Neural Process Lett
DOI 10.1007/s11063-017-9611-9

An Adaptive Extreme Learning Machine for Modeling NOx Emission of a 300 MW Circulating Fluidized Bed Boiler

Xia Li[1,2] · Peifeng Niu[1] · Guoqiang Li[1] · Jianping Liu[2]

Abstract Extreme learning machine (ELM) provides high learning speed, but generalization performance needs to be further improved. Therefore, we propose an adaptive ELM with a relaxation factor λ (A-ELM). In A-ELM, according to the nonlinear degree of actual data, the output layer obtains adaptively $1 - \lambda$ rate information through the hidden layer and λ rate information through the input layer. Since the relaxation factor λ is bound up with the input weights and hidden biases of A-ELM, in order to obtain the optimal λ, λ, input weights and hidden biases are obtained together by teaching–learning-based optimization (A-ELM-TLBO). Then, 15 benchmark regression data sets verify the performance of A-ELM-TLBO. Finally, A-ELM-TLBO is applied to set up the mapping relation between NOx emission and operational conditions of a 300 MW circulating fluidized bed (CFB) boiler. Compared with six other models, experimental results show that A-ELM-TLBO has good approximation ability and generalization performance. So, A-ELM-TLBO provides a good basis for tuning CFB boiler operating parameters to reduce NOx emission.

Keywords Extreme learning machine · Relaxation factor · Teaching–learning-based optimization · NOx emission · Circulating fluidized bed boiler

1 Introduction

With the development of coal-fired power stations, NOx pollutant emission becomes a more serious problem. How to decrease NOx emission has been the hottest study focus. In order to decrease pollutant emission, NOx emission model firstly needs to be established. The model will be an important design tool to help the engineer to optimize the operating conditions, reduce pollution emission, enhance the design of new boilers and evaluate the retrofit of the old

✉ Peifeng Niu
niupeifeng2014@163.com

1 School of Electrical Engineering, Yanshan University, Qinhuangdao 066004, China

2 Hebei Normal University of Science and Technology, Qinhuangdao 066004, China

Published online: 31 March 2017

boilers [1]. The NOx emission model depends on various operating parameters, such as coal feeder rate, air velocity, oxygen content in the flue gas, exhaust gas temperature. However, due to the complexity, uncertainty, strong coupling and the nonlinearity of the combustion process, traditional modeling of industrial-scale processes using the theory of thermodynamics is difficult. Support vector regression [2–4] and artificial neural network(ANN) [1,5–7] are usually intelligent models. The ANN approach is very convenient and direct and can achieve good prediction effects under various operating conditions.

As an important branch of neural network, extreme learning machine (ELM) was introduced in [8] by Huang et al. It overcomes the problems of traditional feed-forward neural networks, such as the presence of local minima and slow rate of convergence. It has been used to solve many problems. Zong et al. [9] studied the performance of ELM in multi-label face recognition applications. Cao et al. [10] used an ensemble based ELM and sparse representation to handle classification problems. ELM and adaptive sparse representation for image classification was studied [11]. ELM was used to solve regression problems [12–14]. ELM and its improvements [15–20] are all single hidden feed-forward neural network. Since they do not have a direct connection between the input layer and output layer, the generalization performance may be affected [21,22]. Fast learning network (FLN)[22] based on ELM increased direct the connection between the input layer and output layer. FLN with much more compact networks could achieve very good generalization performance.

However, for different conditions, FLN does not consider the contribution rate of the input layer and hidden layer. Therefore, we propose an adaptive ELM with a relaxation factor $\lambda \in [0, 1]$, named A-ELM. In order to improve the generalization performance, A-ELM not only receives adaptively $1 - \lambda$ rate information from the hidden layer but also λ rate information from the input layer. In particular, when $\lambda = 0$, the method changes into ELM; when $\lambda = 0.5$, the method obtains the same rate information through the hidden layer and input layer and hence the method changes into FLN in that sense; when $\lambda = 1$, the method changes into a linear neural network (LNN). Therefore, ELM, FLN and LNN are three special cases of A-ELM. In other words, A-ELM extends ELM, FLN and LNN.

In practical application, we need to obtain the optimal relaxation factor λ in the interval [0,1] and thus the key problem is to design an effective optimization algorithm. Teaching–learning-based optimization (TLBO) algorithm was proposed by Rao and Patel [23,24]. The algorithm simulates teaching–learning phenomenon of a classroom to solve multi-dimensional linear and nonlinear problems with high efficiency. It has been applied and proved to be effective in solving many engineering optimization problems [25–29]. Therefore, TLBO algorithm is adopted to obtain the optimal relaxation factor λ of A-ELM (A-ELM-TLBO). Then, 15 benchmark regression data sets verify the performance of proposed method. The computational results show that A-ELM-TLBO has better approximation ability and generalization performance than other methods. Then, A-ELM-TLBO is selected to set up the mapping relation between NOx emissions and operational conditions of a 300 MW circulating fluidized bed (CFB) boiler.

The rest of the paper is organized as follows. A brief review of ELM is given in Sect. 2. A-ELM is proposed in Sect. 3. A hybrid of TLBO and A-ELM to obtain the optimal λ is introduced in Sect. 4. A-ELM-TLBO is applied to model NOx emission of a 300 MW CFB boiler in Sect. 5. The time complexity analysis of the proposed method is discussed in Sect. 6. Finally, the paper is concluded in Sect. 7.

2 Extreme Learning Machine

The outline of ELM is explained in this section. ELM is different from the traditional neural network. In ELM, input weights and hidden layer biases are randomly generated and output weights are obtained by solving a system of linear equations analytically based on the least square.

Suppose that there is a data set with arbitrary distinct samples $(\boldsymbol{x}_r, \boldsymbol{y}_r), r = 1, 2, \ldots, N$, where $\boldsymbol{x}_r = [x_{r1}, x_{r2}, \ldots, x_{rn}]^{\mathrm{T}} \in R^{\mathrm{n}}$ and $\boldsymbol{y}_r = [y_{r1}, y_{r2}, \ldots, y_{rl}]^{\mathrm{T}} \in R^{\mathrm{l}}$. The mathematical model of ELM with m hidden nodes and activation function $g(\cdot)$ for the given data can be formulated as

$$\boldsymbol{y}_r = \sum_{j=1}^{m} \boldsymbol{\beta}_j g(\boldsymbol{\omega}_j \cdot \boldsymbol{x}_r + \boldsymbol{b}_j), \quad r = 1, 2, \ldots, N, \tag{1}$$

where $\boldsymbol{\omega}_j = [\omega_{j1}, \omega_{j2}, \ldots, \omega_{jn}]^{\mathrm{T}}$ denotes the weight vector that connects the input nodes to the jth hidden node. $\boldsymbol{\beta}_j = [\beta_{1j}, \beta_{2j}, \ldots, \beta_{lj}]^{\mathrm{T}}$ is the weight vector that connects the output nodes with the jth hidden node. $\boldsymbol{b}_j$ is the threshold of the jth hidden node. $\boldsymbol{\omega}_j \cdot \boldsymbol{x}_r$ denotes the inner product of $\boldsymbol{\omega}_j$ and $\boldsymbol{x}_r$ in Eq. (1). Equation (1) can be expressed as

$$\boldsymbol{H}\boldsymbol{\beta} = \boldsymbol{Y}. \tag{2}$$

$\boldsymbol{H}$ is the hidden layer output matrix, where

$$\boldsymbol{H} = \begin{bmatrix} g(\boldsymbol{\omega}_1 \cdot \boldsymbol{x}_1 + \boldsymbol{b}_1) & \ldots & g(\boldsymbol{\omega}_m \cdot \boldsymbol{x}_1 + \boldsymbol{b}_m) \\ \vdots & \vdots & \vdots \\ g(\boldsymbol{\omega}_1 \cdot \boldsymbol{x}_N + \boldsymbol{b}_1) & \ldots & g(\boldsymbol{\omega}_m \cdot \boldsymbol{x}_N + \boldsymbol{b}_m) \end{bmatrix},$$

$\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_m]^{\mathrm{T}}$ and $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N]^{\mathrm{T}}$.

The input weight $\boldsymbol{\omega}_j = [\omega_{j1}, \omega_{j2}, \ldots, \omega_{jn}]^{\mathrm{T}}$ and hidden bias $\boldsymbol{b}_j$, $j = 1, 2, \ldots, m$ are randomly generated. The output weight matrix $\hat{\boldsymbol{\beta}}$ is obtained by solving a system of linear equations

$$\hat{\boldsymbol{\beta}} = \boldsymbol{H}^{+}\boldsymbol{Y}, \tag{3}$$

where $\boldsymbol{H}^{+}$ is M–P generalized inverse of matrix $\boldsymbol{H}$.

A general algorithm for ELM can be stated as follows. For a given training set, an infinitely differentiable activation function $g(\cdot)$ and a hidden neuron number m:

(1) Generate random input weight $\boldsymbol{\omega}_j$ and bias $\boldsymbol{b}_j$, $j = 1, 2, \ldots, m$;
(2) Calculate the hidden layer output matrix $\boldsymbol{H}$;
(3) Calculate the output weight $\hat{\boldsymbol{\beta}} : \hat{\boldsymbol{\beta}} = \boldsymbol{H}^{+}\boldsymbol{Y}$.

3 An Adaptive ELM with a Relaxation Factor $\lambda \in [0, 1]$ (A-ELM)

ELM is a single hidden feed-forward neural network. It has no direct connection between the input layer and output layer and so the generalization performance may be affected. In order to increase the connection between the input layer and output layer, Li et al. put forward FLN. FLN increases the connection between the input layer and output layer, but the network does not consider the contribution rate of the input layer and hidden layer according to different data sets. Therefore, in this study, an adaptive ELM with a relaxation factor $\lambda \in [0, 1]$ is proposed, which is named as A-ELM.

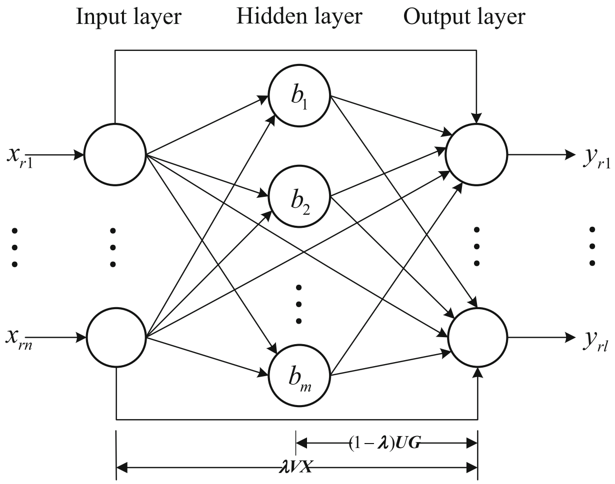

Fig. 1 Structure of A-ELM

In order to improve the generalization performance, output layer of A-ELM not only adaptively receives $1 - \lambda$ rate information from the hidden layer but also receives λ rate information from the input layer for different data sets. Therefore, A-ELM is a parallel mathematical mode of linear and nonlinear structure with a relaxation factor $\lambda \in [0, 1]$. It can handle both nonlinear problems and linear problems. A-ELM, shown in Fig. 1, is described as follows:

Suppose that there is a data set with N arbitrary distinct samples $(\boldsymbol{x}_r, \boldsymbol{y}_r) \in R^{n\times N} \times R^{l\times N}$, $r = 1, 2, \ldots, N$ and that A-ELM has m hidden layer nodes. $\boldsymbol{W} = (\omega_{ji})_{m\times n}$ is the input weight matrix between the hidden layer and input layer. $\boldsymbol{b}_j$, $j = 1, 2, \ldots, m$ is the threshold of the jth hidden node. $\boldsymbol{W}$ and $\boldsymbol{b}_j$ are randomly generated. $\boldsymbol{U} = (u_{kj})_{l\times m}$ is the output weight matrix between the output layer and hidden layer. $\boldsymbol{V} = (v_{ki})_{l\times n}$ is the output weight matrix between the input layer and output layer and λ is the relaxation factor. A-ELM is mathematically modeled as

$$\begin{cases} y_{r1} = \lambda \sum_{i=1}^{n} v_{1i} x_{ri} + (1-\lambda) \sum_{j=1}^{m} u_{1j} g\left(b_j + \sum_{t=1}^{n} \omega_{jt} x_{rt}\right) \\ y_{r2} = \lambda \sum_{i=1}^{n} v_{2i} x_{ri} + (1-\lambda) \sum_{j=1}^{m} u_{2j} g\left(b_j + \sum_{t=1}^{n} \omega_{jt} x_{rt}\right) \\ \ldots \\ y_{rl} = \lambda \sum_{i=1}^{n} v_{li} x_{ri} + (1-\lambda) \sum_{j=1}^{m} u_{lj} g\left(b_j + \sum_{t=1}^{n} \omega_{jt} x_{rt}\right) \end{cases} \tag{4}$$

$r = 1, 2, \ldots, N$. Equation (4) can be transformed into

$$\boldsymbol{y}_r = \lambda \boldsymbol{V} \boldsymbol{x}_r + (1-\lambda) \sum_{j=1}^{m} \boldsymbol{u}_j g(b_j + \boldsymbol{\omega}_j \boldsymbol{x}_r). \tag{5}$$

where $\boldsymbol{u}_j = [u_{1j}, \ldots, u_{lj}]^{\mathrm{T}}$, $\boldsymbol{\omega}_j = [\omega_{j1}, \ldots, \omega_{jn}]^{\mathrm{T}}$ and $\boldsymbol{x}_r = [x_{r1}, \ldots, x_{rn}]^{\mathrm{T}}$. Then, Eq. (5) can be rewritten compactly as Eq. (6), i.e.,

$$\begin{aligned} \boldsymbol{Y}_{l\times N} &= \lambda \boldsymbol{V}_{l\times n}\boldsymbol{X}_{n\times N} + (1-\lambda)\boldsymbol{U}_{l\times m}\boldsymbol{G}_{m\times N} \\ &= [\boldsymbol{V}_{l\times n}, \boldsymbol{U}_{l\times m}]\begin{bmatrix} \lambda \boldsymbol{X} \\ (1-\lambda)\boldsymbol{G} \end{bmatrix}_{(n+m)\cdot N}, \end{aligned} \tag{6}$$

where

$$\boldsymbol{G}_{m\times N} = \begin{bmatrix} g(\boldsymbol{\omega}_1 \cdot \boldsymbol{x}_1 + \boldsymbol{b}_1) & \ldots & g(\boldsymbol{\omega}_1 \cdot \boldsymbol{x}_N + \boldsymbol{b}_1) \\ \vdots & \vdots & \vdots \\ g(\boldsymbol{\omega}_m \cdot \boldsymbol{x}_1 + \boldsymbol{b}_m) & \ldots & g(\boldsymbol{\omega}_m \cdot \boldsymbol{x}_N + \boldsymbol{b}_m) \end{bmatrix}.$$

Equation (6) can be transformed into Eq. (7)

$$\boldsymbol{H}_1\boldsymbol{\beta}_1 = \boldsymbol{Y}_1, \tag{7}$$

where

$$\boldsymbol{H}_1 = [\lambda \boldsymbol{X}^{\mathrm{T}}, (1-\lambda)\boldsymbol{G}^{\mathrm{T}}], \quad \boldsymbol{\beta}_1 = \begin{bmatrix} \boldsymbol{V}^{\mathrm{T}} \\ \boldsymbol{U}^{\mathrm{T}} \end{bmatrix}, \boldsymbol{Y}_1 = \boldsymbol{Y}^{\mathrm{T}}.$$

The input weight vector $\boldsymbol{\omega}_j$ and the hidden bias $\boldsymbol{b}_j$, $j = 1, 2, \ldots, m$ are randomly generated, so A-ELM is simply equivalent to finding a least squares solution $\hat{\boldsymbol{\beta}}_1$ of the linear systems $\boldsymbol{H}_1\boldsymbol{\beta}_1 = \boldsymbol{Y}_1$ by Eq. (8), which is similar to ELM.

$$\hat{\boldsymbol{\beta}}_1 = \boldsymbol{H}_1^{+}\boldsymbol{Y}_1, \tag{8}$$

where $\boldsymbol{H}_1^{+} = [\lambda \boldsymbol{X}^{\mathrm{T}}, (1-\lambda)\boldsymbol{G}^{\mathrm{T}}]^{+}$ is the M–P generalized inverse of matrix $\boldsymbol{H}_1$. Therefore,

$$\begin{cases} \boldsymbol{V} = \hat{\boldsymbol{\beta}}_1(1:l, 1:n) \\ \boldsymbol{U} = \hat{\boldsymbol{\beta}}_1(1:l, n+1:n+m), \end{cases} \tag{9}$$

When $rank(\boldsymbol{H}_1) = N$, Eq. (8) can be rewritten as

$$\begin{aligned} \hat{\boldsymbol{\beta}}_1 &= \boldsymbol{H}_1^{+}\boldsymbol{Y}_1 = \boldsymbol{H}_1^{\mathrm{T}}(\boldsymbol{H}_1\boldsymbol{H}_1^{\mathrm{T}})^{-1}\boldsymbol{Y}_1 \\ &= \begin{bmatrix} \lambda \boldsymbol{X} \\ (1-\lambda)\boldsymbol{G} \end{bmatrix}\left([\lambda \boldsymbol{X}^{\mathrm{T}}, (1-\lambda)\boldsymbol{G}^{\mathrm{T}}]\begin{bmatrix} \lambda \boldsymbol{X} \\ (1-\lambda)\boldsymbol{G} \end{bmatrix}\right)^{-1}\boldsymbol{Y}_1 \\ &= \begin{bmatrix} \lambda \boldsymbol{X} \\ (1-\lambda)\boldsymbol{G} \end{bmatrix}[\lambda^2\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X} + (1-\lambda^2)\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G}]^{-1}\boldsymbol{Y}_1. \end{aligned} \tag{10}$$

When $rank(\boldsymbol{H}_1) = n + m$, Eq. (8) can be rewritten as

$$\begin{aligned} \hat{\boldsymbol{\beta}}_1 &= \boldsymbol{H}_1^{+}\boldsymbol{Y}_1 = (\boldsymbol{H}_1^{\mathrm{T}}\boldsymbol{H}_1)^{-1}\boldsymbol{H}_1^{\mathrm{T}}\boldsymbol{Y}_1 \\ &= \left(\begin{bmatrix} \lambda \boldsymbol{X} \\ (1-\lambda)\boldsymbol{G} \end{bmatrix}[\lambda \boldsymbol{X}^{\mathrm{T}}, (1-\lambda)\boldsymbol{G}^{\mathrm{T}}]\right)^{-1}\begin{bmatrix} \lambda \boldsymbol{X} \\ (1-\lambda)\boldsymbol{G} \end{bmatrix}\boldsymbol{Y}_1 \\ &= \begin{bmatrix} \lambda^2\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}} & \lambda(1-\lambda)\boldsymbol{X}\boldsymbol{G}^{\mathrm{T}} \\ \lambda(1-\lambda)\boldsymbol{G}\boldsymbol{X}^{\mathrm{T}} & (1-\lambda)^2\boldsymbol{G}\boldsymbol{G}^{\mathrm{T}} \end{bmatrix}^{-1}\begin{bmatrix} \lambda \boldsymbol{X} \\ (1-\lambda)\boldsymbol{G} \end{bmatrix}\boldsymbol{Y}_1. \end{aligned} \tag{11}$$

As seen from the above learning process, A-ELM is the parallel structure of a single layer feed-forward neural network and a single hidden layer feed-forward network with a relaxation factor λ. The output layer not only obtains $1-\lambda$ rate information through the hidden layer but also obtains λ rate information through the input layer. In addition, several studies have shown that a single layer feed-forward neural network has high efficiency in solving linear problems and that a single hidden layer feed-forward network can realize complex nonlinear map. Thus, according to the nonlinear degree of actual data, we can find the optimal relaxation factor $\lambda \in [0, 1]$. In particular, when $\lambda = 0$, the network changes into ELM; when $\lambda = 0.5$, the network obtains the same rate information through the hidden layer and input layer, and thus the network changes into FLN in this sense; when $\lambda = 1$, the network changes into a LNN. Therefore, ELM, FLN and LNN are three special cases of A-ELM. In other words, A-ELM extends ELM, FLN and LNN.

4 Hybrid of A-ELM and TLBO to Obtain the Optimal Relaxation Factor, Input Weights and Hidden Biases (A-ELM-TLBO)

In order to improve the generalization performance of A-ELM, we need to obtain the optimal relaxation factor $\lambda \in [0, 1]$. λ is bound up with input weights and hidden biases and thus λ, input weights and hidden biases are optimized together. The key problem is to design the effective algorithm. TLBO algorithm has the advantages of simple operation, fewer parameters, strong global convergence ability and insensitivity to initial point. Therefore, TLBO is suitable for solving a large number of linear and nonlinear optimization problems. The details of the algorithm are introduced as follows.

4.1 An Introduction of TLBO

TLBO is based on the effect of the influence of a teacher on learners in a class. Like other nature-inspired algorithms, TLBO is also a population-based method that uses a population of solutions to proceed to the global solution, but the method has no user-defined parameter. A group of learners is considered as the population $\boldsymbol{P}$. Every learner is considered as an individual $\boldsymbol{P}_s$, $s = 1 : N_p$, where N_p is the population size. In TLBO algorithm, different subjects offered to learners are considered as different design variables. The learning result of a learner is analogous to the 'fitness' (function value $f(\boldsymbol{P}_s)$, $s = 1 : N_p$) as in other optimization algorithms. The teacher is considered as the most knowledgeable person in a class who shares his/her knowledge with the students to improve the marks of a class. The quality of the learners is evaluated by the mean value of the student's marks in a class. There are two parts in TLBO: 'teacher phase' and 'learner phase'. The teacher phase means learning from the teacher and the learner phase means learning through the interaction between learners.

4.1.1 The Teacher Phase

During the teacher phase, the teacher $\boldsymbol{P}_{teacher}$ is assigned to the best individual, whose 'fitness' $f(\boldsymbol{P}_{teacher})$ is best in a class. A teacher tries to enhance the mean value $\boldsymbol{P}_{mean}$ of a class up to his/her level. However, practically, it can be done to some extent according to the learning capability of the class. Suppose $\boldsymbol{P}_s$ and $\boldsymbol{P}_{new,s}$, $s = 1 : N_p$, respectively, denote the previous marks of every learner and his/her new marks through learning from a teacher. The teacher phase is formulated as

$$\boldsymbol{P}_{new,s} = \boldsymbol{P}_s + r_s(\boldsymbol{P}_{teacher} - T_F \boldsymbol{P}_{mean}), \tag{12}$$

where $r_s \in [0, 1]$ is a random number and T_F is a teaching factor. T_F is either 1 or 2. It can be designed as follows

$$T_F = round[1 + rand(0, 1)]. \tag{13}$$

Accept $\boldsymbol{P}_{new,s}$, if it gives a better function value. (When solving minimization problems, we accept $\boldsymbol{P}_{new,s}$, if $f(\boldsymbol{P}_{new,s}) < f(\boldsymbol{P}_s)$. The reverse is true for maximization problems.)

4.1.2 The Learner Phase

Learners increase their knowledge by two different means: one through input from the teacher and the other through the interaction between themselves. A learner interacts randomly with other learners with the help of group discussions, presentations, formal communications, etc. A learner $\boldsymbol{P}_s$ learns something new by Eq. (14), if the other learner $\boldsymbol{P}_h$ has more knowledge than him or her. Otherwise, $\boldsymbol{P}_s$ is moved away from $\boldsymbol{P}_h$ by Eq. (15). The learner phase is expressed as

For $s = 1 : N_P$ randomly selects another learner $\boldsymbol{P}_h$, such that $s \neq h$

if $f(\boldsymbol{P}_s) < f(\boldsymbol{P}_h)$

$$\boldsymbol{P}_{new,s} = \boldsymbol{P}_s + r_s(\boldsymbol{P}_s - \boldsymbol{P}_h), \tag{14}$$

else

$$\boldsymbol{P}_{new,s} = \boldsymbol{P}_s + r_s(\boldsymbol{P}_h - \boldsymbol{P}_s). \tag{15}$$

end if

End

where $r_s \in [0, 1]$ is random number. Accept $\boldsymbol{P}_{new,s}$, if it gives a better function value. The algorithm will continue its iterations until reaching the maximum number of generations. (The above equations are for minimization problems.)

4.2 Learning Process of A-ELM Based on TLBO

The detailed hybrid of TLBO and A-ELM is summarized in the following steps.

Step 1: Initialization. We randomly generate the population $\boldsymbol{P}$. Each individual $\boldsymbol{P}_s$, $s = 1 : N_p$ of the population is composed of a set of input weights, hidden biases and relaxation factor λ as follows:

$$\boldsymbol{P}_s = [\omega_{11}, \omega_{12}, \ldots, \omega_{1n}, \omega_{21}, \omega_{22}, \ldots, \omega_{2n}, \ldots, \omega_{m1}, \omega_{m2}, \ldots, \omega_{mn}, b_1, b_2, \ldots, b_m, \lambda] \tag{16}$$

where ω_{ji} and b_i are randomly generated within the range of $[-1, 1]$. λ is randomly generated within the range of [0, 1].

Step 2: Evaluating the fitness of each individual. For each individual $\boldsymbol{P}_s$, the corresponding output weights are analytically computed by Eq. (8). The fitness f of each individual $\boldsymbol{P}_s$ is evaluated. In order to avoid over-fitting, the fitness f of each individual is evaluated based on Eq. (17), which is adopted as the root mean squared error (RMSE) on the validation set only, instead of the whole training set as used in [19,38], i.e.,

$$f = \sqrt{\frac{\sum_{q=1}^{N_1} \| \lambda \boldsymbol{V}\boldsymbol{x}_q + (1-\lambda)\sum_{j=1}^{m} \boldsymbol{u}_j g(b_j + \boldsymbol{\omega}_j \boldsymbol{x}_q) - \boldsymbol{y}_q \|_2^2}{l \times N_1}}, \tag{17}$$

where $\boldsymbol{N}_1$ is the number of the validation samples.

Step 3: The teacher phase. Calculate the mean of the population. The teacher will try to shift the mean $\boldsymbol{P}_{mean}$ toward $\boldsymbol{P}_{teacher}$. The difference between two values is expressed as

$$\boldsymbol{Differencemean}_s = r_s[\boldsymbol{P}_{teacher} - (T_F \cdot \boldsymbol{P}_{mean})]. \tag{18}$$

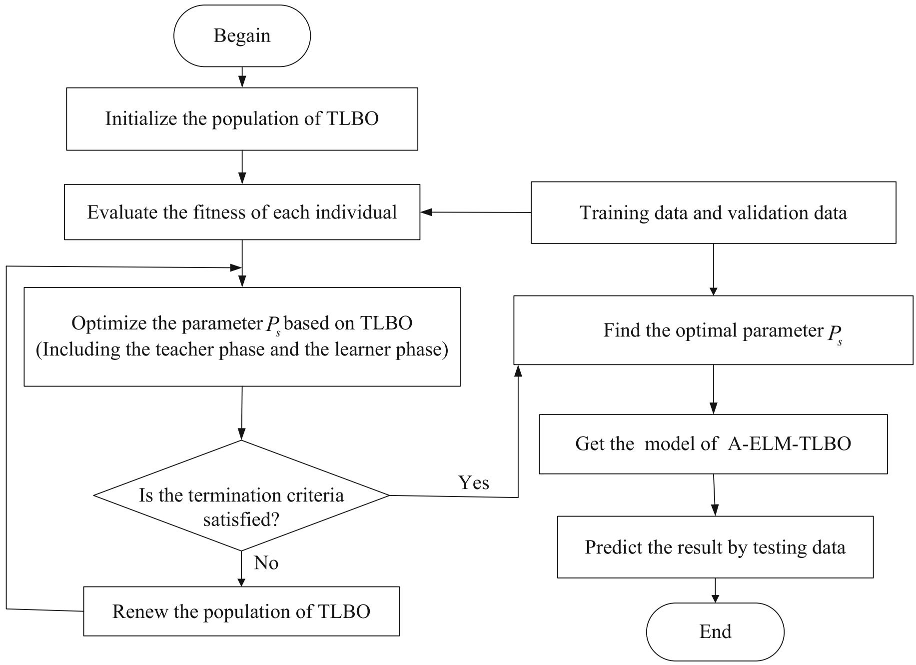

Fig. 2 Flowchart of the proposed hybrid model

where T_F is calculated by Eq. (13). The resulting difference is added to the current solution as a new value, i.e.,

$$\boldsymbol{P}_{new,s} = \boldsymbol{P}_s + \boldsymbol{Differencemean}_s. \tag{19}$$

Accept $\boldsymbol{P}_{new,s}$, if it gives a better fitness than that of $\boldsymbol{P}_s$.

Step 4: The learner phase. As explained above, learners increase their knowledge with the help of their mutual interactions. The mathematical expression is explained in Sect. 4.1.2. Obtain $\boldsymbol{P}_{new,s}$ after the learner phase. Accept $\boldsymbol{P}_{new,s}$, if it gives a better fitness than that of $\boldsymbol{P}_s$.

Step 5: If the maximum number of iterations M is reached, TLBO algorithm is stopped; otherwise, the iteration is repeated from Step 3.

Finally, the optimal λ, input weights and hidden biases are obtained and then A-ELM is applied to the testing data.

For a more vivid observation, the realization flowchart of the proposed hybrid method (A-ELM-TLBO) is shown in Fig. 2.

4.3 Computer Simulation and Analysis of Results

In order to verify the performance of A-ELM-TLBO, we add a comparison on many kinds of regressions with real benchmark data sets. A-ELM-TLBO is applied to 15 benchmark regression data sets verify, where Condition based maintenance of naval propulsion plants, Combined cycle power plant, Yacht_hydrodynamics and Concrete compressive strength are abbreviated as CBM, CCPP, Yacht and CCS, repectively. Every data set is randomly divided into the training set, testing set and validation set. The data sets with their corresponding partitions are shown in Table 1.

Table 1 Specification of real-word benchmark regression problems

Data set	Attribute	Number of observations		
		Training	Testing	Validation
Auto.MPG [30]	7	200	96	96
Triazines [31]	60	100	43	43
Servo [30]	4	80	44	43
Stocks domain [32]	9	450	250	250
Delta-ailerons [32]	6	3000	2065	2064
Abalone [30]	8	2177	1000	1000
Machine CPU [32]	6	100	55	54
Boston housing [30]	13	200	153	153
Cancer [30]	32	100	47	47
Price [30]	15	80	40	39
CBM [33]	16	5000	3467	3467
Airfoil self-noise [30]	5	1000	252	251
CCPP [34]	4	5000	2284	2284
Yacht [30]	6	100	54	54
CCS [35]	8	500	265	265

In order to compare TLBO performance with a standard method, krill herd (KH) algorithm [36] (KH without any genetic operators) is also used to find the appropriate relaxation factor of A-ELM(A-ELM-KH). KH algorithm is also a novel swarm intelligence approach for optimization tasks, which is inspired from the krill herding motions. The crucial advantage of KH algorithm is its strong diversity, simple operation and less adjustment parameters.

A-ELM-TLBO is also compared with ELM, FLN, LNN, RR-ELM [37] (RR-ELM replaced the least square method with ridge regression to calculate output weights of the hidden matrix.) and E-ELM [38] (E-ELM uses differential evolutionary (DE) algorithm to select the input weights and hidden biases). In TLBO, KH and DE, the population size is 10 and the maximum iteration number M is 20 by trial and error. Other parameter settings in E-ELM are the same as described by [38]. 30 hidden neurons are set in every model. Every experiment is repeated 20 times. All the programs are run under MATLAB 2009a environment in a Dua-Core CPU, 2.2 GHz CPU, 1.96GB RAM. The activation function is sigmoid function $g(x) = \frac{1}{1+e^{-x}}$. In all experiments, the inputs and outputs of data sets have been normalized into $[-1, 1]$ and $[0, 1]$, respectively.

Mean root mean square error (RMSE), standard deviation (SD) of RMSE and mean running time of every set for the seven methods are given in Table 2. For every data set, the best mean RMSE of the training set and the testing set are shown in bold face.

As seen from Table 2, according to RMSE of the training sets, A-ELM-TLBO, A-ELM-KH, ELM, FLN, E-ELM and RR-ELM have better approximation capabilities than LNN in almost all data. Since the structure of LNN is simple, the computational time is shortest in all methods and the testing RMSE is worst in almost all data sets. A-ELM-TLBO has better regression accuracy on 9 applications (Stock, Delta-ailerons, Boston housing, CBM^b, CBM^c, Airfoil self-noise, CCPP, Yacht and CCS), A-ELM-KH with 5 applications (Auto.MPG, Servo, Price, Cancer and Machine) and FLN with 3 applications (Triazines, Abalone and Machine). So, A-ELM-TLBO has good approximation ability.

Table 2 Computational results of the seven methods

Data set	Algorithm	Training		Testing		Time[a] (s)
		RMSE	SD	RMSE	SD	
Auto.MPG	A-ELM-TLBO	0.0578	0.0012	**0.0824**	0.0018	3.2417
	A-ELM-KH	**0.0567**	0.0018	0.0842	0.0030	2.4234
	ELM	0.0603	0.0013	0.0845	0.0044	0.0055
	FLN	0.0574	0.0019	0.0838	0.0053	0.0054
	LNN	0.1429	0	0.1544	0	0.0032
	E-ELM	0.0603	0.0014	0.0835	0.0032	1.8557
	RR-ELM	0.0616	0.0020	0.0830	0.0030	0.0062
Triazines	A-ELM-TLBO	0.1459	0.0051	**0.2048**	0.0178	4.7222
	A-ELM-KH	0.0793	0.0094	0.9812	0.4252	3.1177
	ELM	0.1433	0.0051	0.2134	0.0274	0.0032
	FLN	**0.0766**	0.0079	1.6648	1.0363	0.0078
	LNN	0.1279	0	0.2454	0	0.0024
	E-ELM	0.1448	0.0057	0.2135	0.0175	2.4118
	RR-ELM	0.1464	0.0041	0.2162	0.0224	0.0055
Servo	A-ELM-TLBO	0.0672	0.0082	**0.1036**	0.0137	2.7721
	A-ELM-KH	**0.0615**	0.0061	0.1149	0.0153	2.2581
	ELM	0.0740	0.0052	0.1351	0.0279	0.0031
	FLN	0.0658	0.0054	0.1427	0.0301	0.0031
	LNN	0.1982	0	0.2209	0	0.0047
	E-ELM	0.0754	0.0062	0.1261	0.0168	1.7667
	RR-ELM	0.1267	0.0061	0.1234	0.0104	0.0031
Stocks domain	A-ELM-TLBO	**0.0346**	0.0015	**0.0398**	0.0020	3.6700
	A-ELM-KH	0.0375	0.0017	0.0426	0.0021	2.7979
	ELM	0.0460	0.0043	0.0512	0.0054	0.0039
	FLN	0.0403	0.0036	0.0467	0.0040	0.0047
	LNN	0.1837	0	0.2090	0	0.0039
	E-ELM	0.0453	0.0043	0.0491	0.0034	2.0936
	RR-ELM	0.0463	0.0053	0.0520	0.0065	0.0039
Delta-ailerons	A-ELM-TLBO	**0.0379**	0.0001	**0.0393**	0.0002	8.0251
	A-ELM-KH	0.0381	0.0001	0.0395	0.0002	5.5481
	ELM	0.0383	0.0001	0.0397	0.0003	0.0117
	FLN	0.0382	0.0002	0.0396	0.0003	0.0148
	LNN	0.2040	0	0.2025	0	0.0024
	E-ELM	0.0383	0.0002	0.0398	0.0002	4.1525
	RR-ELM	0.0384	0.0002	0.0397	0.0002	0.0171
Abalone	A-ELM-TLBO	0.0723	0.0004	**0.0766**	0.0008	6.8851
	A-ELM-KH	0.0722	0.0004	0.0768	0.0007	4.9062
	ELM	0.0730	0.0004	0.0769	0.0007	0.0102
	FLN	**0.0721**	0.0004	0.0767	0.0009	0.0141
	LNN	0.0792	0	0.0814	0	0.0063
	E-ELM	0.0729	0.0004	0.0770	0.0008	3.4776

Table 2 continued

Data set	Algorithm	Training		Testing		Time[a] (s)
		RMSE	SD	RMSE	SD	
	RR-ELM	0.0730	0.0004	0.0768	0.0006	0.0117
Machine	A-ELM-TLBO	0.0189	0.0009	0.0870	0.0415	2.8581
	A-ELM-KH	**0.0183**	0.0008	0.0984	0.0592	2.2542
	ELM	0.0210	0.0021	0.0793	0.0330	0.0031
	FLN	**0.0183**	0.0009	0.1382	0.1115	0.0039
	LNN	0.1132	0	0.0903	0	0.0015
	E-ELM	0.0220	0.0021	0.0765	0.0336	1.8511
	RR-ELM	0.0314	0.0016	**0.0445**	0.0071	0.0024
Boston housing	A-ELM-TLBO	**0.0578**	0.0027	**0.1051**	0.0062	3.3594
	A-ELM-KH	0.0585	0.0030	0.1162	0.0075	2.7113
	ELM	0.0733	0.0050	0.1301	0.0057	0.0031
	FLN	0.0600	0.0038	0.1191	0.0087	0.0063
	LNN	0.0876	0	0.1306	0	0.0046
	E-ELM	0.0723	0.0038	0.1292	0.0093	1.8892
	RR-ELM	0.0721	0.0035	0.1316	0.0092	0.0070
Cancer	A-ELM-TLBO	0.2185	0.0047	0.3024	0.0263	4.1341
	A-ELM-KH	**0.1640**	0.0076	0.5084	0.0675	2.7908
	ELM	0.2123	0.0047	0.3108	0.0279	0.0031
	FLN	0.1681	0.0113	0.5726	0.1611	0.0046
	LNN	0.2210	0	0.3030	0	0.0023
	E-ELM	0.2089	0.0066	0.3171	0.0273	2.0388
	RR-ELM	0.2145	0.0074	**0.2924**	0.0271	0.0039
Price	A-ELM-TLBO	0.0526	0.0087	0.1166	0.0274	3.4480
	A-ELM-KH	**0.0363**	0.0032	0.1231	0.0311	2.4694
	ELM	0.0539	0.0042	0.1094	0.0168	0.0025
	FLN	0.0463	0.0042	0.1225	0.0218	0.0050
	LNN	0.1008	0	0.1424	0	0.0020
	E-ELM	0.0549	0.0052	0.1184	0.0164	1.9900
	RR-ELM	0.0560	0.0040	**0.1060**	0.0141	0.0050
CBM[b]	A-ELM-TLBO	**0.0158**	0.0056	**0.0156**	0.0054	15.8551
	A-ELM-KH	0.0483	0.0087	0.0482	0.0086	10.5199
	ELM	0.2878	0.0086	0.2887	0.0094	0.0187
	FLN	0.0975	0.0270	0.0967	0.0269	0.0280
	LNN	0.1175	0	0.1166	0	0.0063
	E-ELM	0.2859	0.0103	0.2867	0.0109	7.1862
	RR-ELM	–	–	–	–	–

Table 2 continued

Data set	Algorithm	Training		Testing		Time[a] (s)
		RMSE	SD	RMSE	SD	
CBM[c]	A-ELM-TLBO	**0.0262**	0.0072	**0.0259**	0.0070	16.7739
	A-ELM-KH	0.0543	0.0098	0.0540	0.0097	10.5455
	ELM	0.2963	0.0037	0.3006	0.0034	0.0219
	FLN	0.0820	0.0080	0.0819	0.0081	0.0296
	LNN	0.0900	0	0.0897	0	0.0078
	E-ELM	0.2946	0.0053	0.2987	0.0054	7.1947
	RR-ELM	–	–	–	–	–
Airfoil self-noise	A-ELM-TLBO	**0.0963**	0.0022	**0.0939**	0.0026	4.2409
	A-ELM-KH	0.0999	0.0022	0.0980	0.0026	2.9391
	ELM	0.1069	0.0031	0.1037	0.0039	0.0046
	FLN	0.1032	0.0025	0.0988	0.0028	0.0046
	LNN	0.1393	0	0.1364	0	0.0024
	E-ELM	0.1062	0.0029	0.1040	0.0034	2.4680
	RR-ELM	0.1060	0.0024	0.1026	0.0036	0.0039
CCPP	A-ELM-TLBO	**0.0535**	0.0001	**0.0557**	0.0001	10.6291
	A-ELM-KH	0.0536	0.0001	0.0559	0.0001	7.2891
	ELM	0.0540	0.0001	0.0565	0.0001	0.0187
	FLN	0.0538	0.0001	0.0562	0.0002	0.0218
	LNN	0.3592	0	0.3573	0	0.0031
	E-ELM	0.0539	0.0001	0.0564	0.0002	5.7019
	RR-ELM	0.0540	0.0002	0.0564	0.0002	0.0141
Yacht	A-ELM-TLBO	**0.0232**	0.0036	**0.0388**	0.0056	3.1356
	A-ELM-KH	0.0416	0.0077	0.0720	0.0110	2.3962
	ELM	0.0819	0.0143	0.1251	0.0233	0.0024
	FLN	0.0667	0.0144	0.1100	0.0262	0.0031
	LNN	0.1627	0	0.1614	0	0.0023
	E-ELM	0.0840	0.0120	0.1223	0.0209	1.9282
	RR-ELM	0.1188	0.0120	0.1428	0.0121	0.0031
CCS	A-ELM-TLBO	**0.0880**	0.0033	**0.0995**	0.0050	3.7206
	A-ELM-KH	0.0942	0.0036	0.1058	0.0057	2.8088
	ELM	0.1091	0.0050	0.1168	0.0076	0.0039
	FLN	0.0973	0.0037	0.1073	0.0042	0.0055
	LNN	0.1813	0.0000	0.1494	0	0.0024
	E-ELM	0.1059	0.0056	0.1142	0.0054	2.2098
	RR-ELM	0.1067	0.0053	0.1124	0.0046	0.0039

Bold value denotes the best computational results

[a] The time is the total running time for training and testing

[b] The output is GT compressor decay state coefficient

[c] The output is GT turbine decay state coefficient

The analysis of testing data shows that A-ELM-TLBO presents the lowest mean RMSE on 13 applications (Auto.MPG, Triazines, Servo, Abalone, Stock, Delta-ailerons, Boston housing, CBM[b], CBM[c], Airfoil self-noise, CCPP, Yacht and CCS). RR-ELM presents the lowest mean testing RMSE on 3 applications (Price, Cancer and Machine). A-ELM-KH does not present the lowest mean RMSE on any application. In short, A-ELM-TLBO has better generalization ability.

Taking account of running time, ELM, FLN and RR-ELM show the superiority in running time, while E-ELM, A-ELM-TLBO and A-ELM-KH have bad running time performance, which owes to the global optimization framework to find the optimal network parameters. From Table 2, we conclude that A-ELM-TLBO needs more running time than A-ELM-KH, which attributes to the teacher phase and the learner phase used in TLBO. Therefore, A-ELM-TLBO fits off-line identifications.

In conlude, the comparison results show that the generalization performance of A-ELM-TLBO is better than other methods on many kinds of applications, which demonstrates that A-ELM-TLBO has a wide application and can solve a variety of complex application fields. Therefore, A-ELM-TLBO is selected to model NOx emission of the boiler.

5 Model the NOx Emission of a CFB Boiler

The growing energy crisis is coming and the environment is very important for people, so energy saving and environmental protection are the key subject of coal-fired power plant industry. NOx is one of pollution sources for the boiler combustion. Therefore, combustion technology of low NOx emission has become an important research direction of boiler engineering. If we want a boiler achieves the optimum operation conditions for combustion, we must understand the boiler combustion characteristics and build the NOx emission predicted model of a boiler. However, due to the complexity, uncertainty, strong coupling and the nonlinearity of the combustion process, traditional modeling of industrial-scale processes using the theory of thermodynamics is difficult. In order to obtain good approximations and generalization performances, this paper uses the proposed A-ELM-TLBO to build NOx emission for a 300 MW CFB boiler. It provides a good basis for tuning CFB boiler operating parameters to reduce NOx emission and simultaneously verifies the effectiveness of the A-ELM-TLBO.

There are 240 data samples (80 for 50% load, 80 for 75% load and 80 for 100% load) collected from the CFB boiler, which are sampled once every 30 seconds and are listed in Table 3. In this study, the total 240 cases are divided into three parts: 144 cases (48 for 50% load, 48 for 75% load, and 48 for 100% load) as the training samples, 48 cases (16 for 50% load, 16 for 75% load and 16 for 100% load) as the validation samples and the remaining 48 cases as the testing samples. The NOx emission mainly depends on 20 operational conditions as input variables for A-ELM-TLBO, which are given as follows:

The boiler load (%);

The coal feeder rate (CFR, t/h), including A, B, C, D levels;

The primary air velocity (PAV, kN m^3/h), including left and right levels;

The secondary air velocity (SAV, kN m^3/h), including left and inside, left and outside, right and inside, right and inside levels;

The primary air temperature (PAT, °C), including left and right levels;

The secondary air temperature (SAT, °C), including left and right levels;

The oxygen content in the flue gas (OC, %);

The exhaust gas temperature (EGT, °C);

Table 3 The boiler operating condition

Case	Load (%)	FBT (°C)	CFR (t/h)				PAV (kN m^3/h)		PAT (°C)	
			A	B	C	D	Left	Right	Left	Right
1	54.008	877.622	28.466	28.756	28.534	28.474	186.985	136.176	249.287	247.397
2	54.008	877.622	28.466	28.756	28.534	28.474	237.336	176.915	249.287	247.397
3	54.117	877.962	28.932	28.892	28.95	28.831	183.552	238.48	248.786	247.397
4	53.973	878.422	29.179	29.392	29.22	29.202	219.027	275.099	248.786	246.881
...	...	...	...	...	...	...	...	...	...	...
81	71.845	868.626	52.009	44.378	20.031	54.715	222.46	201.404	262.958	260.565
82	71.845	868.626	52.009	44.378	20.031	54.715	249.237	331.401	262.958	260.565
83	71.837	867.801	51.894	44.157	19.935	54.396	260.909	259.078	262.958	260.565
84	71.806	867.912	51.759	44.026	20.013	53.687	212.389	255.417	262.958	260.565
...	...	...	...	...	...	...	...	...	...	...
161	101.28	867.59	61.435	55.491	55.379	61.098	414.709	311.718	281.339	278.605
162	101.28	867.59	61.435	55.491	55.379	61.198	377.174	288.145	281.339	278.605
163	101.066	865.975	61.435	55.394	55.671	61.35	388.389	268.462	281.339	278.605
164	100.616	864.498	61.082	55.085	55.29	61.178	343.531	333.232	281.339	278.605
...	...	...	...	...	...	...	...	...	...	...
240	101.382	875.358	59.004	53.164	53.136	59.168	242.371	318.813	283.886	281.819

Case	SAV (kN m^3/h)				SAT (°C)		EPFM (A)		OC (%)	EGT (°C)	NOx
	A	B	C	D	Left	Right	A	B			
1	44.126	41.367	41.164	55.661	258.233	250.204	88.419	102.685	8.647	147.803	155.02
2	44.126	41.238	40.975	55.156	258.233	250.204	88.648	102.304	8.647	147.803	155.02

Table 3 continued

Case	SAV (kN m^3/h)				SAT (°C)		EPFM (A)		OC (%)	EGT (°C)	NOx
	A	B	C	D	Left	Right	A	B			
3	45.103	40.938	41.704	53.785	258.233	250.204	88.763	101.922	8.647	147.803	154.257
4	46.71	40.31	43.925	54.364	257.686	249.674	88.877	100.587	8.647	147.803	157.309
...	...	...	...	...	...	...	...	...	...	...	...
81	72.022	77.484	76.963	57.798	272.063	261.516	86.245	106.004	5.96	148.741	155.783
82	71.34	78.664	76.748	59.223	272.063	261.516	85.825	106.652	5.96	148.741	155.783
83	74.389	82.61	77.033	57.479	272.063	261.516	86.321	105.012	5.96	148.741	156.469
84	71.15	78.514	78.764	57.081	272.063	261.516	85.864	104.02	5.96	148.741	156.469
...	...	...	...	...	...	...	...	...	...	...	...
161	101.937	104.988	108.895	87.504	292.292	276.819	122.94	136.291	4.734	160.848	176.381
162	101.481	106.241	108.735	88.814	292.292	276.819	125.763	134.384	4.734	160.848	176.381
163	101.089	93.541	108.998	85.024	292.292	276.819	126.03	137.321	4.734	160.848	181.492
164	101.131	99.22	107.843	87.112	292.292	276.819	125.763	134.04	4.734	160.848	183.705
...	...	...	...	...	...	...	...	...	...	...	...
240	99.793	110.157	110.716	88.392	294.575	279.506	104.135	0.153	3.883	165.677	131.523

Table 4 The performance comparison of testing data for the NOx emission

Model	RMSE	MAE	MAPE (%)
ELM	8.4038	6.3777	4.8139
FLN	5.8898	4.9520	3.6623
LNN	15.731	12.682	9.4137
E-ELM	6.3690	5.1915	3.6758
RR-ELM	5.6688	4.6525	3.4160
A-ELM-KH	4.0117	3.2913	2.2803
A-ELM-TLBO	**2.7089**	**2.2720**	**1.5793**

Bold value denotes the best computational results

Table 5 The performance comparison of training data for the NOx emission

Model	RMSE	MAE	MAPE (%)	Time (s)
ELM	6.8694	5.1905	3.8350	0.2490
FLN	3.9637	3.0774	2.2234	0.2500
LNN	11.754	9.4770	6.7318	0.2180
E-ELM	6.1671	4.6605	3.3109	18.3770
RR-ELM	5.9087	4.6326	3.3204	0.4220
A-ELM-KH	3.6887	3.0454	2.1750	25.1780
A-ELM-TLBO	**3.2219**	**2.5933**	**1.8466**	44.3820

Bold value denotes the best computational results

The fluid bed temperature (FBT, °C);

The electricity of powder feeding machine (EPFM, A), including A and B levels.

NOx emission is molded as outputs variables. In this section, the hidden nodes are set 30 for all models. After many experiments, the maximum iteration number of A-ELM-TLBO and E-ELM is set as 50. All other parameter settings are the same as that of the fourth part. In order to verify the superiority of A-ELM-TLBO model, it is compared with other 6 methods. In addition, RMSE, mean absolute percentage error(MAPE) and mean absolute error (MAE) are employed to state the approximation performance and generalization performance.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y - \hat{y}|, \tag{20}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}|\frac{y - \hat{y}}{y}| \times 100\%, \tag{21}$$

where N is the number of data, y is the actual output and $\hat{y}$ is the predicted output. The smaller the values of indexes are, the better the performance of a model is.

The comparison results of the training samples and the testing samples are given in Tables 5 and 4 and Figs. 3, 4, 5 and 6. As shown in Table 5, the RMSE of A-ELM-TLBO is 3.2219, the MAE is 2.5933 and the MAPE is 1.8466%. Three performance indexes of A-ELM-TLBO are much less than those of other models. Compared to ELM, the RMSE, MAE and MAPE decrease 53.10, 50.04, 51.85% respectively. Therefore, the approximation performance of

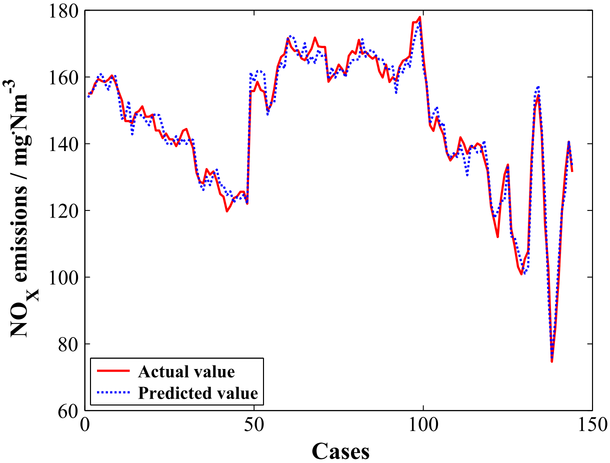

Fig. 3 The output of the proposed model for training data

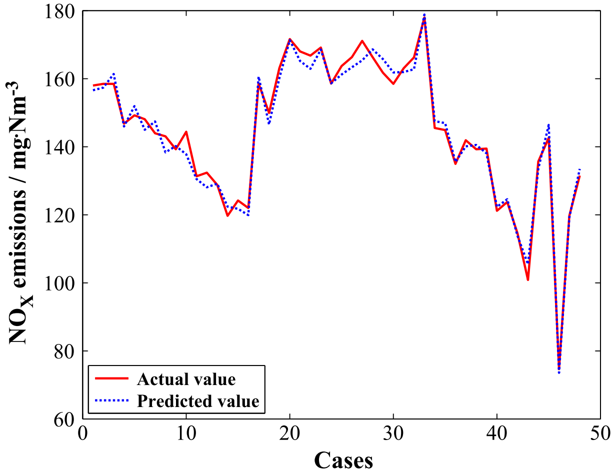

Fig. 4 The output of the proposed model for testing data

A-ELM-TLBO is best for the training data. Figure 3 shows that the output of the proposed model is compared with the original NOx emission for training set. Figure 5 shows the relative training errors of seven models. From the Fig. 5, we could know that the relative errors of A-ELM-TLBO are much smaller and more stable than those of others.

Figure 4 shows that the output of the proposed model is compared with the original NOx emission for testing set. For testing data, from the Table 4, we could know the RMSE of A-ELM-TLBO is 2.7089, the MAE is 2.2720 and the MAPE is 1.5793%. Every performance index is smallest in all models, so A-ELM-TLBO has the best generalization ability. The relative errors of testing data are given in Fig. 6. The relative errors of A-ELM-TLBO fluctuate near zero. The range of fluctuation of A-ELM-TLBO is smaller than others, so A-ELM-TLBO is very suitable for model the NOx emission of the CFB boiler.

In summarize, although A-ELM-TLBO takes much more running time than ELM, the proposed model is more accurate and it has better approximation ability and generalization

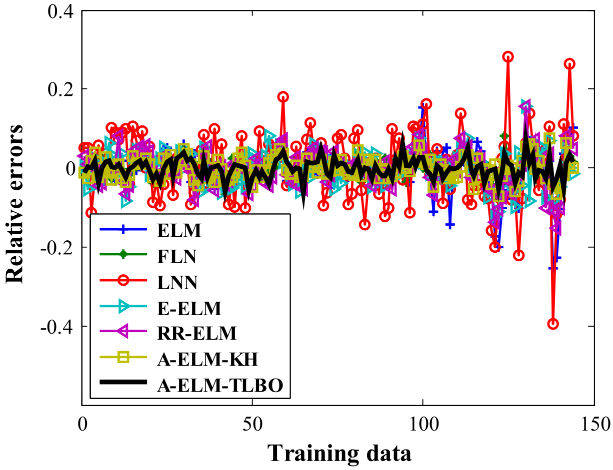

Fig. 5 The relative errors of seven models for training data

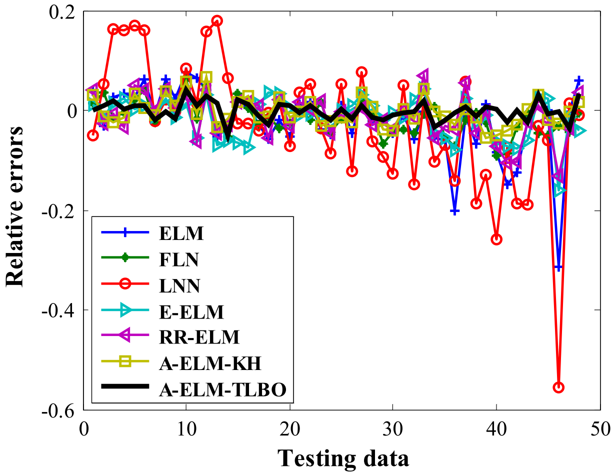

Fig. 6 The relative errors of seven models for testing data

performance. It provides an effective technique to predict the output of actual boiler working for off-line identification.

6 Time Complexity Analysis of A-ELM-TLBO

In A-ELM-TLBO, the training process mainly includes the following two parts: (1) evaluation the fitness function of A-ELM-TLBO and (2) updating of the network parameters. For the network parameter population N_p in each generation, the evaluation fitness function is similar to training ELM for $2N_p$ times on the training data, which will take $O(2N_p(Nm^2+m^3)) = O(N_p(Nm^2+m^3))$[39], where N is the number of the data set and m is the number of the hidden layer nodes. Since $N >> m$ for most cases, the time complexity of evaluation the fitness function is about $O(N_p N m^2)$. In the process of updating the network parameters, the time complexity of TLBO is also based on network parameter vectors, which will take $O(2N_p(nm+m+1)) = O(N_p(nm+m+1))$, where n is the number of the attributes of the data set. Assume that the maximum number of iterations is M, the time complexity of A-ELM-TLBO can be expressed as $O(MN_p N m^2) + O(MN_p(nm+m+1))$. Since $Nm^2 >> nm+m+1$ for most cases, the time complexity of A-ELM-TLBO is dominated by the first part, which is about $O(MN_p N m^2)$. Similarly, the time complexities of A-ELM-KH and E-ELM are about $O(MN_p N m^2)$, which are the same as A-ELM-TLBO. However, the performance of A-ELM-TLBO is better than others in term of learning capability. When just pursuing the approximation ability and generalization performance, the speed-ability would not be essential while all the experiments are off-line.

7 Conclusions

A novel learning network named A-ELM-TLBO is proposed in this study. In A-ELM-TLBO, the optimal input weights, hidden biases and relaxation factor are obtained by TLBO and the output weights are obtained by solving a linear system based on the least square. A-ELM-TLBO can adaptively receive $1-\lambda$ rate information from the hidden laye and λ rate information from the input layer. Then, A-ELM-TLBO model is used to evaluate NOx emission of a 300 MW circulating fluidized bed boiler. The experimental results demonstrate that the proposed A-ELM-TLBO is able to set up a good model for complicated nonlinear system with high accuracy. Compared with other six models, the proposed model shows the good performance. So it is suited to be applied to various complex application fields.

Acknowledgements The authors thank the anonymous reviewers for their very helpful and constructive comments and suggestions. This work is supported by the National Natural Science Foundations of China (No. 61573306 and No. 61403331), the Teaching Research Project of Hebei Normal University of Science and Technology (No. JYZD201413) and the Open Project of State Key Laboratory of Metastable Materials Science and Technology, Yanshan University (No. 201702).

References

1. Zhou H, Cen K, Fan J (2004) Modeling and optimization of the NOx emission characteristics of a tangentially firedboiler with artificial neural networks. Energy 29:167–183
2. Si F, Romero C, Yao Z et al (2009) Optimization of coalfired boiler SCRs based on modified support vector machine models and genetic algorithms. Fuel 88:806–816
3. Gu YP, Zhao WJ, Wu ZS (2011) Online adaptive least squares support vector machine and its application in utility boiler combustion optimization systems. J Process Control 21:1040–1048
4. Zhou H, Zheng LG, Kf Cen (2010) Computational intelligence approach for NOx emissions minimization in a coal-fired utility boiler. Energ Convers Manag 51:580–586
5. Ilamathi P, Selladurai V, Balamurugan K (2013) Modeling and optimization of unburned carbon in coal-fired boiler using artificial neural network and genetic algorithm. J Energy Res Technol 135(3):1–3
6. Suresh M, Reddy KS, Kolar AK (2011) ANN-GA based optimization of a high ash coal-fired supercritical power plant. Appl Energy 88:4867–4873
7. Song JG, Romero CE, Yao Z (2016) Improved artificial bee colony-based optimization of boiler combustion considering NOx emissions, heat rate and fly ash recycling for on-line applications. Fuel 172:20–28
8. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501
9. Zong W, Huang GB (2011) Face recognition based on extreme learning machine. Neurocomputing 74(16):2541–2551

10. Cao J, Hao J, Lai X et al (2016) Ensemble extreme learning machine and sparse representation classification. J Frankl Inst 353(17):4526–4541
11. Cao J, Zhang K, Luo M et al (2016) Extreme learning machine and adaptive sparse representation for image classification. Neural Netw 81:91–102
12. Liang NY, Huang GB, Saratchandran P et al (2006) A fast and accurate on-line sequential learning algorithm for feedforward networks. IEEE Trans Neural Netw 17(6):1411–14236
13. Yang Y, Wang Y, Yuan X (2012) Bidirectional extreme learning machine for regression problem and its learning effectiveness. IEEE Trans Neural Netw Learn Syst 23(9):1498–1505
14. Huang GB, Zhou H, Ding X et al (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern B 42(2):513–529
15. Huang GB, Chen L (2007) Convex incremental extreme learning machine. Neurocomputing 70(16):3056–3062
16. Huang GB, Chen L (2008) Enhanced random search based incremental extreme learning machine. Neurocomputing 71(16):3460–3468
17. Lan Y, Soh YC, Huang GB (2009) Ensemble of online sequential extreme learning machine. Neurocomputing 72(13):3391–3395
18. Cao J, Lin Z, Huang GB (2012) Self-adaptive evolutionary extreme learning machine. Neural Process Lett 36(3):285–305
19. Han F, Yao HF, Ling QH (2013) An improved evolutionary extreme learning machine based on particle swarm optimization. Neurocomputing 116:87–93
20. Fan YT, Wu W, Yang WY et al (2014) A pruning algorithm with $L_{1/2}$ regularizer for extreme learning machine. J Zhejiang Univ Sci C 15(2):119–125
21. He MY (1994) Double parallel feedforward neural network with application to simulation study of flight fault inspection. Acta Aeronaut Astronaut Sin 15(7):877–881
22. Li GQ, Niu PF, Duan XL (2014) Fast learning network: a novel artificial neural network with a fast learning speed. Neural Comput Appl 24(7–8):1683–1695
23. 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
24. Rao RV, Savsani VJ, Vakharia DP (2012) Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. Inf Sci 183(1):1–15
25. Amiri B (2012) Application of teaching–learning-based optimization algorithm on cluster analysis. J Basic Appl Sci Res 2(3):11795–11802
26. Rao RV, Kalyankar VD (2013) Parameter optimization of modern machining processes using teaching–learning-based optimization algorithm. Eng Appl Artif Intell 26(1):524–531
27. Rao RV (2015) Teaching learning based optimization and its engineering applications. Springer, London
28. Rao RV (2016) Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems. Decis Sci Lett 5(1):1–30
29. Singh M, Panigrahi BK, Abhyankar AR (2013) Optimal coordination of directional over-current relays using teaching learning-based optimization (TLBO) algorithm. Int J Electr Power Energy Syst 50:33–41
30. Lichman M (2013) UCI machine learning repository. http://archive.ics.uci.edu/ml
31. Ross DK, Jonathan DH, Michael JES (1995) Comparison of artificial intelligence methods for modeling pharmaceutical qsars. Appl Artif Intell 9(2):213–233
32. Torgo L Regression datasets. http://www.dcc.fc.up.pt/ltorgo/Regression/DataSets.html
33. Coraddu A, Oneto L, Ghio A et al (2016) Machine learning approaches for improving condition-based maintenance of naval propulsion plants. In: Proceedings of the institution of mechanical engineers part M journal of engineering for the maritime environment, pp 56–63
34. Tüfekci P (2014) Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. Int J Elec Power 60(11):126–140
35. Yeh IC (1998) Modeling of strength of high-performance concrete using artificial neural net works. Cement Concrete Res 28(12):1797–1808
36. Gandhi AH, Alack AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear SCI 17(12):4831–4845
37. Li GQ, Niu PF (2013) An enhanced extreme learning machine based on ridge regression for regression. Neural Comp Appl 22(3–4):803–810
38. Zhu QY, Qin AK, Suganthan PN et al (2005) Evolutionary extreme learning machine. Pattern Recogn 38(10):1759–1763
39. Zhao G, Shen Z, Man Z (2011) Robust input weight selection for well-conditioned extreme learning machine. Int J Inform Technol 17(1):1–13