18th International Conference on Information Fusion
Washington, DC - July 6-9, 2015

Fuzzy Extreme Learning Machine and Its Applications

Wenbo Zhang
School of Electronic Engineering
Xidian University
Xi'an, China
zwbsoul@163.com

Hongbing Ji
School of Electronic Engineering
Xidian University
Xi'an, China
hbji@xidian.edu.cn

Abstract - Compared to traditional classifiers, such as support vector machine, extreme learning machine (ELM) achieves similar performance for classification and runs at much faster learning speed. However, in many real applications, the different samples may not be exactly assigned to one of the classes such as the imbalance data problems and the cost-sensitive learning problems. The traditional ELM lacks the ability to solve those problems. We proposed an extension method of ELM called fuzzy ELM (FELM) which introduces a set of fuzzy memberships to the traditional ELM method. Then, the inputs with different fuzzy memberships can make different contributions to the learning of the output weights. Moreover, the fuzzy memberships can be conveniently determined based on classes or examples. For the imbalanced problems, the cost-sensitive learning, or the noise signal problems, FELM can provide a more logical result than that of ELM, implying good application prospects for the real world applications.

Keywords: Extreme Learning Machine (ELM), Fuzzy Extreme Learning Machine (FELM), fuzzy membership, fuzzy matrix, cost-sensitive learning, radar emitter recognition.

1 Introduction

Extreme learning machine (ELM) [1-4] was originally proposed for the single-hidden-layer feedforward neural networks (SLFNs) and then extended to the generalized SLFNs where the hidden layer need not be neuron alike. In ELM, the input weights of the SLFNs are randomly chosen without iterative tuning, and the output weights are analytically determined. Thus, the training speed of ELM can be thousand times faster than that of the traditional iterative implementations of SLFNs. In addition, different from the traditional learning algorithms for a neural type of SLFNs, ELM aims to reach not only the smallest training error but also the smallest norm of output weights. Bartlett's theory [5] shows that for feedforward neural networks reaching smaller training error the smaller the norm of weights is, the better generalization performance the networks tend to have. Because of its good performance, ELM has been attracting the attentions from more and more researchers [6-12]. To solve the classification problems using ELM, Liu et al. [13] propose that ELM can be applied to support vector machines (SVMs) by simply replacing SVM kernels with ELM kernels. Huang et al. study ELM for classification with the standard optimization method [14] and verify that ELM can solve any multiclass classification problems directly [15].

However, in many real applications, the different input points may not be exactly assigned to one of the classes such as the imbalance problems and the cost-sensitive learning problems. The traditional ELM lacks this kind of ability. This paper proposes a novel method called fuzzy ELM (FELM) where a fuzzy membership is applied to each input of ELM such that different inputs can make different contributions to the learning of output weights. Moreover, the fuzzy memberships can be appropriately defined depending on the different real classification applications, such as weighted classification, the problem of noises, or cost-sensitive learning [16].

In cost-sensitive learning systems, different misclassification errors incur different penalties. For example, in medical applications, the cost of "misrecognizing a healthy human as a patient" and that of "misrecognizing a patient as a healthy human" should be different. Compared with the first error, the second one is more serious since it would delay the treatment of patient. The purpose of cost-sensitive learning is to minimize total cost rather than total error [17]. Therefore, it can be applied to many real classification problems [18-22]. In this paper, cost-sensitive learning is achieved by FELM through defining the appropriate fuzzy memberships which reasonably reflect the misclassifying cost.

The remainder of the paper is organized as follows. Section 2 briefly summarizes the principles of ELM, and discusses the cost-sensitive learning problem. The proposed algorithm is described in detail in Section 3, including the description of the FELM algorithm, and the determination of the fuzzy memberships. In Section 4, the experiments and results analysis are presented. The conclusions are drawn in Section 5.

2 Related work

Fuzzy extreme learning machine (FELM) is based on the traditional ELM. This section briefly reviews the ELM. One key principle of the ELM is that the input weights are randomly chosen without iterative tuning and the output weights are analytically determined. Moreover, some real

classification problems which cannot be solved by traditional ELM, such as imbalance problem and cost-sensitive learning, are discussed.

2.1 Extreme learning machine

ELM [17] was originally proposed for the single-hidden layer feedforward neural networks and was then extended to the "generalized" single-hidden layer feedforward networks (SLFNs) where the hidden layer need not be neuron alike [3]. The output of an ELM with $\tilde{l}$ hidden nodes can be represented by

$$f_{\tilde{l}}(\mathbf{x}) = \sum_{i=1}^{\tilde{l}} \boldsymbol{\beta}_i G(\mathbf{a}_i, b_i, \mathbf{x}), \qquad \mathbf{a}_i \in \mathbf{R}^n, \mathbf{x} \in \mathbf{R}^n \quad (1)$$

where $\mathbf{a}_i$ and b_i are the learning parameters of hidden nodes, $\boldsymbol{\beta}_i$ is the weight connecting the ith hidden node to the output node, and $G(\mathbf{a}_i, b_i, \mathbf{x})$ is the output of the ith hidden node with respect to the input $\mathbf{x}$. For N arbitrary distinct samples $(\mathbf{x}_k, \mathbf{t}_k)$, if ELM can classify them accurately, it implies that there exist $\mathbf{a}_i, b_i$ and $\boldsymbol{\beta}_i$ such that

$$\sum_{i=1}^{\tilde{l}} \boldsymbol{\beta}_i G(\mathbf{a}_i, b_i, \mathbf{x}) = \mathbf{t}_k, \qquad k = 1, ..., N. \quad (2)$$

Eq. (2) can be written compactly as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T}, \quad (3)$$

where $\mathbf{T} = \left[t_1, t_2, \ldots, t_N\right]^T$. $\mathbf{H}$ is called the hidden layer output matrix of the network, and the parameters $(\mathbf{a}_i, b_i)$ of $\mathbf{H}$ are randomly chosen. Then, the classification problem for ELM can be formulated as

$$\begin{aligned} &\text{Minimize: } L_{ELM} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\frac{1}{2}\sum_{i=1}^{N}\|\boldsymbol{\xi}_i\|^2 \\ &\text{Subject to: } \mathbf{H}(\mathbf{x}_i)\boldsymbol{\beta} = \mathbf{t}_i^T - \boldsymbol{\xi}_i^T, \quad i = 1, ..., N \end{aligned} \quad (4)$$

where $\boldsymbol{\xi}_i$ is the training error vector for the training sample $\mathbf{x}_i$, and C is the regularization parameter which represents the trade-off between the minimization of training errors and the maximization of the marginal distance. According to Karush-Kuhn-Tucker (KKT) theorem [23], to train ELM is equivalent to solving the following dual optimization problem:

$$\begin{aligned} L_{ELM} = &\frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\frac{1}{2}\sum_{i=1}^{N}\|\boldsymbol{\xi}_i\|^2 \\ &- \sum_{i=1}^{N}\sum_{j=1}^{m}\alpha_{i,j}\left(\mathbf{h}(\mathbf{x}_i)\beta_j - t_{i,j} + \xi_{i,j}\right) \end{aligned}. \quad (5)$$

The KKT corresponding optimality conditions can be obtained as:

$$\begin{aligned} &\frac{\partial L_{ELM}}{\partial \beta_j} = 0 \rightarrow \beta_j = \sum_{i=1}^{N}\alpha_{i,j}\mathbf{h}(x_i)^T \rightarrow \boldsymbol{\beta} = \mathbf{H}^{\mathbf{T}}\boldsymbol{\alpha} \\ &\frac{\partial L_{ELM}}{\partial \boldsymbol{\xi}_i} = 0 \rightarrow \boldsymbol{\alpha}_i = C\boldsymbol{\xi}_i, \qquad i = 1, ..., N \\ &\frac{\partial L_{ELM}}{\partial \boldsymbol{\alpha}_i} = 0 \rightarrow \mathbf{h}(x_i)\beta - \mathbf{t}_i^T + \boldsymbol{\xi}_i^T = 0, \qquad i = 1, ..., N \end{aligned} \quad (6)$$

where $\boldsymbol{\alpha} = \left[\alpha_1, \ldots \alpha_N\right]^T$. From (6), we have

$$\boldsymbol{\beta} = \mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T}. \quad (7)$$

Then, the output function of ELM classifier is

$$\mathbf{f}(x) = \mathbf{h}(x)\boldsymbol{\beta} = \mathbf{h}(x)\mathbf{H}^T\left(\frac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T}. \quad (8)$$

2.2 Imbalanced data problem and cost-sensitive learning

The classification application with data sets exhibiting an unequal distribution between its classes can be considered as imbalanced data problem [24], which often exists in the real-word application. For example, a data set contains 990 "Negative" (majority class) samples and 10 "Positive" (minority class) samples. However, most traditional algorithms expect that the class distributions are balanced. Therefore, when presented with complex imbalanced data sets, these algorithms fail to properly represent the distributive characteristics of the data and resultantly provide unfavorable accuracies for the minority classes. Moreover, the traditional evaluation metrics for classification problems, such as accuracy or error rate, cannot provide comprehensive assessments of imbalanced data problems. Therefore, two important metrics are introduced to evaluate the performance of imbalanced learning method:

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$G - mean = \sqrt{\frac{TP}{TP + FN} \times \frac{TN}{TN + FP}}, \quad (10)$$

where *TP*, *TN*, *FP*, *FN* stand for true positive, true negative, false positive and false negative, respectively.

Currently, there are mainly two ideas to solve the imbalanced data problem. Firstly, Sampling methods modify the imbalanced data set by some mechanisms in order to provide a balanced distribution, e.g. the oversampling method adds data for the minority class and the undersampling method removes data form the majority class. Secondly, cost-sensitive learning targets the imbalanced learning problem by using different cost matrices that describe the costs for misclassifying any particular data example. Moreover, in some applications, cost-sensitive learning is superior to sampling methods [24].

Fundamental to the cost-sensitive learning methodology is the concept of the cost matrix $\mathbf{C}$ which describes the misclassification costs. $\mathbf{C}(i, j)$ is the cost of predicting that an example belongs to class i when in fact it belongs to class j. In a binary classification application, the cost matrix $\mathbf{C}(Min, Maj)$ can be defined as the cost of misclassifying a majority class example as a minority class example and let $\mathbf{C}(Maj, Min)$ represents the cost of the contrary case. Then, to solve the imbalanced problem, the cost of misclassifying minority examples can be determined higher than the contrary case, namely $\mathbf{C}(Maj, Min) > \mathbf{C}(Min, Maj)$. The objective of cost-sensitive learning then is to develop a hypothesis that minimizes total cost on the training data set rather than total error. In addition, via changing the cost matrix, cost-sensitive learning can be applied to many real-word fields, such as military objects recognition.

3 Fuzzy ELM

As the description above, the traditional ELM is a competitive learning method, which achieves excellent performance both in accuracy rate and run time. However, in many real applications, the different input points may not be exactly assigned to one of the classes such as the imbalance problems and the weighted classification problems. The traditional ELM lacks the ability to solve those problems. In this paper, we propose the fuzzy ELM which introduces a set of fuzzy memberships and a fuzzy matrix to the traditional ELM. Then, the inputs with different fuzzy memberships can make different contributions to the learning of the output weights $\boldsymbol{\beta}$. As a result, the fuzzy ELM can solve the problems mentioned above.

3.1 Proposed fuzzy ELM

According to the different weights in the real world classification problems, the effects of the training points should be different. Inspired by the fuzzy support vector machine [25], a set s_i of labeled training points with associated fuzzy membership are introduced

$$(\mathbf{x}_1, t_1, s_1), ..., (\mathbf{x}_N, t_N, s_N). \quad (11)$$

Each training point $\mathbf{x}_i$ is given a label t_i and a fuzzy membership s_i, $0 < s_i \le 1$. The fuzzy membership s_i is the attitude of the corresponding point $\mathbf{x}_i$ toward one class and $\frac{1}{2}\|\xi_i\|$ is a measure of error in ELM. Thus, $\frac{1}{2} s_i \|\xi_i\|$ is a measure of error with different weight s_i. The classification problem for the constrained-optimal-based fuzzy ELM can be formulated as

$$\text{Minimize: } L_{FELM} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\frac{1}{2}\sum_{i=1}^{N} s_i \|\boldsymbol{\xi}_i\|^2$$

$$\text{Subject to: } \mathbf{h}(x_i)\boldsymbol{\beta} = \mathbf{t}_i^T - \boldsymbol{\xi}_i^T, \quad i = 1, ..., N. \quad (12)$$

Based on the KKT theorem, to train fuzzy ELM is equivalent to solving the following dual optimization problem

$$L_{FELM} = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\frac{1}{2}\sum_{i=1}^{N} s_i \|\boldsymbol{\xi}_i\|^2 - \sum_{i=1}^{N}\sum_{j=1}^{m} \alpha_{i,j}(\mathbf{h}(x_i)\beta_j - t_{i,j} + \xi_{i,j}), \quad (13)$$

where α_i is the Lagrange multiplier corresponding to the ith training sample. We can have the KKT corresponding optimality conditions as follows

$$\frac{\partial L_{FELM}}{\partial \beta_j} = 0 \rightarrow \beta_j = \sum_{i=1}^{N} \alpha_{i,j}\mathbf{h}(x_i)^T \rightarrow \boldsymbol{\beta} = \mathbf{H}^T\boldsymbol{\alpha}$$

$$\frac{\partial L_{FELM}}{\partial \boldsymbol{\xi}_i} = 0 \rightarrow \boldsymbol{\alpha}_i = Cs_i\boldsymbol{\xi}_i, \quad i = 1, ..., N \quad (14)$$

$$\frac{\partial L_{FELM}}{\partial \boldsymbol{\alpha}_i} = 0 \rightarrow \mathbf{h}(x_i)\boldsymbol{\beta} - \mathbf{t}_i^T + \boldsymbol{\xi}_i^T = 0, \quad i = 1, ..., N$$

The following equation can be obtained from (14)

$$\left(\frac{\mathbf{S}}{C} + \mathbf{H}\mathbf{H}^T\right)\boldsymbol{\alpha} = \mathbf{T}, \quad (15)$$

where $\mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix}$ and the fuzzy matrix

$$\mathbf{S} = \begin{bmatrix} \frac{1}{s_1} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{s_2} & 0 & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & \frac{1}{s_N} \end{bmatrix}_{N\times N}.$$

From (14) and (15),

$$\beta = \mathbf{H}^T\left(\frac{\mathbf{S}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T}. \quad (16)$$

Therefore, the inputs with different fuzzy memberships can make different contributions to the learning of the output weights β. Then, the output function of FELM classifier is

$$\mathbf{f}(x) = \mathbf{h}(x)\beta = \mathbf{h}(x)\mathbf{H}^T\left(\frac{\mathbf{S}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T}. \quad (17)$$

For binary classification problem, FELM needs only one output node, and the decision function is

$$f(\mathbf{x}) = sign\left(\mathbf{h}(x)\mathbf{H}^T \left(\frac{\mathbf{S}}{C} + \mathbf{H}\mathbf{H}^T \right)^{-1} \mathbf{T} \right). \quad (18)$$

For multiclass cases, the predicted class label of a testing point is the index number of the output node which has the highest output value for the given testing sample

$$label(\mathbf{x}) = \arg\max_{j\in\{1,...,m\}} f_j(\mathbf{x}). \quad (19)$$

Obviously, when $s_1 = s_2 = ... = s_N = 1$, FELM will be the traditional ELM. Thus, FELM is an extension of ELM.

3.2 Determining the fuzzy memberships

In FELM, choosing the appropriate fuzzy memberships according to the real-world applications is very important. There are mainly two kinds of methods to determine the fuzzy memberships, i.e., class-dependent and example-dependent.

3.2.1 Class-dependent determination of the fuzzy memberships

In most applications, we want to improve the accuracy of classifying one class or certain classes, such as the minority class in the imbalanced data problem. Then we can define the fuzzy memberships s_i for each class. The minority class will be defined a big membership, while the membership corresponding to the majority class will be small. The specific value of the fuzzy memberships can be determined directly according to the prior knowledge or our demands. For example, to solve the imbalanced data problem, we can define the fuzzy memberships as $(\mathbf{x}_i, +1, 0.9)$, $(\mathbf{x}_i, -1, 0.1)$. Furthermore, the fuzzy memberships can be generated automatically as

$$s_i = 1 / n_{t_i}, \quad (20)$$

where n_{t_i} is the number of samples belonging to the class t_i.

To solve the cost-sensitive problem, the fuzzy membership can be determined according to the cost matrix. In the medical applications mentioned above, to reduce the occurrence probability of "misrecognizing a patient as a healthy human", we can define the fuzzy memberships as $(\mathbf{x}, t_i, \mathbf{C}(j,i))$, $(\mathbf{x}, t_j, \mathbf{C}(i,j))$, where t_i is the label of the healthy human and t_j is the label of the patient. $\mathbf{C}(i,j)$ is the cost of predicting that an example belongs to the healthy human when in fact it belongs to the patient. When $\mathbf{C}(i,j) > \mathbf{C}(j,i)$, the classifying accuracy of the patient will be higher than that of the healthy human.

3.2.2 Example-dependent determination of the fuzzy memberships

In other applications, the fuzzy memberships should be defined for every sample. Then, the fuzzy memberships correspond to each example instead of each class. For example, in military objects recognition problems, the useful objects often mixes with the noise signal. Then, the traditional classifiers including the ELM are very sensitive to noises, so the classification performance will be reduced. The example-dependent FELM can solve this problem. Suppose we are given a sequence of training points $(\mathbf{x}_1, t_1, s_1), ..., (\mathbf{x}_N, t_N, s_N)$. Denote the mean of the examples belong to class j as $\bar{\mathbf{x}}_j$. Let the radius of class j

$$r_j = \max_{\{\mathbf{x}_i | t_i = j\}} \left| \bar{\mathbf{x}}_j - \mathbf{x}_i \right|. \quad (21)$$

The fuzzy membership s_i can be determined as a function of the mean and radius of each class

$$s_i = 1 - \left| \bar{\mathbf{x}}_j - \mathbf{x}_i \right| / \left(r_j + \varepsilon \right) \quad if\ t_i = j, \quad (22)$$

where sufficient small $\varepsilon > 0$ is used to avoid the case $s_i = 0$. Then, the noise samples farther from the mean of class will be assigned smaller fuzzy memberships. Therefore, the example-dependent FELM can be applied to reduce the effects of noises.

In many real-world problems, it is important to predict values of time series, namely to predict the value of the time series at the moment $t + \Delta$ according to the values at the moment t. Many researchers consider the time series introduced by Mackey and Glass which is solution of the equation

$$\frac{dx(t)}{dt} = -ax(t) + \frac{bx(t-\tau)}{1 + x^{10}(t-\tau)}. \quad (23)$$

where a, b and τ are parameters of the equation. For example, to predict if $x(t+\Delta) > x(t)$, we can use three dimensional vector of observations on time series

$$x_t = \left(x(t-2), x(t-1), x(t) \right). \quad (24)$$

According to the expert experience, the data from recent past is given more weighting than the data far back in the past. Thus, the fuzzy memberships can be generated as

$$s_i = \frac{1-\sigma}{t_l - t_1} t_i + \frac{t_l \sigma - t_1}{t_l - t_1}, \quad (25)$$

where $t_1 < t_2 < \cdots$ is the time the data arrived in the system, and σ is the lower bound of fuzzy memberships.

In addition, the fuzzy memberships can be determined according to the prior knowledge such as priori probability of each example. However, it has not been experimentally verified in this paper, and worthy of further study.

4 Experimental results

In the following, three methods are compared using several classification datasets. The compared methods are SVM, the traditional ELM, and the proposed FELM. The datasets are collected form the UCI Machine Learning Repository [26] and a set of radar emitter datasets [27]. In this paper, SVM, the traditional ELM and FELM are used with Gaussian kernel function $K(\mathbf{u},\mathbf{v})=\exp(-\gamma\|\mathbf{u}-\mathbf{v}\|^2)$, which is the popular choice in the classification problems. In order to achieve good generalization performance, the trade-off constant C and the kernel parameter γ need to be chosen appropriately. C and γ are searched in the range of $\{2^{-18},2^{-17},...,2^{24},2^{25}\}$, respectively.

4.1 Benchmark Datasets

Table 1
Specification of classification problems from UCI.

Datasets	train	test	classes	features
Glass	144	70	6	9
Iris	100	50	3	4
Leukemia	38	34	2	7129
Mushroom	1500	6624	2	22
Segment	1540	770	7	19
Vehicle	564	282	4	18
Wine	118	60	3	13

In order to verify the performance of FELM, wide types of balanced data sets, as shown in Table 1, are tested in our simulations, which are of binary class data, multiclass data, small sizes, low dimensions, large sizes, or high dimensions. In this experiment, determination of the fuzzy membership is not the focal point, so the fuzzy memberships are set randomly as: **Glass** (0.1, 0.1, 0.2, 0.1, 0.2, 0.1), **Iris** (0.35, 0.4, 0.45), **Leukemia** (0.8, 0.9), **Mushroom** (0.22, 0.18), **Segment** (0.35, 0.40, 0.35, 0.40, 0.35, 0.40, 0.35), **Vehicle** (0.3, 0.3, 0.4, 0.4), **Wine** (0.3, 0.3, 0.2). Form Table 2, it can be seen that FELM can always achieve comparable accuracy as SVM and ELM. Moreover, the learning speed of ELM and FELM are much faster than that of SVM. Then, we change the fuzzy memberships of two datasets to: **Iris** (0.5, 0.3, 0.2), **Glass** (0.3, 0.2, 0.2, 0.1, 0.1, 0.1). As shown in Table 3 and Table 4, in order to verify FELM having the ability to solve the weighted problems, we calculate the detailed result of each class respectively. In FELM, compared with the results of ELM, the classes with larger fuzzy memberships have higher accuracy. In other words, the separate boundaries of classifier are moved towards the classes with smaller fuzzy memberships. Therefore, the accuracies of classes with different fuzzy memberships are changed, which is the purpose of proposing FELM.

Table 2
Performance comparison of SVM, ELM and FELM.

Datasets	SVM		ELM		FELM	
	Rate (%)	Time (s)	Rate (%)	Time (s)	Rate (%)	Time (s)
Glass	67.83	0.279	67.14	0.025	67.14	0.027
Iris	95.12	0.071	94.64	0.013	96.39	0.015
Leukemia	82.34	0.993	82.35	0.029	82.32	0.030
Mushroom	89.88	35.882	88.84	1.520	87.93	1.531
Segment	96.53	13.901	96.07	1.799	96.19	1.806
Vehicle	84.37	1.470	83.48	0.225	83.39	0.258
Wine	98.37	0.071	98.47	0.019	98.41	0.020

Table 3
The accuracy of each label in the dataset **Iris**.

Label	1	2	3
ELM	94.82	94.05	94.59
FELM	99.31	96.67	93.18

Table 4
The accuracy of each label in the dataset **Glass**.

Label	1	2	3	4	5	6
ELM	66.85	67.24	67.18	67.12	66.90	67.03
FELM	78.64	72.52	71.48	61.55	60.93	61.21

4.2 Imbalanced data problem

As shown in Table 5, the datasets **adult** and **banana** are chosen in this experiment which is used to demonstrate the performance of the proposed algorithm when the datasets are imbalanced. The fuzzy memberships can be generated by (20), and if the users want to enlarge the effect of FELM, the fuzzy membership can be determined as

$$s_i=\left(1/n_{t_i}\right)^2. \tag{26}$$

In this experiment, we evaluate the performance of FELM in terms of accuracy, precision and G-mean. In addition, in order to explain the experimental results conveniently, precision and G-mean are calculated as

$$Precision=\frac{Tm}{Tm+Fm} \tag{27}$$

$$G-mean=\sqrt{\frac{Tm}{Tm+FM}\times\frac{TM}{TM+Fm}} \tag{28}$$

where *Tm*, *TM*, *Fm*, *FM* stand for true minority, true Majority, false minority and false Majority, respectively. Seen from Table 6, the accuracies of SVM, ELM and FELM are quite similar. However, in FELM, precision and G-mean are increased, and these two metrics can evaluate the imbalanced data problems more comprehensively. Moreover, we can also summarize that the higher the imbalance degree is, the more visibly precision and G-mean are increased, which is consistent with the principle of FELM.

Table 5
Specification of binary classification problems from UCI.

Datasets	Train	Test	Features	Imbalance ratio
Adult	4781	27780	123	0.3306
Banana	400	4900	2	0.8605

Table 6
Performance result of imbalanced data problems.

	Adult			Banana		
	Accuracy	Precision	G-mean	Accuracy	Precision	G-mean
SVM	84.51	70.53	72.28	89.84	87.11	89.93
ELM	84.58	70.36	72.19	89.83	87.02	89.28
FELM	84.42	84.49	79.96	89.82	90.61	90.42

4.3 Cost-sensitive learning

To verify FELM have the ability to solve cost-sensitive learning problems, a set of radar emitter datasets [27] are taken to the simulation. The radar pulse signal is single frequency, and the signal/noise ratio (SNR) is in the range of 15 ~ 25dB. Each category contains 100 radar emitter pulse signals, and 50 samples of that are selected for training. Firstly, we consider a binary classification problem. The cost matrix is defined as $\mathbf{C} = \begin{bmatrix} 0 & 0.8 \\ 0.2 & 0 \end{bmatrix}$, where $\mathbf{C}(i, j)$ is the cost of predicting that an example belongs to class i when in fact it belongs to class j. Therefore, the fuzzy memberships are set as (0.2, 0.8). In the testing phase, the total cost will add 0.2 cumulatively when the example belongs to class *1* is misclassified. Similarly, if the example belongs to class *2* is misclassified, the total cost will add 0.8 cumulatively. Then, we consider a 4-classes classification problem and the cost matrix is defined as $\mathbf{C} = \begin{bmatrix} 0 & 0 & 0.6 & 0.9 \\ 0 & 0 & 0.6 & 0.9 \\ 0.15 & 0.15 & 0 & 0.45 \\ 0.1 & 0.1 & 0.2 & 0 \end{bmatrix}$. According to the relationship of $\mathbf{C}(i, j)$, the fuzzy memberships are set as (0.15, 0.15, 0.25, 0.45). As shown in Table 7, the accuracies of ELM and FELM for the binary classification or the 4-classification are quite similar. However, in FELM, total cost is reduced obviously, which verifies its ability to solve cost-sensitive learning problems.

Table 7
Total cost comparison of ELM and FELM.

	Binary classification		4-classes classification	
	Rate (%)	Total cost	Rate (%)	Total cost
ELM	72.00	14.6	70.50	20.2
FELM	74.00	8.5	69.00	10.3

4.4 Noise signal problems

This experiment is used to demonstrate the performance of the proposed algorithm as the valid signal is mixed by noise. We selected eight categories of radar emitter signals, and experiment binary classification, 4-classes classification and 8-classes classification, respectively. The fuzzy memberships are generated by (21), (22). In Table 8, we can observe that the classification accuracy of FELM is higher than that of ELM. This determination of the fuzzy memberships could not be the optimal solution to solve the noise signal problem. How to determine a better set of fuzzy memberships is worthy of study.

Table 8
Performance result of classification problems with noises.

Rate (%)	Binary classification	4-classes classification	8-classes classification
ELM	69.00	67.50	62.75
FELM	71.00	69.00	65.25

4.5 Mackey-Glass time series datasets

We use the Mackey-Glass series datasets with parameters $a = 0.1$, $b = 0.2$ and $\tau = 17$, which are the usual parameters for experimental studies. Then, we consider three different problems, one step ($\Delta = 1$), five steps ($\Delta = 5$) and eight steps ($\Delta = 8$) predictions. In ELM, the time information is ignored in the training stage. In FELM, the fuzzy memberships are generated by (25). Form Table 2, it can be seen that the example-dependent fuzzy memberships are helpful for improving classification performance.

Table 9
Performance result of Mackey-Glass time series datasets.

Steps	1	5	8
ELM	91.03	89.87	88.42
FELM	93.15	91.84	90.69

5 Conclusion

A new extension of ELM called FELM is proposed in this paper, which introduces a set of fuzzy memberships and a fuzzy matrix to the traditional ELM method. The proposed method retains the advantages of ELM, such as all the hidden node parameters are randomly generated and the output weights are analytically determined. In FELM, different from ELM, the inputs with different fuzzy memberships can make different contributions to the learning of the output weights. In addition, according to the specific applications, the fuzzy memberships can be conveniently determined based on classes or examples. The experiments results show that FELM can achieve comparable accuracy as SVM and ELM with much faster learning speed than SVM. More importantly, by introducing the appropriate fuzzy memberships, FELM

can solve some complicated classification problems which are beyond the ability of the traditional ELM, such as imbalanced data problems, cost-sensitive learning, and noise signal problems. In future study, we will investigate to find the better determination methods of the fuzzy memberships which are suited to wider scopes of real-world applications.

References

[1] G. B. Huang, Q. Y. Zhu, and S. K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks", *International Joint Conference on Neural Networks* 2004, vol. 2, pp. 985-990.

[2] G. B. Huang, Q. Y. Zhu, and S. K. Siew, "Extreme learning machine: theory and applications", *Neurocomputing*, vol. 70, no. 1 - no. 3, pp. 489-501, 2006.

[3] G. B. Huang and L. Chen, "Convex incremental extreme learning machine", *Neurocomputing*, vol. 70, pp. 3056-3062, 2007.

[4] G. B. Huang, L. Chen, and C. K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes", *IEEE Trans. Neural Networks*, vol. 17, no. 4, pp. 879-892, 2006.

[5] P. L. Bartlett, "The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network", *IEEE Trans. Information Theory*, vol. 44, no. 2, pp. 525-536, S1998.

[6] W. Deng, Q. Zheng, and L. Chen, "Regularized extreme learning machine", *IEEE Symposium on Computational Intelligence and Data Mining 2009*, pp. 389-395.

[7] A. Y. Chen, J. C. Yang, and C. Wang, "Variational Bayesian extreme learning machine". *Neural Computing & Applications*. DOI 10. 1007/s00521-014-1710-1, 2014.

[8] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, and A. Lendasse, "OP-ELM: Optimally pruned extreme learning machine", *IEEE Trans. Neural Networks*, vol. 21, no. 1, pp. 158-162, 2010.

[9] K. Neumann and J. J. Steil, "Optimizing extreme learning machines via ridge regression and batch intrinsic plasticity", *Neurocomputing*, vol. 102, no. 23 – no. 30, 2013.

[10] S. F. Ding, X. Z. Xu, and R. Nie, "Extreme learning machine and its applications", *Neural Computing & Applications*, DOI 10. 1007/s00521-013-1522-8, 2014.

[11] P. Horata, S. Chiewchanwattana, and K. Sunat, "Robust extreme learning machine", *Neurocomputing*, vol. 105, pp. 31-44, 2013.

[12] W. Y. Deng, Q. H. Zheng, S. G. Lian, L. Chen, and X. Wang, "Ordinal extreme learning machine". *Neurocomputing*, vol. 74, no. 1 – no. 3, pp. 447-456, 2010.

[13] Q. Liu, Q. He, and Z. Shi, "Extreme support vector machine classifier", *Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science*, no. 5012, pp. 222-233, 2008.

[14] G. B. Huang, X. J. Ding, and H. M. Zhou, "Optimization method based extreme learning machine for classification", *Neurocomputing*, vol. 74, pp. 155-163, 2010.

[15] G. B. Huang, X. J. Ding, H. M. Zhou and R. Zhang, "Extreme learning machine for regression and multiclass classification", *IEEE Trans. Syst. Man Cybern*, vol. 42, no. 2, pp. 513-529, 2012.

[16] Domingos, "MetaCost: A general method for making classifiers cost-sensitive", *The 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1999*, pp. 155–164.

[17] C. Elkan, "The foundations of cost-sensitive learning", *The 17th International Joint Conference on Artificial Intelligence, 2001*, pp. 973–978.

[18] C. L. Castro and A. P. Braga, "Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data", *IEEE Trans. Neural Networks and Learning Systems*, vol. 24, no. 6, pp. 888–899, 2013.

[19] Q. Yang, C. Ling, X. Y. Chai, and R. Pan, "Test-cost sensitive classification on data with missing value", *IEEE Trans. Knowledge and Data Engineering*, vol. 18, no. 5, pp. 626–638, 2006.

[20] Y. Zhang and Z. H. Zhou, "Cost-sensitive face recognition", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1758–1769, 2010.

[21] J. W. Lu and X. Z. Zhou, "Cost-sensitive semi-supervised discriminant analysis for face recognition", *IEEE Trans. Information Forensics and Security*, vol. 7, no. 3, pp. 944–953, 2012.

[22] J. W. Lu and Y. P. Tan, "Cost-sensitive subspace analysis and extensions for face recognition", *IEEE Trans. Information Forensics and Security*, vol. 8, no. 3, pp. 510–519, 2013.

[23] R. Fletcher, *Practical Methods of Optimization: Constrained Optimization*, vol. 2, Wiley, New York, 1981.

[24] H. He and E. A. Garcia, "Learning from imbalanced data", *IEEE Trans. Knowl. Data Eng*, vol. 21, no. 9, pp. 1263-1284, 2009.

[25] C. F. Lin and S. D. Wang, "Fuzzy support vector machine", *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 464-471, 2002.

[26] C.B.S. Hettich and C. Merz, UCI Repository of Machine Learning Databases, 1998 [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html.

[27] V.C. Chen, Time-frequency / time-scale analysis for radar applications, [Online]. Available: http://airborne.nrl.navy.mil/~vchen/tftsa.html.