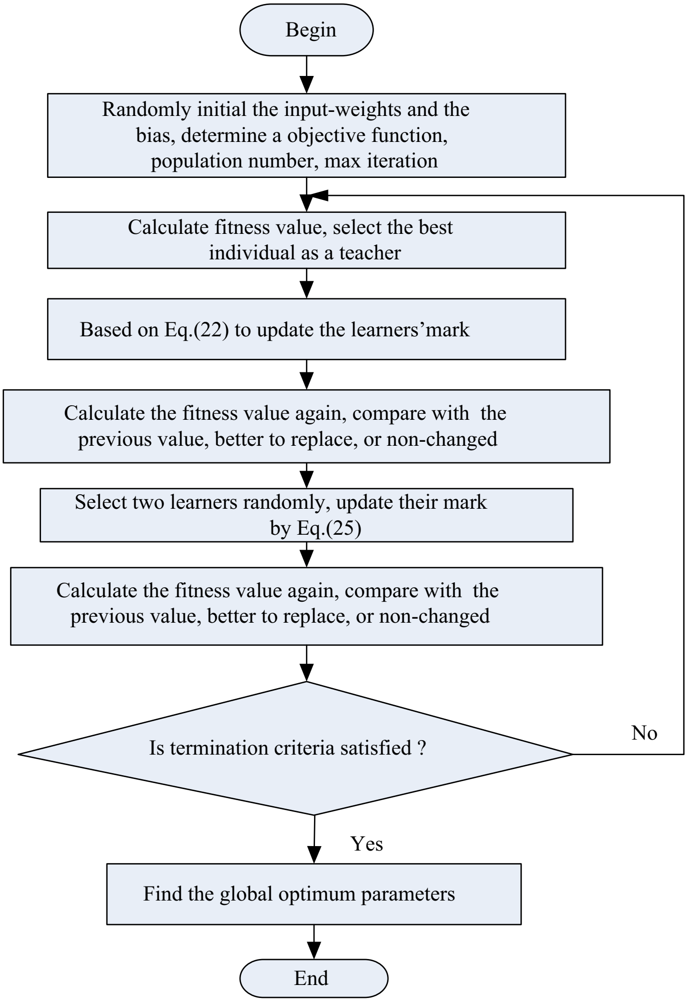

Fig. 2 Flowchart of the SA-ELM algorithm

applications (Servo, Auto MPG, Forest fires, Delta elevators, Bank domains, Machine CPU, Boston housing), which are chosen from the web: http://www.liaad.up.pt/~ltorgo/Regression/DtaSets.html. All evaluations for original ELM, FLN and SA-ELM are carried out in Windows XP and Matlab 7.8.0 environment running on a desktop with AMD Athlon(tm), 64×2 Dual Core Processor 2.61 GHz and 1 G RAM.

In order to evaluate the superiority of the SA-ELM, the ELM and FLN are employed to compare the regression accuracy and generalization performance on eight regression applications. In this paper, the regression accuracy is defined as the root-mean-square error (RMSE) of training or testing model. The RMSE is smaller, the regression accuracy is higher. And the main idea of the generalization performance of one model is that the building data model possesses widely application range. For training data, the building model has high model regression accuracy, so are the testing data. That is the output of building model is appro-

Table 1 Notation table about SA-ELM

Notations	Implications
θ	Input-weight and bias assemble
ρ	The output-weight matrix
$\theta_{old,i}$	The previous ith solution
$\theta_{new,i}$	The new ith solution
θ_{best}	The best solution
ω_i	The inertia weight
ϕ_i	The acceleration coefficient
$iter$	The current iteration
a	The max fitness value in the first iteration
α_i	The acceleration coefficients
β_i	The acceleration coefficients

Table 2 Specification of real-world regression data sets

Data sets	Attributes	Observations	Training set	Testing set
Auto MPG	7	392	215	177
Servo	4	167	91	76
Bank domains	8	8192	4505	3687
Machine CPU	6	209	114	95
Abalone	8	4177	2297	1880
Forest fires	12	517	284	233
Delta elevators	6	9517	5234	4283
Boston housing	13	506	278	228

priate with the un-training data. In this case, considering that the building model possesses better generalization performance ability.

The number of hidden nodes is set as 20 for three methods randomly, the hidden activation function is the sigmoid function Eq. (28). The population size of the SA-ELM is 40 and the iteration number is 200. Each experiment is repeated 30 times. Its mean and standard deviations (SD) of RMSE for the three methods are given in Table 3 for training samples and Table 4 for testing samples. The RMSE represents the regression accuracy of network model, and the SD delegates steady for model. The RMSE is smaller, the regression accuracy is higher. The SD is smaller, the building model is steadier.

$$g(x) = \frac{1}{1 + \exp(-x)} \tag{28}$$

In addition, the paper has applied the ABC algorithm to optimize the input-weights and bias of extreme learning machine, comparing with SA-ELM. Li et al. has proved that the ATLBO has better performance than ABC and Gravitation Search Algorithm (GSA) in [25]. In order to prove the SA-ELM has generalization ability, the ABC-ELM is applied to compare with SA-ELM. The testing results are shown in Tables 5 and 6. Note that the population of ABC sets 40, the iteration is 200.

Table 3 Comparison of the mean and its standard deviation of RMSE for ELM, FLN and SA-ELM using training samples

Data sets	ELM		FLN		SA-ELM	
	RMSE		RMSE		RMSE	
	Mean	SD	Mean	SD	Mean	SD
Auto MPG	1.43×10^{-1}	5.9×10^{-3}	1.32×10^{-1}	3.60×10^{-3}	1.25×10^{-1}	2.38×10^{-3}
Servo	1.98×10^{-1}	1.21×10^{-2}	1.83×10^{-1}	9.90×10^{-3}	1.29×10^{-1}	2.48×10^{-3}
Bank domains	1.33×10^{-1}	1.85×10^{-2}	9.55×10^{-2}	6.23×10^{-4}	9.02×10^{-2}	1.71×10^{-3}
Machine CPU	5.15×10^{-2}	4.30×10^{-3}	4.36×10^{-2}	2.80×10^{-3}	3.35×10^{-2}	6.40×10^{-4}
Abalone	1.62×10^{-1}	1.50×10^{-3}	1.58×10^{-1}	8.44×10^{-4}	1.55×10^{-1}	2.94×10^{-4}
Forest fires	1.24×10^{-1}	7.43×10^{-4}	1.22×10^{-1}	4.67×10^{-4}	1.30×10^{-1}	2.47×10^{-3}
Delta elevators	1.25×10^{-1}	2.03×10^{-4}	1.25×10^{-1}	2.61×10^{-4}	1.06×10^{-1}	2.02×10^{-4}
Boston housing	1.88×10^{-1}	1.24×10^{-2}	1.53×10^{-1}	9.3×10^{-3}	1.59×10^{-1}	3.05×10^{-3}

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{29}$$

$$SD = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{n-1}} \tag{30}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\ \% \tag{31}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{32}$$

As shown in Table 3, for the training samples of data sets, compared with ELM and FLN, the SA-ELM shows better regression accuracy on the regression applications (Auto MPG, Servo, Bank domains, Machine CPU, Abalone, Delta elevators, Boston housing). Besides, for the SD, the SA-ELM shows better steady performance on six applications (Auto MPG, Servo, Machine CPU, Abalone, Delta elevators, Boston housing) than ELM and FLN. As shown in Table 4, for the testing samples of data sets, compared with ELM and FLN, the SA-ELM shows better regression accuracy and generalization performance on six regression applications (Auto MPG, Bank domains, Machine CPU, Abalone, Delta elevators, Boston housing). In addition, the SD of SA-ELM on five regression applications (Auto MPG, Machine CPU, Abalone, Delta elevators, Boston housing) are superior to ELM and FLN. In a word, the SA-ELM has shown better performance on most regression applications in training and testing experiments, so it can be used as an effective machine learning tool. Compared with ELM and FLN, the SA-ELM has high regression accuracy and generalization performance. However, the SA-ELM spends much more running time to build data model than ELM and FLN, it is not fit the on-line computation. In this paper, all training and testing experiments belong to off-line model identification, so this algorithm mainly pursue the regression accuracy and stability of the building model.

Table 4 Comparison of the mean and its standard deviation of RMSE for ELM, FLN and SA-ELM using testing samples

Data sets	ELM		FLN		SA-ELM	
	RMSE		RMSE		RMSE	
	Mean	SD	Mean	SD	Mean	SD
Auto MPG	2.20×10^{-1}	7.4×10^{-3}	1.75×10^{-1}	4.20×10^{-3}	1.52×10^{-1}	2.92×10^{-3}
Servo	2.47×10^{-1}	2.61×10^{-2}	2.34×10^{-1}	3.87×10^{-2}	2.49×10^{-1}	5.26×10^{-2}
Bank domains	1.79×10^{-1}	7.80×10^{-3}	9.59×10^{-2}	6.33×10^{-4}	9.02×10^{-2}	1.71×10^{-3}
Machine CPU	2.00×10^{-1}	7.84×10^{-2}	2.70×10^{-1}	1.125×10^{-1}	1.07×10^{-1}	2.73×10^{-3}
Abalone	3.07×10^{-1}	3.97×10^{-2}	1.69×10^{-1}	1.2×10^{-3}	1.81×10^{-1}	7.85×10^{-4}
Forest fires	1.58×10^{-1}	3.10×10^{-3}	1.08×10^{-1}	2.80×10^{-3}	2.01×10^{-1}	3.85×10^{-3}
Delta elevators	2.77×10^{-1}	9.82×10^{-4}	1.25×10^{-1}	2.81×10^{-4}	1.06×10^{-1}	2.01×10^{-4}
Boston housing	2.45×10^{-1}	1.32×10^{-2}	2.09×10^{-1}	1.01×10^{-2}	1.93×10^{-1}	3.89×10^{-3}

Table 5 Comparison of the mean and its standard deviation of RMSE for ABC-ELM and SA-ELM using training samples

Data Sets	ABC-ELM			SA-ELM		
	RMSE			RMSE		
	Mean	SD	Runtime (s)	Mean	SD	Runtime (s)
Auto MPG	1.48×10^{-1}	4.6×10^{-3}	16.7010	1.25×10^{-1}	2.38×10^{-3}	15.0203
Servo	1.76×10^{-1}	1.04×10^{-2}	10.1724	1.29×10^{-1}	2.48×10^{-3}	8.9234
Bank domains	1.05×10^{-1}	3.2×10^{-3}	151.2239	9.02×10^{-2}	1.71×10^{-3}	194.0933
Machine CPU	6.29×10^{-2}	5.2×10^{-3}	11.423	3.35×10^{-2}	6.40×10^{-4}	10.3858
Abalone	1.54×10^{-1}	1.2×10^{-3}	74.7463	1.55×10^{-1}	2.94×10^{-4}	69.5860
Forest fires	2.08×10^{-1}	1.5×10^{-3}	19.2668	1.30×10^{-1}	2.47×10^{-3}	17.8540
Boston housing	2.08×10^{-1}	1.04×10^{-2}	77.0926	1.59×10^{-1}	3.05×10^{-3}	17.8288

As shown in Table 5, for the training samples of data sets, there are seven regression applications tested by ABC-ELM and SA-ELM. The SA-ELM shows better performance on all the applications about the mean of RMSE, that is the SA-ELM has better regression accuracy than ABC-ELM. In addition, for the SD, SA-ELM shows better steady on most applications than ABC-ELM. The running time of SA-ELM is smaller than ABC-ELM on six applications (Auto MPG, Servo, Machine CPU, Abalone, Forest fires, Boston housing). As shown in Table 6, for the testing samples of data sets, the RMSE and SD of SA-ELM are reasonable. The experiment results show that the SA-ELM possesses better regression precision and generalization ability than ABC-ELM on most applications.

5 Real-World Design Problem

Nowadays, the heat and electricity we used are mainly generated via power plants, consuming large amount of coal. However, the coal resource becomes scarce day by day, the target of

Table 6 Comparison of the mean and its standard deviation of RMSE for ABC-ELM and SA-ELM using testing samples

Data Sets	ABC-ELM			SA-ELM		
	RMSE			RMSE		
	Mean	SD	Runtime (s)	Mean	SD	Runtime (s)
Auto MPG	1.49×10^{-1}	6.7×10^{-3}	16.7010	1.52×10^{-1}	2.92×10^{-3}	15.0203
Servo	2.12×10^{-1}	1.38×10^{-2}	10.1724	2.49×10^{-1}	5.26×10^{-2}	8.9234
Bank domains	1.06×10^{-1}	3.1×10^{-3}	151.2239	9.02×10^{-2}	1.71×10^{-3}	194.0933
Machine CPU	2.03×10^{-1}	5.97×10^{-2}	11.423	1.07×10^{-1}	2.73×10^{-3}	10.3858
Abalone	1.63×10^{-1}	1.3×10^{-3}	74.7463	1.81×10^{-1}	7.85×10^{-4}	69.5860
Forest fires	6.34×10^{-1}	1.9×10^{-3}	19.2668	2.01×10^{-1}	3.85×10^{-3}	17.8540
Boston housing	2.25×10^{-1}	1.57×10^{-2}	77.0926	1.93×10^{-1}	3.89×10^{-3}	17.8288

Note that: the runtime is the total time for training and testing

energy saving and emission reduction is so urgent. Aiming at the coal consuming in fire plants, if the workers of power plant want to make the boiler operates in best case, they need understand the combustion characteristic of boiler and find the best operation parameters. So they should build the combustion model of boiler firstly. Due to the complexity, uncertainty, non-stability and nonlinearity of combustion process, it is difficult to set up an accurate mathematical model of a boiler's combustion process by the theory of thermodynamics [29,30]. ELM is a special artificial neural network, which is good at solving nonlinearity and complexity problem efficiently. Therefore, this paper adopts the SA-ELM to model the pulverized coal furnace thermal efficiency.

There are 600 data samples are collected from a 300 MW coal-fired boiler and shown in Table 7, which are divided into two parts: 400 training samples and 200 testing samples. These samples contain 16 operational conditions: the boiler load (MW), one variable; coal feeder feeding rate (t/h), five variables; the primary air velocity (t/h), five variables; the secondary air velocity (t/h), one variables; oxygen concentration in the flue gas (%), one variable; the carbon content of fly ash (%), two variables; thermal efficiency, one variables. These samples contain the combustion properties of coal boiler. The SA-ELM is applied to build the mathematical model of the boiler thermal efficiency based on the historical data of boiler, which illuminates the mapping relation between the boiler efficiency and 15 operational conditions of the boiler. And the complex mapping relation can be simplified as a model, which is shown in Fig. 3. As shown in Fig. 3, there are 15 input values and 1 output.

In this section, the hidden nodes are set 30 for ELM, FLN, and the SA-ELM. And the activation function is shown as Eq. (28). The maximum iteration of SA-ELM is set as 200, and the population size is 40. Each experiment for modeling the thermal efficiency repeats 30 times.

In order to verify the validity of SA-ELM model, it is compared with original ELM and the FLN on building the coal-fired boiler thermal efficiency model. In addition, three performance criteria (MAPE, MAE, RMSE) are employed to state the model superiority and the prediction performance. The smaller the values of MAPE and RMSE are, the better the prediction performance of a model [23]. The comparison results of the training samples and the testing samples are given in Tables 8 and 9, separately. As shown in Table 8 and Fig. 4, the results of RMSE of the SA-ELM are better than ELM and FLN, but the MAPE of SA-ELM

Table 7 Operational conditions of a 300 MW boiler

No.	Load	Coal feeder feeding rate					Secondary air	Oxygen content
		A	B	C	D	E		
1	160.8085	−0.06155	44.4767	42.5422	−0.01759	30.9350	402.9749	4.79091
2	160.2589	−0.06155	44.4767	42.1025	−0.01759	31.4186	408.3533	4.77809
3	160.2589	−0.06155	46.7630	41.6629	−0.01759	30.9350	412.9899	4.52161
4	160.2589	−0.06155	44.4767	43.9052	−0.01759	31.4186	402.6017	4.56649
5	160.8085	−0.06155	43.5534	42.5422	−0.01759	30.0556	403.0992	4.59855
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
201	222.3619	−0.06155	43.1577	40.7396	37.1343	37.7059	528.3385	4.49917
202	222.3619	−0.06155	43.1138	41.2232	36.4748	36.7826	528.3116	3.98623
203	222.3619	−0.06155	40.8715	40.7396	35.3317	36.34294	535.9919	3.96378
204	221.8123	−0.06155	43.1138	41.2232	35.3317	36.34294	531.1865	3.96378
205	221.2628	−0.06155	42.2344	40.7396	34.89204	35.8593	530.4282	3.97340
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
413	300.9526	45.13629	50.3683	−0.0175	49.35711	48.4338	700.6372	2.67821
414	300.4030	45.57596	49.0053	−0.0175	50.72008	47.55447	696.7144	2.70386
415	302.0517	44.82852	49.9286	−0.0175	51.64338	48.4338	703.0914	2.72310
416	302.0517	45.13629	49.9286	−0.0175	49.35711	48.4338	702.6057	2.72951
417	303.1509	46.4552	47.6424	−0.0175	50.2804	49.3571	708.7287	2.72310
No.	Carbon content of fly ash		The primary air					Thermal efficiency
	A	B	A	B	C	D	E	
1	0.3	1.76	0	78.66995	107.97	0	83.50362	0.892466
2	0.3	1.76	0	76.05958	109.6666	0	89.15303	0.891925

Table 7 continued

No.	Carbon content of fly ash		The primary air					Thermal efficiency
	A	B	A	B	C	D	E	
3	0.3	1.76	0	74.46180	103.5111	0	86.19042	0.894624
4	0.3	1.76	0	73.316704	108.0006	0	89.50769	0.901254
5	0.3	1.76	0	74.976799	103.4074	0	95.79134	0.903737
…	…	…	…	…	…	…	…	…
201	0	0.71	0	116.77588	115.7692	92.09789	107.6606	0.932501
202	0	0.71	0	122.18026	117.5323	98.13989	107.849	0.932041
203	0	0.71	0	116.98778	118.6345	95.2541	113.6684	0.931793
204	0	0.71	0	117.13087	114.5216	97.80649	111.2614	0.931543
205	0	0.71	0	118.21737	116.6968	94.20261	107.2822	0.903737
…	…	…	…	…	…	…	…	…
413	0.3	1.98	150	95.48228	0	89.05338	101.2684	0.969718
414	0.3	1.98	150	96.25814	0	82.50047	100.9544	0.970677
415	0.3	1.98	150	96.21373	0	89.76531	101.7423	0.97097
416	0.3	1.98	150	93.42113	0	81.58241	100.9364	0.970294
417	0.3	1.98	150	100.92089	0	89.87859	100.7846	0.969863

Note: some data are negative in A, C, D of coal feeder mean no work, and that is the testing error for equipment

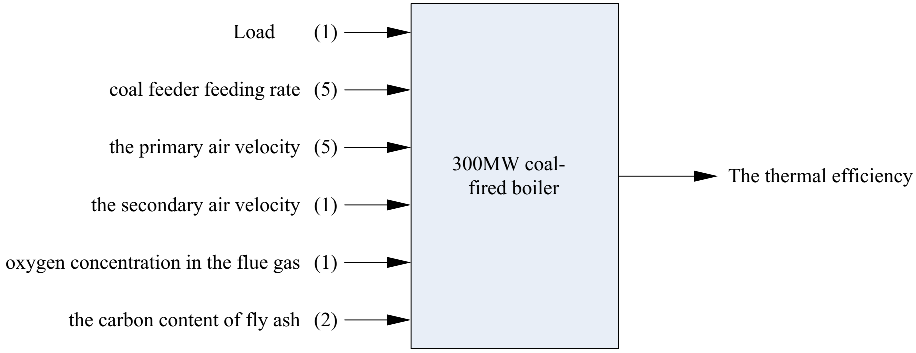

Fig. 3 Simplified boiler model

Table 8 Training samples performance values for the thermal efficiency

Method	RMSE				MAE	MAPE
	Min	Max	Std	Mean		
ELM	6.77×10^{-2}	7.55×10^{-2}	1.90×10^{-3}	7.04×10^{-2}	0.0022	0.0024
FLN	6.30×10^{-2}	6.71×10^{-2}	1.10×10^{-3}	6.52×10^{-2}	0.0020	0.0021
SA-ELM	0	6.74×10^{-2}	1.22×10^{-3}	6.44×10^{-2}	0.0022	0.0024

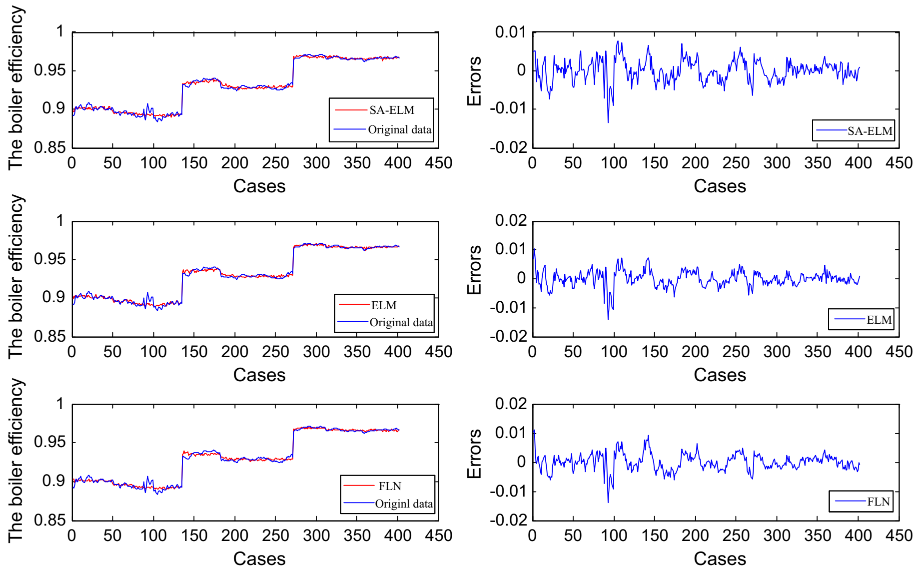

Fig. 4 Compare outputs and errors with the original system for training samples

is less than the FLN. That is to say, the SA-ELM model owns good identification ability. As shown in Table 9 and Fig. 5, the MAE, MAPE and RMSE of the SA-ELM are better than ELM and FLN. For the criteria: MAPE, the best model is SA-ELM, the next the FLN and

Table 9 Testing samples performance values for the thermal efficiency

Method	RMSE				MAE	MAPE
	Min	Max	Std	Mean		
ELM	1.53×10^{-1}	2.38×10^{-1}	2.21×10^{-2}	1.94×10^{-1}	0.0061	0.0067
FLN	8.81×10^{-2}	1.11×10^{-1}	4.90×10^{-3}	9.94×10^{-2}	0.0051	0.0056
SA-ELM	0	1.17×10^{-1}	1.94×10^{-3}	9.77×10^{-2}	0.0044	0.0049

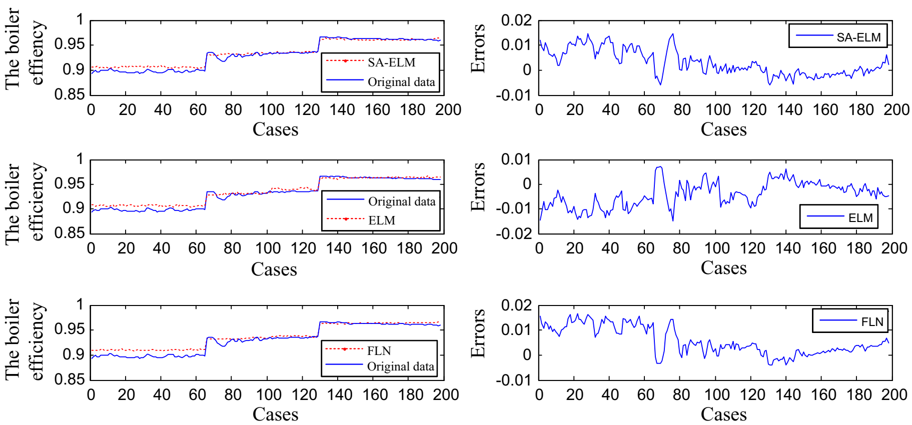

Fig. 5 Compare outputs and errors with the original system for testing samples

ELM is the worst. Thus, it can be seen that the SA-ELM shows better nonlinear identification and generalization ability.

In summary, the SA-ELM algorithm takes advantage over the others on identification ability and generalization performance. It can achieve favorable training accuracy and prediction performance under various operating conditions. So the thermal efficiency model of the pulverized coal furnace built by the SA-ELM model has presented its reference value for engineering application.

6 Complexity Analyses

So far, four methods have been listed and discussed to test the classical regression applications. The SA-ELM shows better performance than others at most applications based on the ideology of the ATLBO to optimize the input-weights and bias of hidden layer for ELM. The process would need iteration many times and that would also lead to spending much more time than ELM. In Sect. 4, as shown in Table 5, the SA-ELM reaches the same training or testing accuracy as ELM after spending thousands of running times. Therefore, the improved algorithm is extraordinary complexity relatively for ELM and FLN.

However, the SA-ELM has higher training accuracy and stronger generalization ability than ELM and FLN after repetitional iteration. When just pursuing the regression accuracy and stability of a model, the speed-ability would not be essential while all the experiments are off-line. Comparing with ABC-ELM, the SA-ELM shows better performance, such as high regression and stability. For model structure, the SA-ELM is more complex than ELM and FLN. The input-weights and threshold of ELM generate randomly without any changes. However, for SA-ELM, these parameters update via Eqs. (22) and (25) step by step before the stop criterion reached. This process spends much running time as well. In conclusion, the SA-ELM fits for off-line identifications.

7 Conclusions

In order to obtain better input-weights and bias of hidden layer of the ELM, inspired with the idea of the teaching-learning-based optimization algorithm, this paper proposes a kind of SA-ELM. The input-weights and the bias of hidden nodes of the SA-ELM are updated in “teaching phase” and “learning phase”. As a result, the optimal structure parameters of the SA-ELM can be found. According to the application of SA-ELM on eight benchmark functions, the SA-ELM has shown its better recognition ability and generalization ability. In addition, the thermal efficiency model of the pulverized coal furnace is built by the SA-ELM, which presents its practical engineering application value.

Acknowledgments Project supported by the National Natural Science Foundation of China (Grant No. 61573306, 61403331).

References

1. Salakhutdinov R, Larochelle H (2010) Efficient learning of deep Boltzmann machines. In: International conference on artificial intelligence and statistics, pp 693–700
2. Vapnik V (2000) The nature of statistical learning theory. Springer, Berlin
3. Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feedforward neural network. In: Proceedings of IEEE international joint conference on neural network, vol 2, pp 985–990
4. Wang SC (2003) Artificial neural network. Interdisciplinary computing in Java programming. Springer, Heidelberg
5. Li L, Liu D, Ouyang J (2012) A new regularization classification method based on extreme learning machine in network data. J Inf Comput Sci 9(12):3351–3363
6. Karpagachelvi S, Arthanari M, Sivakumar M (2012) Classification of electrocardiogram signals with support vector machines and extreme learning machine. Neural Comput Appl 21(6):1331–1339
7. Yu J, Rui Y, Tang YY et al (2014) High-order distance-based multiview stochastic learning in image classification. IEEE Trans Cybern 44(12):2431–2442
8. Yu J, Hong R, Wang M et al (2014) Image clustering based on sparse patch alignment framework. Pattern Recognit 47(11):3512–3519
9. Lan Y, Soh YC, Huang GB (2010) Two-stage extreme learning machine for regression. Neurocomputing 73(16–18):3028–3038
10. Tang X, Han M (2009) Partial Lanczos extreme learning machine for singal-output regression problems. Neurocomputing 72(13–15):3066–3076
11. Mikolov T, Chen K, Corrado G et al (2013) Efficient estimation of word representations in vector space. arXiv preprint. arXiv:1301.3781
12. Zong W, Huang GB (2011) Face recognition based on extreme learning machine. Neurocomputing 74(16):2541–2551
13. Pan C, Park DS, Yang Y et al (2012) Leukocyte image segmentation by visual attention and extreme learning machine. Neural Comput Appl 21(6):1217–1227
14. Yu J, Hong R, Wang M et al (2014) Image clustering based on sparse patch alignment framework. Pattern Recognit 47(11):3512–3519
15. Yu J, Rui Y, Tao D (2014) Click prediction for web image reranking using multimodal sparse coding. IEEE Trans Image Process 23(5):2019–2032

16. Wu S, Wang Y, Cheng S (2013) Extreme learning machine based wind speed estimation and sensorless control for wind turbine power generation system. Neurocomputing 102:163–175
17. Luo Y, Tang J, Yan J et al (2014) Pre-trained multi-view word embedding using two-side neural network. In: Twenty-eighth AAAI conference on artificial intelligence
18. Yu J, Tao D (2013) Modern machine learning techniques and their applications in cartoon animation research. Wiley, Hoboken
19. Cao JW, Lin ZP, Huang GB (2012) Self-adaptive evolutionary extreme learning machine. Neural Process Lett 36:285–305
20. Zhu QY, Qin AK, Suganthan PN, Huang GB (2005) Evolutionary extreme learning machine. Pattern Recognit 38:1759–1763
21. Matias T, Souza F, Araujo R, Antunes CH (2014) Learning of a single-hidden layer feedforward neural network using an optimized extreme learning machine. Neurocomputing 129:428–436
22. Han F, Yao HF, Ling QH (2013) An improved evolutionary extreme learning machine based on particle swarm optimization. Neurocomputing 116:87–93
23. Li GQ, Niu PF, Ma YP, Wang HB, Zhang WP (2014) Tuning extreme learning machine by an improved artificial bee colony to model and optimize the boiler efficiency. Knowl Based Syst 67:278–289
24. Li GQ, Niu PF, Liu C, Zhang WP (2012) Enhanced combination modeling method for combustion efficiency in coal-fired boilers. Appl Soft Comput 12:3132–3140
25. Li GQ, Niu PF, Zhang WP, Liu YC (2013) Model NOx emissions by least squares support vector machine with tuning based on ameliorated teaching-learning-based optimization. Chemom Intell Lab Syst 126:11–20
26. Li GQ, Niu PF, Duan XL, Zhang XY (2013) Fast learning network: a novel artificial neural network with a fast learning speed. Neural Comput Appl. doi:10.1007/s00521-013-1398-7
27. Liang NY, Huang gb, Saratchandran P, Sundararajan N (2006) A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans Neural Netw 17:1411–1423
28. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315
29. Zhou H, Zhao JP, Zheng LG, Wang CL, Cen KF (2012) Modeling NOx emissions from coal-fired utility boilers using support vector regression with ant colony optimization. Eng Appl Artif Intell 25:147–158
30. Song Z, Kusiak A (2007) Constraint-based control of boiler efficiency: a data-mining approach. IEEE Trans Ind Inform 3(1):73–83