Fuzzy extreme learning machine for classification

W.B. Zhang and H.B. Ji

Compared to traditional classifiers, such as SVM, the extreme learning machine (ELM) achieves similar performance for classification and runs at a much faster learning speed. 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. Proposed is a fuzzy ELM, which introduces a fuzzy membership to the traditional ELM method. Then, the inputs with different fuzzy matrix can make different contributions to the learning of the output weights. For the weighted classification problems, FELM can provide a more logical result than that of ELM.

Introduction: The extreme learning machine (ELM) [1] was originally proposed for the single-hidden-layer feedforward neural networks (SLFNs), and then extended to the generalised SLFNs where the hidden layer need not be neuron like. 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 a thousand times faster than that of the traditional iterative implementations of SLFNs. To solve the classification problems using ELM, [2] proposes that ELM can be applied to support vector machines (SVMs) by simply replacing SVM kernels with ELM kernels. The study of ELM for classification with the standard optimisation method in [3] verifies that ELM can solve any multiclass classification problems directly [4]. 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 this kind of ability. This Letter 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. The proposed method is suitable for real classification applications because of its ability to solve the problems with different weights.

Extreme learning machine: ELM [1] studies the generalised SLFNs whose hidden layer need not be tuned and may not be neuron like. In ELM, all the hidden node parameters are randomly generated, and the output weights are analytically determined. Different from traditional learning method ELM not only tends to reach the smallest training error but also the smallest norm of output weights:

$$\text{minimise}: \sum_{i=1}^{N} \| \boldsymbol{\beta}\mathbf{h}(x_i) - t_i\|$$

$$\text{and minimise}: \|\boldsymbol{\beta}\| \tag{1}$$

where β_i is the output weight form the ith hidden node to the output node, $\mathbf{h}(x_i)$ is the output vector of the hidden layer with respect to the input x_i and t_i is the label corresponding to x_i. According to [5], for feedforward neural networks reaching a smaller training error, the smaller the norm of weights is, the better generalisation performance the networks have. Then, the minimal norm least square method is used in the implementation of ELM:

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

where $\mathbf{T} = [t_1, t_2, \ldots, t_N]^T$, and $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalised inverse of matrix $\mathbf{H}$. When $\mathbf{H}^T\mathbf{H}$ is nonsingular, $\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{h}^T$, or when $\mathbf{H}\mathbf{H}^T$ is nonsingular, $\mathbf{H}^{\dagger} = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}$. Compared to SVM, ELM achieves similar performance for classification and runs at a much faster learning speed. However, a majority of real applications are weighted classification problems, whereas the traditional ELM lacks this kind of ability. Therefore, this Letter proposes fuzzy ELM to solve this problem.

Fuzzy ELM: In real world classification problems, according to the different weights, the effects of the training points should be different. A set S of labelled training points with associated fuzzy membership [6] are introduced:

$$(\mathbf{x}_1, t_1, s_1), \ldots, (\mathbf{x}_N, t_N, s_N) \tag{3}$$

Each training point $\mathbf{x}_i$ is given a label t_i and a fuzzy membership $\sigma \le s_i \le 1$ with sufficiently by small $\sigma > 0$. 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, thus $\frac{1}{2}s_i\|\xi_i\|$ is a measure of error with different weighting. The classification problem for the constrained-optimal-based fuzzy ELM can be formulated as

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

$$\text{subject to}: \mathbf{h}(x_i)\beta = \mathbf{t}_i^T - \boldsymbol{\xi}_i^T, \; i = 1, \ldots, N \tag{4}$$

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

$$L_{FELM} = \frac{1}{2}\|\beta\|^2 + C\frac{1}{2}\sum_{i=1}^{N} s_i\|\boldsymbol{\xi}_i\|^2 - \sum_{i=1}^{N}\sum_{j=1}^{m} \alpha_{i,j}(\mathbf{h}(x_i)\beta_j - t_{i,j} + \xi_{i,j}) \tag{5}$$

We can have the KKT corresponding optimality conditions as follows:

$$\frac{\partial L_{FELM}}{\partial \beta_j} = 0 \rightarrow \beta_j = \sum_{i=1}^{N} \alpha_{i,j}\mathbf{h}(x_i)^T \rightarrow \beta = \mathbf{H}^{\mathbf{T}}\boldsymbol{\alpha} \tag{6a}$$

$$\frac{\partial L_{FELM}}{\partial \boldsymbol{\xi}_i} = 0 \rightarrow \boldsymbol{\alpha}_i = Cs_i\boldsymbol{\xi}_i, \; i = 1, \ldots, N \tag{6b}$$

$$\frac{\partial L_{FELM}}{\partial \boldsymbol{\alpha}_i} = 0 \rightarrow \mathbf{h}(x_i)\beta - \mathbf{T}_i^T + \boldsymbol{\xi}_i^T = 0, \; i = 1, \ldots, N \tag{6c}$$

By substituting (6a) and (6b) into (6c), the equations can be equivalently written as

$$\left(\frac{\mathbf{S}}{C} + \mathbf{H}\mathbf{H}^T\right)\boldsymbol{\alpha} = \mathbf{T} \tag{7}$$

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 (6a) and (7),

$$\beta = \mathbf{H}^T\left(\frac{\mathbf{S}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T} \tag{8}$$

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

$$\mathbf{f}(x) = \mathbf{h}(x)\beta = \mathbf{h}(x)\mathbf{H}^T\left(\frac{\mathbf{S}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T} \tag{9}$$

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

$$f(\mathbf{x}) = sign\left(\mathbf{h}(x)\mathbf{H}^T\left(\frac{\mathbf{S}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1}\mathbf{T}\right) \tag{10}$$

For m-class cases, the predicted class label of a testing point is the index number of the output node which has the highest output value.

$$label(\mathbf{x}) = \arg\max_{j\in\{1,\ldots,m\}} f_j(\mathbf{x}) \tag{11}$$

Furthermore, according to different applications, the fuzzy matrix $\mathbf{S}$ can be set flexibly to solve the different problems.

Simulation results: The performance of different algorithms (SVM, ELM and FELM) is compared in real world benchmark multiclass classification data sets which are taken from the UCI Machine Learning Repository [7]. The feature mapping used in ELM and

FELM is the sigmoid function [4] whose parameters are randomly generated. In the simulations, FELM is used to solve the weighted classification problems. The fuzzy memberships s_i are respectively set as: **Glass** (0.3, 0.2, 0.2, 0.1, 0.1, 0.1), **Iris** (0.5, 0.3, 0.2), **Segment** (0.3, 0.2, 0.1, 0.1, 0.1, 0.1, 0.1), **Vehicle** (0.4, 0.4, 0.1, 0.1), **Wine** (0.4, 0.4, 0.2).

Table 1 shows the testing rate and training time comparison of SVM, ELM and FELM. It can be seen that FELM can achieve comparable performance to SVM and ELM. Simultaneously, the learning speed of ELM and FELM are much faster than that of SVM.

Table 1: Performance comparison of SVM, ELM and FELM

Datasets	SVM		ELM		FELM	
	Rate (%)	Time(s)	Rate (%)	Time(s)	Rate (%)	Time(s)
Glass	67.83	0.2791	67.12	0.0255	67.75	0.0271
Iris	95.12	0.0714	97.64	0.0138	96.67	0.0152
Segment	96.53	13.90	96.07	1.7993	96.52	1.8063
Vehicle	84.37	1.4708	83.48	0.2257	83.44	0.2589
Wine	98.37	0.0719	98.47	0.0190	98.39	0.0208

Tables 2 and 3 show that FELM has the ability to solve the weighted classification problems. The detailed results of datasets **Glass** and **Vehicle** are shown, respectively. Compared to ELM, for the weighted problems, the results of FELM are more logical, which means that the bigger corresponding weight is, the more accurately the label can be classified. On the other hand, if all the labels are considered, FELM can achieve the performance close to ELM. Furthermore, the fuzzy membership s_i and the fuzzy matrix **S** can be set flexibly according to the different problems.

Table 2: Accuracy of each label in **Glass** dataset

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

Table 3: Accuracy of each label in **Vehicle** dataset

Label	1	2	3	4
ELM	84.26	83.33	82.56	84.92
FELM	97.31	96.67	70.67	71.30

Conclusions: In this Letter, we have proposed the FELM which introduces a fuzzy membership and a fuzzy matrix to the traditional ELM method. In FELM, all the hidden node parameters are randomly generated, and the output weights are analytically determined. Different from ELM, the inputs with different fuzzy matrix can make different contributions to the learning of the output weights. The simulation results show that FELM can achieve comparable accuracy to SVM and ELM with much faster learning speed than SVM. More importantly, FELM can provide a more logical result than ELM for the weighted classification problems, implying good application prospects for real world applications.

Acknowledgment: This work was supported by the National Natural Science Foundation of China under grant 60871074.

© The Institution of Engineering and Technology 2013
16 October 2012
doi: 10.1049/el.2012.3642

W.B. Zhang and H.B. Ji (*School of Electronic Engineering, Xidian University, PO Box 229, Xi'an, 710071, People's Republic of China*)

E-mail: zwbsoul@163.com

References

1 Huang, G.B., Zhu, Q.Y., and Siew, C.K.: 'Extreme learning machine: Theory and applications'. *Neurocomputing*, 2006, **70**, pp. 489–501
2 Liu, Q., He, Q., and Shi, Z.: 'Extreme support vector machine classifier'. *Lect. Notes Comput. Sci.*, 2008, **5012**, pp. 222–233
3 Huang, G.B., Ding, X.J., and Zhou, H.M.: 'Optimization method based extreme learning machine for classification', *Neurocomputing*, 2010, **74**, pp. 155–163
4 Huang, G.B., Ding, X.J., Zhou, H.M., and Zhang, R.: 'Extreme learning machine for regression and multiclass classification'. *IEEE Trans. Syst. Man Cybern. B, Cybern.*, 2012, **42**, (2), pp. 513–529
5 Bartlett, P.L.: '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. Inf. Theory*, 1998, **44**, (2), pp. 525–536
6 Lin, C.F., and Wang, S.D.: 'Fuzzy support vector machine', *IEEE Trans. Neural Netw.*, 2002, **13**, (2), pp. 464–471
7 UCI Repository of Machine Learning Databases, C. Blake, E.Keogh, and C. Merz, 1998, [Online]. Available: http:http://www.ics.uci.edu/mlearn/MLRepository.html