Neural Comput & Applic
DOI 10.1007/s00521-016-2232-9

ORIGINAL ARTICLE

Extreme learning machine with fuzzy input and fuzzy output for fuzzy regression

Hai-tao Liu[1] · Jing Wang[2] · Yu-lin He[3] · Rana Aamir Raza Ashfaq[4]

Received: 16 March 2015 / Accepted: 16 February 2016

Abstract It is practically and theoretically significant to approximate and simulate a system with fuzzy inputs and fuzzy outputs. This paper proposes a extreme learning machine (ELM)-based fuzzy regression model (FR_{ELM}) in which both inputs and outputs are triangular fuzzy numbers. Algorithm for training FR_{ELM} is designed, and its computational complexity is analyzed. Furthermore, the convergence and error estimation for FR_{ELM} are discussed. Numerical simulations show that the proposed FR_{ELM} can effectively approximate a fuzzy input and fuzzy output system.

Keywords Extreme learning machine · Fuzzy input and fuzzy output · Fuzzy linear regression · Triangular fuzzy number

We thank three anonymous reviewers whose valuable comments and suggestions help us significantly improve this paper. This work is supported by China Postdoctoral Science Foundation (2015M572361), National Natural Science Foundations of China (61503252), and Hebei Province Science and Technology Support Program Project (13210351).

✉ Yu-lin He
csylhe@126.com

Jing Wang
csjwang@yeah.net

[1] College of Mathematics and Information Technology, Xingtai University, Xingtai 054001, China

[2] Modern Education Technology Center, Hebei Institute of Physical Education, Shijiazhuang 050041, China

[3] Big Data Institute, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

[4] Department of Computer Science, Bahauddin Zakariya University, Multan, Pakistan

1 Introduction

Learning systems with crisp input and crisp output have been well studied in the fields of machine learning and pattern recognition. However, in many practical problems, the crisp input or crisp output cannot be given due to the ambiguity and vagueness in human cognition and thinking [22] so that the existing learning systems with crisp input and crisp output cannot be directly applied. For example, the polices predict the age of a suspect according to his or her height and weight, where the suspect's height and weight are sometimes represented in the forms of fuzzy number [e.g., the height is (172, 2, 3 cm) which expresses the height of the suspect is between 170 and 175 cm] because of the limitation of peripheral investigation. In addition, the estimated age is usually a fuzzy value rather than a crisp value, e.g., (25-year-old, 3, 5 years) indicates the age of suspect is between 22 and 30 years old. Such situations also exist in automatic control filed [2], e.g., the water quantity prediction of fuzzy-logic-based washing machine according to the weight of cloths and concentration of washing liquid. Thus, the study on learning system with fuzzy input and fuzzy output is practically required and has the theoretical significance.

A kind of important learning system for dealing with the fuzzy numbers is fuzzy linear regression (FLR) which is firstly introduced by Tanaka et al. [27]. Three basic elements that need to be considered when constructing FLR are input, output and coefficient. In [27], the authors discussed the case of FLR with crisp input and fuzzy output with fuzzy coefficients. The main defect of FLR in [27] is its sensitiveness to outliers [24]. Then, many scholars developed the studies in [27] and proposed some improved FLR models. These improved FLR models can mainly be classified into three categories:

- Crisp input and fuzzy output with fuzzy coefficients. Besides FLR in [27], another representative FLR belonging to this category is proposed in [10]. Different from least-squares-based solving strategy in [27], FLR in [10] used a goal programming approach to determining the fuzzy coefficient.
- Fuzzy input and fuzzy output with crisp coefficients. The typical works belonging to this category can be found from references [1, 3, 11, 19, 20]. Reference [1] calculated the crisp coefficient for FLR by using the normal equations corresponding to a least-squares model. [3] proposed a fuzzy least absolute deviations method to calculate FLR coefficient. [11] used a goal programming approach to determining the crisp coefficients in FLR. References [19] and [20], respectively, used the algebraic formula and fuzzy least-squares to calculate the mentioned areas.
- Fuzzy input and fuzzy output with fuzzy coefficients. Such representative studies can be referred in [13, 25, 26, 29, 31, 32]. [13] used T_W-based fuzzy arithmetic operations to solve FLR's fuzzy coefficients. References [25] and [26], respectively, discussed the solution to fuzzy coefficients under single-objective and multi-objective FLRs. References [29] and [31] constructed the fuzzy least-squares-based FLR estimators. Reference [32] discussed the solution to FLRs with triangular fuzzy number coefficients.

From aforementioned descriptions, we can know that the main strategy to solve FLR is the least-squares methods: classical least-squares (e.g., [1, 25–27]) or fuzzy least-squares (e.g., [3, 20, 29, 31]). Two main limitations exist for least-squares methods: One is high time consumption when optimizing the FLR's coefficients and another is the low efficiency for nonlinear fitting. Thus, it is necessary to explore a new fuzzy regression model which has higher prediction accuracy for nonlinear cases and lower computational complexity.

Recently, a new classification and regression method, named Extreme Learning Machine (ELM) [15, 17], is proposed and attracts more and more attentions from academia and industry. ELM is a special single-hidden layer feedforward neural network (SLFN) where the input layer weights and hidden layer biases are randomly chosen and output layer weights are analytically determined. Due to having no the iterative tuning of weights, ELM has the extremely fast training speed [7, 28]. Meanwhile, the universal approximate capability of ELM has also been theoretically proved [8, 14]. Motivated by developing a fast and efficient fuzzy regression model, in this paper we propose an ELM-based fuzzy regression method ($\mathrm{FR_{ELM}}$) which considers the regression problem with triangular fuzzy input and fuzzy output data. An algorithm for training $\mathrm{FR_{ELM}}$ is designed and its computational complexity is analyzed. Furthermore, the convergence and error estimation for $\mathrm{FR_{ELM}}$ are discussed. Finally, the numerical simulations are conducted to demonstrate the effectiveness of $\mathrm{FR_{ELM}}$.

The rest of this paper is organized as follows. In Sect. 2, a brief introduction to fuzzy input and fuzzy output system is given. In Sect. 3, the ELM-based fuzzy regression model ($\mathrm{FR_{ELM}}$) for fuzzy input and fuzzy output is proposed. In Sect. 4, the convergence of $\mathrm{FR_{ELM}}$ is discussed. In Sect. 5, experimental comparisons are conducted to show the effectiveness of the proposed method. Finally, conclusions are given in Sect. 6.

2 Fuzzy input and fuzzy output system

2.1 Fuzzy number

Definition 1 [21]: Fuzzy number A is expressed as a fuzzy set defined on the domain of real numbers $\Re$ if A meets the following conditions:

1. $\exists x_0 \in \Re$, $\mu_A(x_0) = 1$, where $\mu_A(x)$ is the membership function of fuzzy set A;
2. $\forall \alpha \in (0, 1]$, $A_\alpha = \{x | x \in \Re, \mu_A(x) \geq \alpha\}$ is a finite closed interval.

Triangular fuzzy number A is the most popular fuzzy number, which is represented by two end points a_1 and a_3 and one peak point a_2 as

$A = (a_1, a_2, a_3)$.

It can be interpreted as a membership function

$$\mu_A(x) = \begin{cases} \dfrac{x - a_1}{a_2 - a_1}, & a_1 \leq x \leq a_2 \\ \dfrac{a_3 - x}{a_3 - a_2}, & a_2 \leq x \leq a_3 \\ 0, & x > a_3 \text{ or } x < a_1. \end{cases} \quad (1)$$

The α-cut operation on triangular fuzzy number A can generate a α-cut interval

$$A_\alpha = \left[a_1^{(\alpha)}, a_3^{(\alpha)}\right],$$

where $a_1^{(\alpha)} = (a_2 - a_1)\alpha + a_1$ and $a_3^{(\alpha)} = -(a_3 - a_2)\alpha + a_3$.

2.2 Operations of triangular fuzzy number

Some important operations on triangular fuzzy number are summarized as follows. Assume there are two triangular fuzzy numbers $A = (a_1, a_2, a_3)$ and $B = (b_1, b_2, b_3)$.

- Addition:

$$\begin{aligned} A(+)B &= (a_1, a_2, a_3)(+)(b_1, b_2, b_3) \\ &= (a_1 + b_1, a_2 + b_2, a_3 + b_3); \end{aligned} \quad (2)$$

- Subtraction:

$$A(-)B = (a_1, a_2, a_3)(-)(b_1, b_2, b_3) \\ = (a_1 - b_3, a_2 - b_2, a_3 - b_1); \tag{3}$$

- Multiplication by a real number r:

$$r(\cdot)A = r(\cdot)(a_1, a_2, a_3) \\ = \begin{cases} (ra_1, ra_2, ra_3), & r \geq 0 \\ (ra_3, ra_2, ra_1), & r < 0 \end{cases}; \tag{4}$$

- Multiplication by another triangular fuzzy number [9]:

$$A(\cdot)B = (a_1, a_2, a_3)(\cdot)(b_1, b_2, b_3) \\ \approx (\min\{a_1b_1, a_1b_3, a_3b_1, a_3b_3\}, a_2b_2, \\ \max\{a_1b_1, a_1b_3, a_3b_1, a_3b_3\}); \tag{5}$$

- Division [9]:

$$A(\div)B = (a_1, a_2, a_3)(\div)(b_1, b_2, b_3) \\ \approx \left(\min\left\{\frac{a_1}{b_1}, \frac{a_1}{b_3}, \frac{a_3}{b_1}, \frac{a_3}{b_3}\right\}, \frac{a_2}{b_2}, \right. \\ \left. \max\left\{\frac{a_1}{b_1}, \frac{a_1}{b_3}, \frac{a_3}{b_1}, \frac{a_3}{b_3}\right\}\right), \tag{6}$$

where each of b_1, b_2 and b_3 is not zero.

2.3 Fuzzy linear regression (FLR) with fuzzy input and fuzzy output

Table 1 gives a dataset with triangular fuzzy input and fuzzy output. The objective of FLR is to construct an underlying function with domain $\left(\tilde{X}_1, \tilde{X}_2, \ldots, \tilde{X}_n\right)$ and codomain $\tilde{Y}$ by using the following two models.

1. FLR_1: Fuzzy input and fuzzy output with crisp coefficients

$$\tilde{Y} = a_0 + a_1\tilde{X}_1 + a_2\tilde{X}_2 + \cdots + a_n\tilde{X}_n, \tag{7}$$

where a_0, a_1, a_2, …, a_n are real numbers.

2. FLR_2: Fuzzy input and fuzzy output with fuzzy coefficients

$$\tilde{Y} = \tilde{A}_0 + \tilde{A}_1\tilde{X}_1 + \tilde{A}_2\tilde{X}_2 + \cdots + \tilde{A}_n\tilde{X}_n, \tag{8}$$

where $\tilde{A}_0$, $\tilde{A}_1$, $\tilde{A}_2$, …, $\tilde{A}_n$ are triangular fuzzy numbers.

The keys of two above-mentioned models are the determinations of coefficients a_0, a_1, a_2, …, a_n and $\tilde{A}_0$, $\tilde{A}_1$, $\tilde{A}_2$, …, $\tilde{A}_n$. In order to determine these crisp and fuzzy coefficients, the corresponding distance measure is required to evaluate the similarity between two triangular fuzzy numbers $A = (a_1, a_2, a_3)$ and $B = (b_1, b_2, b_3)$. In this paper, we use the following Euclidean distance which is a kind of simplest similarity measure and firstly introduced by Diamond [4] and then developed by Arabpour and Tata [1]:

$$\text{Dis}(A, B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2}. \tag{9}$$

There are also some other distance measures for FLR, e.g., Y-K distance [30] and D_K distance [5]. Here, our aim is not to evaluate which distance measure is more suitable for FLR. For the sake of simplicity, we select the Euclidean distance as shown in Eq. (9). Regarding the comparison among different distance measures in FLR, the interested readers can refer to [33].

After the distance measure is given, the error criterion for FLR_1 and FLR_2 should be defined to evaluate the qualities of different coefficients. Based on the dataset as shown in Table 4, the error criterion for FLR_1 is

$$E_1 = \sum_{i=1}^{N} \text{Dis}^2\left(a_0 + \sum_{j=1}^{n} a_j\tilde{X}_{ij}, \tilde{Y}_i\right) \\ = \sum_{i=1}^{N}\left[\sum_{k=1}^{3}\left[a_0 + \sum_{j=1}^{n} p_{jk}^{(i)} - y_k^{(i)}\right]^2\right], \tag{10}$$

where

$$\left(p_{j1}^{(i)}, p_{j2}^{(i)}, p_{j3}^{(i)}\right) = \begin{cases} \left(a_jx_{j1}^{(i)}, a_jx_{j2}^{(i)}, a_jx_{j3}^{(i)}\right), & a_j \geq 0 \\ \left(a_jx_{j3}^{(i)}, a_jx_{j2}^{(i)}, a_jx_{j1}^{(i)}\right), & a_j < 0 \end{cases}; \tag{11}$$

and the error criterion for FLR_2 is

Table 1 Triangular fuzzy input and fuzzy output dataset

Data	Fuzzy input				Fuzzy output
	$\tilde{X}_1$	$\tilde{X}_2$	$\cdots$	$\tilde{X}_n$	$\tilde{Y}$
D_1	$\tilde{X}_{11} = \left(x_{11}^{(1)}, x_{12}^{(1)}, x_{13}^{(1)}\right)$	$\tilde{X}_{12} = \left(x_{21}^{(1)}, x_{22}^{(1)}, x_{23}^{(1)}\right)$	$\cdots$	$\tilde{X}_{1n} = \left(x_{n1}^{(1)}, x_{n2}^{(1)}, x_{n3}^{(1)}\right)$	$\tilde{Y}_1 = \left(y_1^{(1)}, y_2^{(1)}, y_3^{(1)}\right)$
D_2	$\tilde{X}_{21} = \left(x_{11}^{(2)}, x_{12}^{(2)}, x_{13}^{(2)}\right)$	$\tilde{X}_{22} = \left(x_{21}^{(2)}, x_{22}^{(2)}, x_{23}^{(2)}\right)$	$\cdots$	$\tilde{X}_{2d} = \left(x_{n1}^{(2)}, x_{n2}^{(2)}, x_{n3}^{(2)}\right)$	$\tilde{Y}_2 = \left(y_1^{(2)}, y_2^{(2)}, y_3^{(2)}\right)$
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$
D_N	$\tilde{X}_{N1} = \left(x_{11}^{(N)}, x_{12}^{(N)}, x_{13}^{(N)}\right)$	$\tilde{X}_{N2} = \left(x_{21}^{(N)}, x_{22}^{(N)}, x_{23}^{(N)}\right)$	$\cdots$	$\tilde{X}_{Nn} = \left(x_{n1}^{(N)}, x_{n2}^{(N)}, x_{n3}^{(N)}\right)$	$\tilde{Y}_N = \left(y_1^{(N)}, y_2^{(N)}, y_3^{(N)}\right)$

$$E_2 = \sum_{i=1}^{N} \mathrm{Dis}^2\left(\tilde{A}_0 + \sum_{j=1}^{n} \tilde{A}_j \tilde{X}_{ij}, \tilde{Y}_i\right) = \sum_{i=1}^{N}\left[\sum_{k=1}^{3}\left[a_{0k} + \sum_{j=1}^{n} q_{jk}^{(i)} - y_k^{(i)}\right]^2\right], \tag{12}$$

where $\tilde{A}_0 = (a_{01}, a_{02}, a_{03})$ and $\tilde{A}_j = (a_{j1}, a_{j2}, a_{j3})$, $j = 1, 2, \ldots, n$ are triangular fuzzy numbers and

$$q_{j1}^{(i)} = \min\left\{a_{j1}x_{j1}^{(i)}, a_{j1}x_{j3}^{(i)}, a_{j3}x_{j1}^{(i)}, a_{j3}x_{j3}^{(i)}\right\}, \tag{13}$$

$$q_{j2}^{(i)} = a_{j2}x_{j2}^{(i)}, \tag{14}$$

$$q_{j3}^{(i)} = \max\left\{a_{j1}x_{j1}^{(i)}, a_{j1}x_{j3}^{(i)}, a_{j3}x_{j1}^{(i)}, a_{j3}x_{j3}^{(i)}\right\}. \tag{15}$$

Meanwhile, the optimal coefficients for minimizing E_1 and E_2 should, respectively, satisfy the some constraint conditions [12, 27]. For the case of crisp coefficient a_0, a_1, a_2, …, a_n, the α-cut interval of fuzzy number $a_0 + \sum_{j=1}^{n} a_j \tilde{X}_{ij}$ corresponding to the i-th fuzzy input D_i, $i = 1, 2, \ldots, N$ should contain the α-cut interval of fuzzy number $\tilde{Y}_i$, i.e., Eq. (16). Similarity, the α-cut interval of fuzzy number $\tilde{A}_0 + \sum_{j=1}^{n} \tilde{A}_j \tilde{X}_{ij}$ corresponding to fuzzy coefficient $\tilde{A}_0$, $\tilde{A}_1$, $\tilde{A}_2$, …, $\tilde{A}_n$ should also contain the α-cut interval of fuzzy number $\tilde{Y}_i$, i.e., Eq. (17).

$$\begin{aligned}&\left[\left(y_2^{(i)} - y_1^{(i)}\right)\alpha + y_1^{(i)}, -\left(y_3^{(i)} - y_2^{(i)}\right)\alpha + y_3^{(i)}\right] \\ &\quad \subseteq \left[\left(\sum_{j=1}^{n} z_{j2}^{(i)} - \sum_{j=1}^{n} z_{j1}^{(i)}\right)\alpha + a_0 + \sum_{j=1}^{n} z_{j1}^{(i)},\right. \\ &\qquad \left. -\left(\sum_{j=1}^{n} z_{j3}^{(i)} - \sum_{j=1}^{n} z_{j2}^{(i)}\right)\alpha + a_0 + \sum_{j=1}^{n} z_{j3}^{(i)}\right]\end{aligned} \tag{16}$$

$$\begin{aligned}&\left[\left(y_2^{(i)} - y_1^{(i)}\right)\alpha + y_1^{(i)}, -\left(y_3^{(i)} - y_2^{(i)}\right)\alpha + y_3^{(i)}\right] \\ &\quad \subseteq \left[\left(a_{02} + \sum_{j=1}^{n} q_{j2}^{(i)} - a_{01} - \sum_{j=1}^{n} q_{j1}^{(i)}\right)\alpha + a_{01} + \sum_{j=1}^{n} q_{j1}^{(i)},\right. \\ &\qquad \left. -\left(a_{03} + \sum_{j=1}^{n} q_{j3}^{(i)} - a_{02} - \sum_{j=1}^{n} q_{j2}^{(i)}\right)\alpha + a_{03} + \sum_{j=1}^{n} q_{j3}^{(i)}\right]\end{aligned} \tag{17}$$

Above all, we can get the following optimization problems corresponding to FLR_1 and FLR_2 models:

$$\begin{array}{lll}\text{Minimize} & & (E_1) \\ \quad \text{s.t.} & \text{Eq. (16)}, & i = 1, 2, \ldots, N\end{array} \tag{18}$$

and

$$\begin{array}{lll}\text{Minimize} & & (E_2) \\ \quad \text{s.t.} & \text{Eq. (17)}, & i = 1, 2, \ldots, N\end{array} \tag{19}$$

By solving Eqs. (18) and (19), we can get the optimal coefficients a_0, $a_j, j = 1, 2, \ldots, n$ and $\tilde{A}_0$, $\tilde{A}_j, j = 1, 2, \ldots, n$.

3 Extreme learning machine-based fuzzy regression model-$\mathrm{FR}_{\mathrm{ELM}}$

3.1 Extreme learning machine (ELM)

ELM is a special single-hidden layer feed-forward neural network (SLFN). Based on the given dataset X containing N samples with n inputs and m outputs, i.e., $X = \{(x_i, \mathrm{y}_i) | x_i = (x_{i1}, \ldots, x_{in}), \mathrm{y}_i = (y_{i1}, \ldots, y_{im}), x_{ij} \in \Re, y_{ik} \in \Re, i = 1, \ldots, N, j = 1, \ldots, n, k = 1, \ldots, m\}$, ELM's mathematical model is:

$$\underset{N\times L}{\mathrm{H}}\ \underset{L\times m}{\beta} = \underset{N\times m}{\mathrm{T}}, \tag{20}$$

where

$$\underset{N\times L}{\mathrm{H}} = \begin{bmatrix} h(x_1) \\ h(x_2) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} g(\mathrm{w}_1 x_1 + b_1) & \cdots & g(\mathrm{w}_L x_1 + b_L) \\ g(\mathrm{w}_1 x_2 + b_1) & \cdots & g(\mathrm{w}_L x_2 + b_L) \\ \vdots & \ddots & \vdots \\ g(\mathrm{w}_1 x_N + b_1) & \cdots & g(\mathrm{w}_L x_N + b_L) \end{bmatrix} \tag{21}$$

is the hidden layer output matrix,

$$\underset{L\times m}{\beta} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_L \end{bmatrix} = \begin{bmatrix} \beta_{11} & \cdots & \beta_{1m} \\ \beta_{21} & \cdots & \beta_{2m} \\ \vdots & \ddots & \vdots \\ \beta_{L1} & \cdots & \beta_{Lm} \end{bmatrix} \tag{22}$$

is the output weight,

$$\underset{N\times m}{\mathrm{T}} = \begin{bmatrix} \mathrm{y}_1 \\ \mathrm{y}_2 \\ \vdots \\ \mathrm{y}_N \end{bmatrix} = \begin{bmatrix} y_{11} & \cdots & y_{1m} \\ y_{21} & \cdots & y_{2m} \\ \vdots & \ddots & \vdots \\ y_{N1} & \cdots & y_{Nm} \end{bmatrix} \tag{23}$$

is the target output,

$$\underset{n\times L}{\mathrm{w}} = [\mathrm{w}_1 \ \cdots \ \mathrm{w}_L] = \begin{bmatrix} w_{11} & \cdots & w_{L1} \\ w_{12} & \cdots & w_{L2} \\ \vdots & \ddots & \vdots \\ w_{1n} & \cdots & w_{Ln} \end{bmatrix} \tag{24}$$

is the input weight, $\mathrm{b} = [b_1, \ldots, b_L]$ is the hidden layer bias vector, L is the number of hidden layer nodes, and g is an infinitely differential activation function.

Unlike the iterative gradient descent-based training algorithms for SLFNs, the input weight w and hidden layer bias b in ELM [17] are randomly chosen and output weight β is analytically determined as follows:

$$\underset{L\times m}{\beta} = \underset{L\times N}{\mathrm{H}^{\dagger}}\underset{N\times m}{\mathrm{T}} = \begin{cases} (\mathrm{H}^{\mathrm{T}}\mathrm{H})^{-1}\mathrm{H}^{\mathrm{T}}\mathrm{T}, & N \geq L \\ \mathrm{H}^{\mathrm{T}}(\mathrm{H}\mathrm{H}^{\mathrm{T}})^{-1}\mathrm{T}, & N < L \end{cases}, \tag{25}$$

where $\mathrm{H}^{\dagger}$ is the Moore-Penrose generalized inverse of matrix H.

3.2 FR$_{\mathrm{ELM}}$ and its training algorithm

The representative works regarding using neural networks to conduct fuzzy regression analysis can be found from references [16] and [18]. These works all studied the case of crisp input–interval output with interval coefficient and used the back-propagation algorithm to train the designed neural networks. As far as we know, there are no works that have discussed how to design neural networks to deal with the fuzzy regression with fuzzy input and fuzzy output. Here, we give a new fuzzy regression model, FR$_{\mathrm{ELM}}$, which use ELM to deal with fuzzy input and fuzzy output. Its structure is presented in Fig. 1. As shown in Fig. 1, we introduce the training procedures of FR$_{\mathrm{ELM}}$.

In FR$_{\mathrm{ELM}}$, there are totally six ELMs which are, respectively, used to estimate the left endpoint (ELM$_1$ and ELM$'_1$), peak point (ELM$_2$ and ELM$'_2$) and right endpoint (ELM$_3$ and ELM$'_3$) of a triangular fuzzy number. Here, regarding the training of ELM$_1$ and ELM$_3$, we refer to I-T algorithm in [18].

1. Training for ELM$_1$. Dataset

$$X_1 = \{(x_1^{(i)}, y_1^{(i)}) | x_1^{(i)} = (x_{11}^{(i)}, x_{21}^{(i)}, \ldots, x_{n1}^{(i)}), i = 1, 2, \ldots, N\}$$

is used in this phase. The loss function for ELM$_1$ is

$$L_{\mathrm{ELM}_1} = \frac{1}{2}\sum_{i=1}^{N}[w_1^{(i)}(o_1^{(i)} - y_1^{(i)})^2], \tag{26}$$

where $o_1^{(i)} = h_1(x_1^{(i)})\beta_1$ is the actual output of sample $x^{(i)} = (x_{11}^{(i)}, x_{21}^{(i)}, \ldots, x_{n1}^{(i)})$ and

$$w_1^{(i)} = \begin{cases} 1, & y_1^{(i)} > t_1^{(i)} \\ 0.001, & y_1^{(i)} \leq t_1^{(i)} \end{cases}. \tag{27}$$

$t_1^{(i)} = h_1(x_1^{(i)})\beta'_1$, $i = 1, 2, \ldots, N$ is calculated with an additional ELM$'_1$, where β'_1 can be determined as Eq. (25). The role of weight $w_1^{(i)}$ is to guarantee the samples of which the real left endpoint is greater than the estimated left endpoint can be fully used to train ELM$_1$. The α-cut interval of real fuzzy output has higher probability falling into α-cut interval of actual fuzzy output when the real left endpoint is greater than the estimated left endpoint. Let $\frac{\partial L_{\mathrm{ELM}_1}}{\partial \beta_1} = 0$, we can get

$$\beta_1 = \begin{cases} (\mathrm{H}_1^{\mathrm{T}}\mathrm{W}_1\mathrm{H})_1^{-1}\mathrm{H}_1^{\mathrm{T}}\mathrm{W}_1\mathrm{T}_1, & N \geq L \\ \mathrm{H}_1^{\mathrm{T}}(\mathrm{W}_1\mathrm{H}_1\mathrm{H}_1^{\mathrm{T}})^{-1}\mathrm{W}_1\mathrm{T}_1, & N < L \end{cases}, \tag{28}$$

where $\mathrm{W}_1 = diag[w_1^{(1)}, w_1^{(2)}, \ldots, w_1^{(N)}]$.

2. Training for ELM$_3$. Dataset

$$X_3 = \{(x_3^{(i)}, y_3^{(i)}) | x_3^{(i)} = (x_{13}^{(i)}, x_{23}^{(i)}, \ldots, x_{n3}^{(i)}), \quad i = 1, 2, \ldots, N\}$$

is used in this phase. The loss function for ELM$_3$ is

$$L_{\mathrm{ELM}_3} = \frac{1}{2}\sum_{i=1}^{N}[w_3^{(i)}(o_3^{(i)} - y_3^{(i)})^2], \tag{29}$$

where $o_3^{(i)} = h_2(x_3^{(i)})\beta_3$ is the actual output of sample $x_3^{(i)} = (x_{13}^{(i)}, x_{23}^{(i)}, \ldots, x_{n3}^{(i)})$ and

$$w_3^{(i)} = \begin{cases} 1, & y_3^{(i)} < t_3^{(i)} \\ 0.001, & y_3^{(i)} \geq t_3^{(i)} \end{cases}. \tag{30}$$

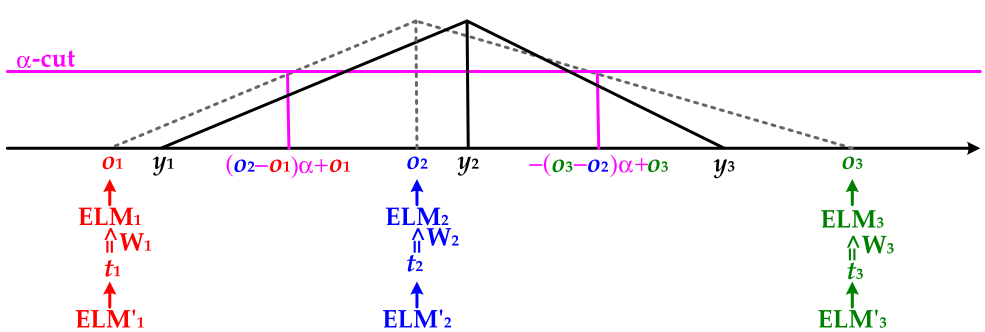

Fig. 1 An illustration of FR$_{\mathrm{ELM}}$. The real fuzzy output of fuzzy input $((x_{11}, x_{12}, x_{13}), (x_{21}, x_{22}, x_{23}), \ldots, (x_{n1}, x_{n2}, x_{n3}))$ is (y_1, y_2, y_3) and its actual fuzzy output calculated with FR$_{\mathrm{ELM}}$ is (o_1, o_2, o_3). (t_1, t_2, t_3) is used to determine the weight matrices W_1, W_2 and W_3.

$t_3^{(i)} = h_2(x_3^{(i)})\beta_3'$, $i = 1, 2, \ldots, N$ is calculated with an additional ELM_3', where β_3' can be determined as Eq. (25). The role of weight $w_3^{(i)}$ is to guarantee the samples of which the real right endpoint is smaller than the estimated right endpoint can be fully used to train ELM_3. The α-cut interval of real fuzzy output has higher probability falling into α-cut interval of actual fuzzy output when the real right endpoint is smaller than the estimated right endpoint. Let $\frac{\partial L_{\mathrm{ELM}_3}}{\partial \beta_3} = 0$, we can get

$$\beta_3 = \begin{cases} (\mathrm{H}_3^{\mathrm{T}}\mathrm{W}_3\mathrm{H})_3^{-1}\mathrm{H}_3^{\mathrm{T}}\mathrm{W}_3\mathrm{T}_3, & N \geq L \\ \mathrm{H}_3^{\mathrm{T}}(\mathrm{W}_3\mathrm{H}_3\mathrm{H}_3^{\mathrm{T}})^{-1}\mathrm{W}_3\mathrm{T}_3, & N < L \end{cases}, \tag{31}$$

where $\mathrm{W}_3 = diag[w_3^{(1)}, w_3^{(2)}, \ldots, w_3^{(N)}]$.

3. Training for ELM_2. Dataset

$$X_2 = \{(x_2^{(i)}, y_2^{(i)}) | x_2^{(i)} = (x_{12}^{(i)}, x_{22}^{(i)}, \ldots, x_{n2}^{(i)}), i = 1, 2, \ldots, N\}$$

is used in this phase. The loss function for ELM_2 is

$$L_{\mathrm{ELM}_2} = \frac{1}{2}\sum_{i=1}^{N}[w_2^{(i)}(o_2^{(i)} - y_2^{(i)})^2], \tag{32}$$

where $o_2^{(i)} = h_3(x_2^{(i)})\beta_2$ is the actual output of sample $x_2^{(i)} = (x_{12}^{(i)}, x_{22}^{(i)}, \ldots, x_{n2}^{(i)})$ and $w_2^{(i)}$ as shown in Eq. (33).

$$w_2^{(i)} = \begin{cases} 1, & [(y_2^{(i)} - y_1^{(i)})\alpha + y_1^{(i)}, -(y_3^{(i)} - y_2^{(i)})\alpha + y_3^{(i)}] \subseteq [(t_2^{(i)} - o_1^{(i)})\alpha + o_1^{(i)}, -(o_3^{(i)} - t_2^{(i)})\alpha + o_3^{(i)}] \\ 0.001, & [(y_2^{(i)} - y_1^{(i)})\alpha + y_1^{(i)}, -(y_3^{(i)} - y_2^{(i)})\alpha + y_3^{(i)}] \not\subseteq [(t_2^{(i)} - o_1^{(i)})\alpha + o_1^{(i)}, -(o_3^{(i)} - t_2^{(i)})\alpha + o_3^{(i)}] \end{cases} \tag{33}$$

$t_2^{(i)} = h_3(x_2^{(i)})\beta_2'$, $i = 1, 2, \ldots, N$ is calculated with an additional ELM_2', where β_2' can be determined as Eq. (25). The role of weight $w_2^{(i)}$ is to guarantee the samples of which the α-cut interval of real fuzzy output is contained in the α-cut interval of actual fuzzy output can be fully used to train ELM_2. In this situation, the peak point of actual fuzzy output will be closer to the peak point of real fuzzy output. Let $\frac{\partial L_{\mathrm{ELM}_2}}{\partial \beta_2} = 0$, we can get

$$\beta_2 = \begin{cases} (\mathrm{H}_2^{\mathrm{T}}\mathrm{W}_2\mathrm{H})_2^{-1}\mathrm{H}_2^{\mathrm{T}}\mathrm{W}_2\mathrm{T}_2, & N \geq L \\ \mathrm{H}_2^{\mathrm{T}}(\mathrm{W}_2\mathrm{H}_2\mathrm{H}_2^{\mathrm{T}})^{-1}\mathrm{W}_2\mathrm{T}_2, & N < L \end{cases}, \tag{34}$$

where $\mathrm{W}_2 = \mathrm{diag}[w_2^{(1)}, w_2^{(2)}, \ldots, w_2^{(N)}]$.

After we get ELM_1 (i.e., β_1), ELM_2 (i.e., β_2) and ELM_3 (i.e., β_3), the actual fuzzy output for D_i, $i = 1, 2, \ldots, N$ can be calculated as $(o_1^{(i)}, o_2^{(i)}, o_3^{(i)})$. However, because ELM_1, ELM_2 and ELM_3 are independently determined, there may be a case that $o_1^{(i)} \leq o_2^{(i)} \leq o_3^{(i)}$ does not hold. At this time, ELM_1, ELM_2 and ELM_3 are meaningless for D_i. Thus, we give the following tuning strategy to deal with this case. Let

$$F(D_i) = (f_1^{(i)}, f_2^{(i)}, f_3^{(i)})$$

represent the final output of D_i via fuzzy regression system $\mathrm{FR}_{\mathrm{ELM}}$, where

$$\begin{cases} f_1^{(i)} = \min\{o_1^{(i)}, o_2^{(i)}, o_3^{(i)}\} \\ f_2^{(i)} = \mathrm{med}\{o_1^{(i)}, o_2^{(i)}, o_3^{(i)}\} \\ f_3^{(i)} = \max\{o_1^{(i)}, o_2^{(i)}, o_3^{(i)}\} \end{cases}. \tag{35}$$

The feasibility of this tuning strategy can be reflected from the following example as shown in Fig. 2. Assume $o_2^{(i)} \leq o_3^{(i)} \leq o_1^{(i)}$ and $[y_1^{(i)}, y_3^{(i)}] \subset [o_2^{(i)}, o_1^{(i)}]$ hold for D_i, then the estimated errors of on D_i based on $(o_1^{(i)}, o_2^{(i)}, o_3^{(i)})$ and $(f_1^{(i)}, f_2^{(i)}, f_3^{(i)})$ can be, respectively, expressed as

$$e_i = \sqrt{(y_1^{(i)} - o_1^{(i)})^2 + (y_2^{(i)} - o_2^{(i)})^2 + (y_3^{(i)} - o_3^{(i)})^2} \tag{36}$$

and

$$e_i' = \sqrt{(y_1^{(i)} - f_1^{(i)})^2 + (y_2^{(i)} - f_2^{(i)})^2 + (y_3^{(i)} - f_3^{(i)})^2}. \tag{37}$$

We can easily find $e_i > e_i'$ hold for the example as shown in Fig. 2.

4 Convergence and error estimation analysis

This section discusses the convergence and estimated error of $\mathrm{FR}_{\mathrm{ELM}}$. We try to extend the theoretical results (in [17] and [8]) focusing on the real number dataset to triangular fuzzy number dataset.

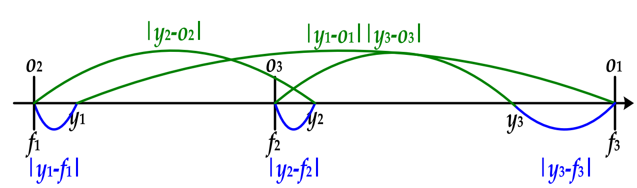

Fig. 2 Feasibility of Eq. (35)

4.1 Convergence

Theorem 1 (Convergence on the real number dataset) [17]: *Given any small positive value* $\varepsilon > 0$ *and activation function* $g : \Re \longrightarrow \Re$ *which is infinitely differentiable in any interval, there exists* $L \leq N$ *such that for* N *arbitrary distinct samples* $\{(x_i, \mathrm{y}_i)|x_i = (x_{i1}, \ldots, x_{in}), \mathrm{y}_i = (y_{i1}, \ldots, y_{im}), \quad x_{ij} \in \Re, y_{ik} \in \Re, i = 1, \ldots, N; j = 1, \ldots, n; k = 1, \ldots, m\}$ *and any* $w_{lj} \in \Re$ *and* $b_l \in \Re$, $l = 1, 2, \ldots, L$,

$$\left\| \underset{N\times L}{\mathrm{H}} \underset{L\times m}{\beta} - \underset{N\times m}{\mathrm{T}} \right\| < \varepsilon \tag{38}$$

holds for any continuous probability distribution with probability 1.

Theorem 2 (Convergence on the triangular fuzzy number dataset): *Given any three small positive values* $\varepsilon_1, \varepsilon_2, \varepsilon_3 > 0$ *and activation function* $g : \Re \longrightarrow \Re$ *which is infinitely differentiable in any interval, there exists* $L_1, L_2, L_3 \leq N$ *such that for* N *arbitrary triangular fuzzy number samples* $\{(\tilde{X}_{i1}, \tilde{X}_{i2}, \ldots, \tilde{X}_{in}, \tilde{Y}_i)|\tilde{X}_{ij} = (x_{j1}^{(i)}, x_{j2}^{(i)}, x_{j3}^{(i)}), \tilde{Y}_i = (y_1^{(i)}, y_2^{(i)}, y_3^{(i)}), x_{jk}^{(i)} \in \Re, y_k^{(i)} \in \Re, i = 1, 2, \quad \ldots, N, j = 1, 2, \ldots, n, k = 1, 2, 3\}$ *and any input weights* $\mathrm{w}_{1n\times L}$, $\mathrm{w}_{2n\times L}$, $\underset{n\times L}{\mathrm{w}_3}$ *and hidden layer biases* $\underset{1\times L}{\mathrm{b}_1}, \underset{1\times L}{\mathrm{b}_2}, \underset{1\times L}{\mathrm{b}_3}$,

$$\left\| \underset{N\times L}{\mathrm{H}_1} \underset{L\times 1}{\beta_1} - \underset{N\times 1}{\mathrm{T}_1} \right\| < \varepsilon_1, \tag{39}$$

$$\left\| \underset{N\times L}{\mathrm{H}_2} \underset{L\times 1}{\beta_2} - \underset{N\times 1}{\mathrm{T}_2} \right\| < \varepsilon_2, \tag{40}$$

$$\left\| \underset{N\times L}{\mathrm{H}_3} \underset{L\times 1}{\beta_3} - \underset{N\times 1}{\mathrm{T}_3} \right\| < \varepsilon_3 \tag{41}$$

hold for any continuous probability distribution with probability 1.

Proof The validity of this theorem is obvious. Eqs. (39)–(41) can, respectively, be proved according to Theorem 1. □

4.2 Error estimation analysis

The objective of error estimation analysis is to investigate the change of $\mathrm{FR_{ELM}}$'s estimated error with the gradual increase of hidden layer nodes.

Theorem 3 (Error estimation on the real number dataset) [8]: *Let* $\underset{N\times(L+1)}{\mathrm{H}^*}$ *(Eq. (42)) and*

$$\underset{N\times(L+1)}{\mathrm{H}^*} = \begin{bmatrix} g(\mathrm{w}_1 x_1 + b_1) & \cdots & g(\mathrm{w}_L x_1 + b_L) & g(\mathrm{w}_{L+1} x_1 + b_{L+1}) \\ g(\mathrm{w}_1 x_2 + b_1) & \cdots & g(\mathrm{w}_L x_2 + b_L) & g(\mathrm{w}_{L+1} x_2 + b_{L+1}) \\ \vdots & \ddots & \vdots & \vdots \\ g(\mathrm{w}_1 x_N + b_1) & \cdots & g(\mathrm{w}_L x_N + b_L) & g(\mathrm{w}_{L+1} x_N + b_{L+1}) \end{bmatrix} \tag{42}$$

$$\underset{(L+1)\times m}{\beta^*} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_L \\ \beta_{L+1} \end{bmatrix} = \begin{bmatrix} \beta_{11} & \cdots & \beta_{1m} \\ \vdots & \ddots & \vdots \\ \beta_{L1} & \cdots & \beta_{Lm} \\ \beta_{L+1,1} & \cdots & \beta_{L+1,m} \end{bmatrix} \tag{43}$$

denote the hidden layer output matrix and output weight corresponding to ELM with $L+1$ *hidden layer nodes. Then,*

$$\min_{\underset{(L+1)\times m}{\beta^*}} \left\| \underset{N\times(L+1)}{\mathrm{H}^*} \underset{(L+1)\times m}{\beta^*} - \underset{N\times m}{\mathrm{T}} \right\| \leq \min_{\underset{L\times m}{\beta}} \left\| \underset{N\times L}{\mathrm{H}} \underset{L\times m}{\beta} - \underset{N\times m}{\mathrm{T}} \right\|. \tag{44}$$

Theorem 3 tells us that when we incrementally train an ELM by gradually adding nodes in hidden layer, the training error is monotonically decreasing.

Theorem 4 (Error estimation on the triangular fuzzy number dataset): *Eq. (45) holds for* $\underset{N\times(L+1)}{\mathrm{H}_1^*}$, $\underset{(L+1)\times 1}{\beta_1^*}$, $\underset{N\times(L+1)}{\mathrm{H}_2^*}$, $\underset{(L+1)\times 1}{\beta_2^*}$, $\underset{N\times(L+1)}{\mathrm{H}_3^*}$ *and* $\underset{(L+1)\times 1}{\beta_3^*}$.

$$\begin{aligned} &\min_{\underset{(L+1)\times 1}{\beta_1^*}, \underset{(L+1)\times 1}{\beta_2^*}, \underset{(L+1)\times 1}{\beta_3^*}} \left[\left\| \underset{N\times(L+1)}{\mathrm{H}_1^*} \underset{(L+1)\times 1}{\beta_1^*} - \underset{N\times 1}{\mathrm{T}_1} \right\| \right. \\ &\left. + \left\| \underset{N\times(L+1)}{\mathrm{H}_2^*} \underset{(L+1)\times 1}{\beta_2^*} - \underset{N\times 1}{\mathrm{T}_2} \right\| + \left\| \underset{N\times(L+1)}{\mathrm{H}_3^*} \underset{(L+1)\times 1}{\beta_3^*} - \underset{N\times 1}{\mathrm{T}_3} \right\| \right] \\ &\leq \min_{\underset{L\times 1}{\beta_1}, \underset{L\times 1}{\beta_2}, \underset{L\times 1}{\beta_3}} \left[\left\| \underset{N\times L}{\mathrm{H}_1} \underset{L\times 1}{\beta_1} - \underset{N\times 1}{\mathrm{T}_1} \right\| + \left\| \underset{N\times L}{\mathrm{H}_2} \underset{L\times 1}{\beta_2} - \underset{N\times 1}{\mathrm{T}_2} \right\| \right. \\ &\left. + \left\| \underset{N\times L}{\mathrm{H}_3} \underset{L\times 1}{\beta_3} - \underset{N\times 1}{\mathrm{T}_3} \right\| \right] \end{aligned} \tag{45}$$

Proof According to Theorem 3, we get

$$\min_{\underset{(L+1)\times 1}{\beta_1^*}} \left\| \underset{N\times(L+1)}{\mathrm{H}_1^*} \underset{(L+1)\times 1}{\beta_1^*} - \underset{N\times 1}{\mathrm{T}_1} \right\| \leq \min_{\underset{L\times 1}{\beta_1}} \left\| \underset{N\times L}{\mathrm{H}_1} \underset{L\times 1}{\beta_1} - \underset{N\times 1}{\mathrm{T}_1} \right\|, \tag{46}$$

$$\begin{aligned}\min_{\substack{\beta_2^*\\(L+1)\times 1}}\left\|\underset{N\times(L+1)}{\mathrm{H}_2^*}\ \underset{(L+1)\times 1}{\beta_2^*}-\underset{N\times 1}{\mathrm{T}_2}\right\| \\ \leq \min_{\substack{\beta_2\\L\times 1}}\left\|\underset{N\times L}{\mathrm{H}_2}\ \underset{L\times 1}{\beta_2}-\underset{N\times 1}{\mathrm{T}_2}\right\|\end{aligned}, \tag{47}$$

$$\begin{aligned}\min_{\substack{\beta_3^*\\(L+1)\times 1}}\left\|\underset{N\times(L+1)}{\mathrm{H}_3^*}\ \underset{(L+1)\times 1}{\beta_3^*}-\underset{N\times 1}{\mathrm{T}_3}\right\| \\ \leq \min_{\substack{\beta_3\\L\times 1}}\left\|\underset{N\times L}{\mathrm{H}_3}\ \underset{L\times 1}{\beta_3}-\underset{N\times 1}{\mathrm{T}_3}\right\|\end{aligned}. \tag{48}$$

Then, Eq. (45) can be easily obtained. □

5 Experimental demonstration

In this section, we select three commonly used triangular fuzzy number datasets, i.e., restaurants (in Table 2) [6], CEC-OM-SAND (in Table 3) [23] and Hsien–Chung Wu (in Table 4) [29], to test the regression performance of our proposed $\mathrm{FR_{ELM}}$. We compare the estimation errors and training times of $\mathrm{FLR_1}$, $\mathrm{FLR_2}$ and $\mathrm{FR_{ELM}}$, which are all implemented with MATLAB 7.1. The estimation error on single sample is measured by Eq. (36), and average estimation error on whole dataset is measured by

$$\mathrm{AveE}=\frac{1}{N}\sum_{i=1}^{N}e_i. \tag{49}$$

The training times of all models are, respectively, the average values of 10 independent training times corresponding to 10 repeated runs. All the experiments are configured on a Thinkpad T440s PC having Windows 8.1 with i5-4200 1.60 GHz processor and 8 GB RAM.

The experimental results in Tables 2, 3 and 4 corresponding to $\mathrm{FR_{ELM}}$ are obtained by ELMs with Sigmoid activation function $g(x)=\frac{1}{1+e^{-x}}$ and 1000 hidden layer nodes. The input layer weights and hidden layer biases are randomly selected from interval $[0,1]$. From Tables 2, 3 and 4, we can find that

1. For 17 samples in restaurants, all samples in CEC-OM-SAND, and all samples in Hsien–Chung Wu, $\mathrm{FR_{ELM}}$ obtains the less estimation errors. In particular, the estimated errors on samples belonging to CEC-OM-SAND and Hsien–Chung Wu are nearly close to 0.
2. For every dataset, $\mathrm{FR_{ELM}}$ obtains the less average estimation error, e.g., 0.921 for restaurants, 0.0000000002 for CEC-OM-SAND and 0.000089 for Hsien–Chung Wu.
3. $\mathrm{FR_{ELM}}$ gets the lower training time in comparison with $\mathrm{FLR_1}$ and $\mathrm{FLR_2}$. Compared with $\mathrm{FLR_1}$ and $\mathrm{FLR_2}$, the reduction of training time for our proposed ELM-based fuzzy regression algorithm is above 10 times, even though ELMs used in $\mathrm{FR_{ELM}}$ are with 1000 hidden layer nodes.

The above-mentioned experimental results reflect that our proposed $\mathrm{FR_{ELM}}$ not only has the better fitting capacity for triangular fuzzy number but also the lower computational complexity compared with the existing fuzzy linear regression models, e.g., $\mathrm{FLR_1}$ and $\mathrm{FLR_2}$, which have two main shortcomings: one is to fit the nonlinear fuzzy data with linear regression models and another is the high computational time used to determine the least-squares solutions for Eqs. (10) and (12). However, these two shortcomings are effectively overcame by ELMs used in our proposed $\mathrm{FR_{ELM}}$.

In addition, we also test the convergence of $\mathrm{FR_{ELM}}$, i.e., the change of estimated error of $\mathrm{FR_{ELM}}$ with the increase of hidden layer nodes in ELMs. The numbers of hidden layer nodes in every ELM are ranged from 10 to 1000 with a step of 10. For CEC-OM-SAND and Hsien–Chung Wu datasets, we plot the estimated errors corresponding to different numbers of hidden layer nodes in Fig. 3. From these figures, we can see that our proposed $\mathrm{FR_{ELM}}$ is convergent, i.e., the estimated error gradually approaches 0 with the increase of hidden layer nodes. These experimental results demonstrate the validity of our theoretical analysis conducted in Sect. 4.

6 Concluding remarks

In this paper, we proposed an extreme learning machine-based fuzzy regression method ($\mathrm{FR_{ELM}}$) which can effectively deal with the regression analysis with triangular fuzzy input and fuzzy output data. We designed the new training algorithm for $\mathrm{FR_{ELM}}$ and theoretical analyzed its the convergence and error estimation. The final experimental results show $\mathrm{FR_{ELM}}$ has the better fitting performance and the lower computational complexity compared with the existing fuzzy linear regression models for

Table 2 Restaurants dataset [6]

	Data			FLR_1		FLR_2		FR_{ELM}	
	$\tilde{X}_1$	$\tilde{X}_2$	$\tilde{Y}$	Predicted value	Error	Predicted value	Error	Predicted value	Error
1	(6.50, 7.00, 8.25)	(7.25, 8.00, 9.00)	(7.25, 8.00, 9.00)	(6.49, 7.11, 8.32)	1.350	(6.67, 7.21, 8.28)	1.216	(6.90, 7.40, 8.48)	0.871
2	(6.50, 7.00, 8.25)	(6.50, 7.00, 8.25)	(5.75, 6.00, 6.50)	(6.20, 6.72, 8.03)	1.749	(6.41, 6.75, 7.77)	1.621	(6.28, 6.70, 7.65)	1.445
3	(5.75, 6.00, 6.50)	(6.50, 7.00, 8.25)	(5.75, 6.00, 6.50)	(5.71, 6.07, 6.88)	0.391	(5.50, 5.77, 6.35)	0.369	(5.60, 5.80, 6.40)	0.269
4	(7.25, 8.00, 9.00)	(9.00, 9.00, 10.00)	(9.00, 9.00, 10.00)	(7.67, 8.16, 9.20)	1.767	(8.18, 8.65, 9.56)	0.992	(9.00, 9.00, 10.00)	0.000
5	(7.25, 8.00, 9.00)	(7.25, 8.00, 9.00)	(7.25, 8.00, 9.00)	(6.98, 7.77, 8.81)	0.401	(7.58, 8.20, 8.89)	0.398	(7.25, 8.00, 9.00)	0.000
6	(5.75, 6.00, 6.50)	(6.50, 7.00, 8.25)	(5.00, 5.00, 6.00)	(5.71, 6.07, 6.88)	1.556	(5.50, 5.77, 6.35)	0.984	(5.60, 5.80, 6.40)	1.077
7	(6.50, 7.00, 8.25)	(7.25, 8.00, 9.00)	(6.50, 7.00, 8.25)	(6.49, 7.11, 8.32)	0.133	(6.67, 7.21, 8.28)	0.272	(6.90, 7.40, 8.48)	0.609
8	(6.50, 7.00, 8.25)	(6.50, 7.00, 8.25)	(5.00, 5.00, 6.00)	(6.20, 6.72, 8.03)	2.918	(6.41, 6.75, 7.77)	2.866	(6.28, 6.70, 7.65)	2.690
9	(6.50, 7.00, 8.25)	(7.25, 8.00, 9.00)	(6.50, 7.00, 8.25)	(6.49, 7.11, 8.32)	0.133	(6.67, 7.21, 8.28)	0.272	(6.90, 7.40, 8.48)	0.609
10	(5.75, 6.00, 6.50)	(6.50, 7.00, 8.25)	(5.75, 6.00, 6.50)	(5.71, 6.07, 6.88)	0.391	(5.50, 5.77, 6.35)	0.369	(5.60, 5.80, 6.40)	0.269
11	(6.50, 7.00, 8.25)	(7.25, 8.00, 9.00)	(7.25, 8.00, 9.00)	(6.49, 7.11, 8.32)	1.350	(6.67, 7.21, 8.28)	1.216	(6.90, 7.40, 8.48)	0.871
12	(6.50, 7.00, 8.25)	(5.75, 6.00, 6.50)	(5.75, 6.00, 6.50)	(5.91, 6.33, 7.34)	0.920	(6.15, 6.30, 6.59)	0.509	(5.75, 6.00, 6.50)	0.000
13	(6.50, 7.00, 8.25)	(7.25, 8.00, 9.00)	(9.00, 9.00, 10.00)	(6.49, 7.11, 8.32)	3.559	(6.67, 7.21, 8.28)	3.405	(6.90, 7.40, 8.48)	3.049
14	(6.50, 7.00, 8.25)	(7.25, 8.00, 9.00)	(7.25, 8.00, 9.00)	(6.49, 7.11, 8.32)	1.350	(6.67, 7.21, 8.28)	1.216	(6.90, 7.40, 8.48)	0.871
15	(6.50, 7.00, 8.25)	(6.50, 7.00, 8.25)	(6.50, 7.00, 8.25)	(6.20, 6.72, 8.03)	0.466	(6.41, 6.75, 7.77)	0.544	(6.28, 6.70, 7.65)	0.708
16	(6.50, 7.00, 8.25)	(6.50, 7.00, 8.25)	(6.50, 7.00, 8.25)	(6.20, 6.72, 8.03)	0.466	(6.41, 6.75, 7.77)	0.544	(6.28, 6.70, 7.65)	0.708
17	(5.75, 6.00, 6.50)	(6.50, 7.00, 8.25)	(5.75, 6.00, 6.50)	(5.71, 6.07, 6.88)	0.391	(5.50, 5.77, 6.35)	0.369	(5.60, 5.80, 6.40)	0.269
18	(6.50, 7.00, 8.25)	(7.25, 8.00, 9.00)	(6.50, 7.00, 8.25)	(6.49, 7.11, 8.32)	0.133	(6.67, 7.21, 8.28)	0.272	(6.90, 7.40, 8.48)	0.609
19	(6.50, 7.00, 8.25)	(6.50, 7.00, 8.25)	(7.25, 8.00, 9.00)	(6.20, 6.72, 8.03)	1.919	(6.41, 6.75, 7.77)	1.939	(6.28, 6.70, 7.65)	2.113
20	(6.50, 7.00, 8.25)	(9.00, 9.00, 10.00)	(6.50, 7.00, 8.25)	(7.18, 7.50, 8.71)	0.961	(7.27, 7.67, 8.96)	1.240	(6.50, 7.00, 8.25)	0.000
21	(6.50, 7.00, 8.25)	(7.25, 8.00, 9.00)	(6.50, 7.00, 8.25)	(6.49, 7.11, 8.32)	0.133	(6.67, 7.21, 8.28)	0.272	(6.90, 7.40, 8.48)	0.609
22	(6.50, 7.00, 8.25)	(7.25, 8.00, 9.00)	(5.75, 6.00, 6.50)	(6.49, 7.11, 8.32)	2.259	(6.67, 7.21, 8.28)	2.341	(6.90, 7.40, 8.48)	2.680
23	(6.50, 7.00, 8.25)	(9.00, 9.00, 10.00)	(6.50, 7.00, 8.25)	(7.18, 7.50, 8.71)	0.961	(7.27, 7.67, 8.96)	1.240	(6.50, 7.00, 8.25)	0.000
24	(6.50, 7.00, 8.25)	(6.50, 7.00, 8.25)	(7.25, 8.00, 9.00)	(6.20, 6.72, 8.03)	1.919	(6.41, 6.75, 7.77)	1.939	(6.28, 6.70, 7.65)	2.113
25	(6.50, 7.00, 8.25)	(6.50, 7.00, 8.25)	(5.75, 6.00, 6.50)	(6.20, 6.72, 8.03)	1.749	(6.41, 6.75, 7.77)	1.621	(6.28, 6.70, 7.65)	1.445
26	(6.50, 7.00, 8.25)	(6.50, 7.00, 8.25)	(5.75, 6.00, 6.50)	(6.20, 6.72, 8.03)	1.749	(6.41, 6.75, 7.77)	1.621	(6.28, 6.70, 7.65)	1.445
27	(6.50, 7.00, 8.25)	(6.50, 7.00, 8.25)	(6.50, 7.00, 8.25)	(6.20, 6.72, 8.03)	0.466	(6.41, 6.75, 7.77)	0.544	(6.28, 6.70, 7.65)	0.708
28	(6.50, 7.00, 8.25)	(7.25, 8.00, 9.00)	(6.50, 7.00, 8.25)	(6.49, 7.11, 8.32)	0.133	(6.67, 7.21, 8.28)	0.272	(6.90, 7.40, 8.48)	0.609
29	(6.50, 7.00, 8.25)	(6.50, 7.00, 8.25)	(6.50, 7.00, 8.25)	(6.20, 6.72, 8.03)	0.466	(6.41, 6.75, 7.77)	0.544	(6.28, 6.70, 7.65)	0.708
30	(5.75, 6.00, 6.50)	(6.50, 7.00, 8.25)	(5.75, 6.00, 6.50)	(5.71, 6.07, 6.88)	0.391	(5.50, 5.77, 6.35)	0.369	(5.60, 5.80, 6.40)	0.269
Average error				1.085		1.046		0.921	
Training time (s)				0.431		0.395		0.037	

Table 3 CEC-OM-SAND dataset [23]

	Data			FLR_1		FLR_2		FR_{ELM}	
	$\tilde{X}_1$	$\tilde{X}_2$	$\tilde{Y}$	Predicted value	Error	Predicted value	Error	Predicted value	Error
1	(0.85, 0.88, 0.99)	(33.28, 35.00, 38.55)	(15.61, 16.50, 18.69)	(16.88, 16.65, 16.32)	2.693	(15.35, 16.38, 18.62)	0.290	(15.61, 16.50, 18.69)	0.0000000002
2	(1.04, 1.13, 1.28)	(36.49, 37.00, 42.25)	(16.80, 18.60, 20.77)	(16.86, 17.05,16.52)	4.528	(15.12, 16.56, 18.55)	3.456	(16.80, 18.60, 20.77)	0.0000000002
3	(1.20, 1.31, 1.47)	(26.16, 27.00, 30.39)	(18.54, 19.30, 21.68)	(19.30, 19.48, 19.34)	2.471	(17.85, 19.22, 21.59)	0.699	(18.54, 19.30, 21.68)	0.0000000002
4	(1.82, 1.98, 2.23)	(26.68, 29.00, 33.23)	(19.27, 20.30, 23.09)	(21.11, 21.17, 21.14)	2.820	(19.32, 20.43, 22.88)	0.257	(19.27, 20.30, 23.09)	0.0000000002
5	(0.95, 1.02, 1.16)	(35.54, 38.00, 41.92)	(17.05, 17.30, 19.86)	(16.77, 16.52, 16.21)	3.743	(15.10, 16.06, 18.32)	2.775	(17.05, 17.30, 19.86)	0.0000000002
6	(1.25, 1.29, 1.47)	(31.73, 32.00, 35.72)	(18.84, 20.40, 23.36)	(18.41, 18.48, 18.33)	5.399	(16.73, 18.06, 20.44)	4.300	(18.84, 20.40, 23.36)	0.0000000001
7	(1.39, 1.52, 1.69)	(27.91, 29.00, 32.47)	(17.90, 19.30, 21.89)	(19.56, 19.75, 19.62)	2.846	(17.94, 19.30, 21.69)	0.205	(17.90, 19.30, 21.89)	0.0000000001
8	(1.27, 1.33, 1.49)	(17.71, 18.00, 20.08)	(20.28, 21.90, 24.72)	(21.11, 21.24, 21.34)	3.541	(19.93, 21.27, 23.87)	1.116	(20.28, 21.90, 24.72)	0.0000000001
9	(1.66, 1.71, 1.95)	(36.56, 40.00, 45.36)	(14.37, 15.90, 17.54)	(18.76, 18.26, 17.99)	5.006	(16.69, 17.32, 19.55)	3.382	(14.37, 15.90, 17.54)	0.0000000001
10	(1.93, 2.00, 2.24)	(27.71, 28.00, 30.84)	(16.75, 18.30, 20.18)	(21.26, 21.42, 21.62)	5.667	(19.37, 20.70, 23.42)	4.808	(16.75, 18.30, 20.18)	0.0000000001
11	(1.53, 1.68, 1.85)	(12.52, 13.00, 14.92)	(20.98, 22.60, 25.58)	(22.89, 23.26, 23.42)	2.956	(21.77, 23.24, 25.88)	1.061	(20.98, 22.60, 25.58)	0.0000000003
12	(1.97, 2.15, 2.45)	(18.73, 19.00, 20.90)	(21.42, 23.70, 26.58)	(23.07, 23.58, 24.14)	2.949	(21.49, 23.07, 26.09)	0.797	(21.42, 23.70, 26.58)	0.0000000002
13	(3.31, 3.52, 3.92)	(29.36, 31.00, 35.13)	(24.06, 24.40, 27.36)	(25.19, 25.53, 25.98)	2.111	(22.52, 23.80, 26.69)	1.781	(24.06, 24.40, 27.36)	0.0000000002
14	(2.13, 2.33, 2.66)	(29.12, 31.00, 35.08)	(20.31, 21.80, 24.85)	(21.61, 21.87, 22.11)	3.028	(19.56, 20.85, 23.55)	1.772	(20.31, 21.80, 24.85)	0.0000000002
15	(1.55, 1.71, 1.90)	(15.80, 17.00, 19.24)	(22.35, 23.80, 26.41)	(22.33, 22.60, 22.76)	3.842	(21.08, 22.43, 25.07)	2.297	(22.35, 23.80, 26.41)	0.0000000003
16	(1.11, 1.14, 1.25)	(13.96, 14.00, 15.94)	(18.88, 20.80, 23.11)	(21.33, 21.41, 21.38)	3.055	(20.37, 21.69, 24.16)	2.028	(18.88, 20.80, 23.11)	0.0000000002
17	(0.90, 0.99, 1.09)	(17.92, 19.00, 20.96)	(17.48, 17.50, 20.08)	(19.93, 20.01, 19.94)	3.510	(18.94, 20.21, 22.68)	4.025	(17.48, 17.50, 20.08)	0.0000000002
18	(1.12, 1.14, 1.30)	(27.67, 28.00, 31.02)	(16.68, 17.80, 20.30)	(18.77, 18.77, 18.70)	2.811	(17.31, 18.58, 21.03)	1.236	(16.68, 17.80, 20.30)	0.0000000003
19	(1.37, 1.46, 1.66)	(23.79, 26.00, 28.66)	(19.47, 20.20, 22.33)	(20.27, 20.13, 20.25)	2.233	(18.82, 19.81, 22.44)	0.764	(19.47, 20.20, 22.33)	0.0000000003
20	(1.75, 1.81, 2.04)	(30.53, 32.00, 35.76)	(18.87, 20.00, 22.63)	(20.17, 20.08, 20.08)	2.866	(18.27, 19.35, 21.85)	1.177	(18.87, 20.00, 22.63)	0.0000000001
21	(1.31, 1.38, 1.52)	(9.50, 10.00, 11.39)	(21.41, 22.80, 25.08)	(22.78, 22.90, 23.07)	2.434	(21.88, 23.17, 25.82)	0.954	(21.41, 22.80, 25.08)	0.0000000002
22	(0.77, 0.84, 0.95)	(35.09, 38.00, 42.18)	(17.50, 19.10, 21.22)	(16.30, 15.96, 15.52)	6.619	(14.74, 15.62, 17.74)	5.644	(17.50, 19.10, 21.22)	0.0000000001
23	(1.41, 1.48, 1.64)	(48.04, 49.00, 55.47)	(11.01, 12.10, 13.83)	(15.83, 15.86, 15.13)	6.248	(13.46, 14.76, 16.59)	4.549	(11.01, 12.10, 13.83)	0.0000000002
24	(1.04, 1.08, 1.24)	(40.86, 42.00, 47.52)	(11.92, 12.80, 14.70)	(16.04, 15.95, 15.40)	5.234	(14.13, 15.32, 17.31)	4.251	(11.92, 12.80, 14.70)	0.0000000002
Average error				3.692		2.234		0.0000000002	
Training time (s)				0.461		0.436		0.031	

Table 4 Hsien–Chung Wu dataset [29]

	Data			FLR_1		FLR_2		FR_{ELM}	
	$\tilde{X}_1$	$\tilde{X}_2$	$\tilde{Y}$	Predicted value	Error	Predicted value	Error	Predicted value	Error
1	(151, 274, 322)	(1432, 2450, 3461)	(111, 162, 194)	(103.01, 167.42, 194.93)	9.700	(109.34, 161.90, 178.75)	15.340	(111.00, 162.00, 194.00)	0.000093
2	(101, 180, 291)	(2448, 3254, 4463)	(88, 120, 161)	(82.35, 124.31, 183.56)	23.648	(91.16, 122.67, 183.78)	23.154	(88.00, 120.00, 161.00)	0.000084
3	(221, 375, 539)	(2592, 3802, 5116)	(161, 223, 288)	(141.91, 222.31, 308.02)	27.673	(152.15, 224.43, 313.32)	26.861	(161.00, 223.00, 288.00)	0.000096
4	(128, 205, 313)	(1414, 2838, 3252)	(83, 131, 194)	(91.63, 134.99, 189.70)	10.440	(97.72, 131.24, 170.42)	27.799	(83.00, 131.00, 194.00)	0.000095
5	(62, 86, 112)	(1024, 2347, 3766)	(51, 67, 83)	(57.67, 74.58, 92.85)	14.106	(62.09, 67.70, 85.86)	11.475	(51.00, 67.00, 83.00)	0.000084
6	(132, 265, 362)	(2163, 3782, 5091)	(124, 169, 213)	(96.49, 168.14, 220.89)	28.631	(104.75, 169.68, 229.51)	25.363	(124.00, 169.00, 213.00)	0.000090
7	(66, 98, 152)	(1687, 3008, 4325)	(62, 81, 102)	(62.20, 83.04, 114.67)	12.836	(68.55, 79.73, 115.64)	15.184	(62.00, 81.00, 102.00)	0.000080
8	(151, 330, 463)	(1524, 2450, 3864)	(138, 192, 241)	(103.37, 194.96, 265.82)	42.710	(109.96, 189.67, 253.02)	30.595	(138.00, 192.00, 241.00)	0.000087
9	(115, 195, 291)	(1216, 2137, 3161)	(82, 116, 159)	(84.48, 127.37, 178.53)	22.733	(89.89, 119.83, 158.28)	8.799	(82.00, 116.00, 159.00)	0.000096
10	(35, 53, 71)	(1432, 2560, 3782)	(41, 55, 71)	(45.97, 59.18, 72.75)	6.725	(51.33, 53.29, 66.87)	11.253	(41.00, 55.00, 71.00)	0.000081
11	(307, 430, 584)	(2592, 4020, 5562)	(168, 252, 367)	(184.20, 250.20, 331.87)	38.724	(195.16, 253.72, 343.24)	36.127	(168.00, 252.00, 367.00)	0.000093
12	(284, 372, 498)	(2792, 4427, 6163)	(178, 232, 346)	(173.66, 223.25, 291.91)	54.970	(185.00, 228.69, 314.53)	32.413	(178.00, 232.00, 346.00)	0.000094
13	(121, 236, 370)	(1734, 2660, 4094)	(111, 144, 198)	(89.43, 149.55, 220.98)	32.002	(96.37, 144.98, 213.74)	21.515	(111.00, 144.00, 198.00)	0.000097
14	(103, 157, 211)	(1426, 2088, 3312)	(78, 103, 148)	(79.39, 108.50, 139.77)	9.990	(85.30, 100.53, 123.58)	25.609	(78.00, 103.00, 148.00)	0.000089
15	(216, 370, 516)	(1785, 2605, 4042)	(167, 212, 267)	(136.34, 215.23, 292.57)	40.054	(144.22, 210.94, 281.46)	26.997	(167.00, 212.00, 267.00)	0.000083
Average error				24.996		22.566		0.000089	
Training time (s)				0.622		0.552		0.027	

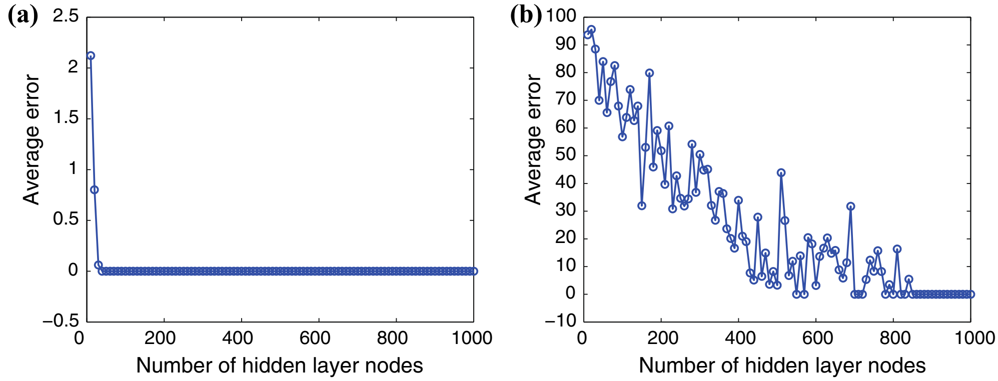

Fig. 3 Impact of number of hidden layer nodes on performance of FR_{ELM}. **a** CEC-OM-SAND dataset [23], **b** Hsien–Chung Wu dataset [29]

triangular fuzzy input and fuzzy output data. Our future works include the following two topics. Firstly, some practical applications will be sought for FR_{ELM}. Secondly, we try to construct ELM-based fuzzy regression model to handle the more general types of fuzzy data.

References

1. Arabpour AR, Tata M (2008) Estimating the parameters of a fuzzy linear regression model. Iran J Fuzzy Syst 5(2):1–19
2. Berenji HR (1992) Fuzzy logic controllers. An introduction to fuzzy logic applications in intelligent systems. Springer, New York
3. Choi SH, Buckley JJ (2008) Fuzzy regression using least absolute deviation estimators. Soft Comput 12(3):257–263
4. Diamond P (1988) Fuzzy least squares. Inf Sci 46(3):141–157
5. Diamond P, Körner R (1997) Extended fuzzy linear models and least squares estimates. Comput Math Appl 33(9):15–32
6. D'Urso P (2003) Linear regression analysis for fuzzy/crisp input and fuzzy/crisp output data. Comput Stat Data Anal 42(1):47–72
7. Fu AM, Dong CR, Wang LS (2014) An experimental study on stability and generalization of extreme learning machines. Int J Mach Learn Cybern. doi:10.1007/s13042-014-0238-0
8. Fu AM, Wang XZ, He YL, Wang LS (2014) A study on residence error of training an extreme learning machine and its application to evolutionary algorithms. Neurocomputing. doi:10.1016/j.neucom.2014.04.067
9. Gani AN (2012) A new operation on triangular fuzzy number for solving fuzzy linear programming problem. Appl Math Sci 6(11):525–532
10. Hassanpour H, Maleki HR, Yaghoobi MA (2009) A goal programming approach to fuzzy linear regression with non-fuzzy input and fuzzy output data. Asia–Pac J Oper Res 26(5):587–604
11. Hasanpour H, Maleki HR, Yaghoubi MA (2010) Fuzzy linear regression model with crisp coefficients: a goal programming approach. Iran J Fuzzy Syst 7(2):19–39
12. Hojati M, Bector CR, Smimou K (2005) A simple method for computation of fuzzy linear regression. Eur J Oper Res 166(1):172–184
13. Hong DH, Lee S, Do HY (2001) Fuzzy linear regression analysis for fuzzy input–output data using shape-preserving operations. Fuzzy Sets Syst 122(3):513–526
14. Huang GB, Chen L, Siew CK (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17(4):879–892
15. Huang GB, Wang DH, Lan Y (2011) Extreme learning machines: a survey. Int J Mach Learn Cybern 2(2):107–122
16. Huang L, Zhang BL, Huang Q (1998) Robust interval regression analysis using neural networks. Fuzzy Sets Syst 97(3):337–347
17. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501
18. Ishibuchi H, Tanaka H (1992) Fuzzy regression analysis using neural networks. Fuzzy Sets Syst 50(3):257–265
19. Kao C, Chyu CL (2002) A fuzzy linear regression model with better explanatory power. Fuzzy Sets Syst 126(3):401–409
20. Kao C, Chyu CL (2003) Least-squares estimates in fuzzy regression analysis. Eur J Oper Res 148(2):426–435
21. Kwang HL (2005) First course on fuzzy theory and applications. Adv Soft Comput 27:129–151
22. Pedrycz W (1990) Fuzzy sets in pattern recognition: methodology and methods. Pattern Recognit 23(1):121–146
23. Rabiei MR, Arghami NR, Taheri SM, Sadeghpour B (2013) Fuzzy regression model with interval-valued fuzzy input-output data. In: Proceedings of 2013 IEEE international conference on fuzzy systems, pp 1–7
24. Redden DT, Woodall WH (1994) Properties of certain fuzzy linear regression methods. Fuzzy Sets Syst 64:361–375
25. Sakawa M, Yano H (1992) Fuzzy linear regression analysis for fuzzy input-output data. Inf Sci 63(3):191–206
26. Sakawa M, Yano H (1992) Multiobjective fuzzy linear regression analysis for fuzzy input-output data. Fuzzy Sets Syst 47(2):173–181
27. Tanaka H, Uejima S, Asai K (1982) Linear regression analysis with fuzzy model. IEEE Trans Syst Man Cybern 12(6):903–907
28. Wang XZ, Shao QY, Miao Q, Zhai JH (2013) Architecture selection for networks trained with extreme learning machine using localized generalization error model. Neurocomputing 102:3–9
29. Wu HC (2003) Linear regression analysis for fuzzy input and output data using the extension principle. Comput Math Appl 45(12):1849–1859
30. Yang MS, Ko CH (1996) On a class of fuzzy *c*-numbers clustering procedures for fuzzy data. Fuzzy Sets Syst 84(1):49–60
31. Yang MS, Lin TS (2002) Fuzzy least-squares linear regression analysis for fuzzy input-output data. Fuzzy Sets Syst 126(3):389–399
32. Yen KK, Ghoshray S, Roig G (1999) A linear regression model using triangular fuzzy number coefficients. Fuzzy Sets Syst 106(2):167–177
33. Zhang AW (2012) Contrast between fuzzy linear least-squares regression based on different distances. J Jiangsu Univ Sci Technol 26(5):509–513