Neural Comput & Applic
DOI 10.1007/s00521-015-1918-8

ORIGINAL ARTICLE

A wavelet extreme learning machine

Shifei Ding[1,2] · Jian Zhang[1] · Xinzheng Xu[1] · Yanan Zhang[1]

Received: 24 October 2014 / Accepted: 24 April 2015

Abstract Extreme learning machine (ELM) has been widely used in various fields to overcome the problem of low training speed of the conventional neural network. Kernel extreme learning machine (KELM) introduces the kernel method to ELM model, which is applicable in Stat ML. However, if the number of samples in Stat ML is too small, perhaps the unbalanced samples cannot reflect the statistical characteristics of the input data, so that the learning ability of Stat ML will be influenced. At the same time, the mix kernel functions used in KELM are conventional functions. Therefore, the selection of kernel function can still be optimized. Based on the problems above, we introduce the weighted method to KELM to deal with the unbalanced samples. Wavelet kernel functions have been widely used in support vector machine and obtain a good classification performance. Therefore, to realize a combination of wavelet analysis and KELM, we introduce wavelet kernel functions to KELM model, which has a mix kernel function of wavelet kernel and sigmoid kernel, and introduce the weighted method to KELM model to balance the sample distribution, and then we propose the weighted wavelet–mix kernel extreme learning machine. The experimental results show that this method can effectively improve the classification ability with better generalization. At the same time, the wavelet kernel functions perform very well compared with the conventional kernel functions in KELM model.

Keywords Wavelet kernel function · Extreme learning machine · Wavelet–mix kernel function · Weighted method

✉ Shifei Ding
dingsf@cumt.edu.cn

1 School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

2 Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

1 Introduction

ELM algorithm is based on the perceptron model with one hidden layer. Huang et al. summed up the idea of parameter iteration and raised ELM algorithm to train SLFNs [1]. The learning ability of ELM is only related to the number of hidden neurons if the hidden units can be increased gradually [2]. The whole learning process of ELM algorithm needs no iteration, so ELM algorithm saved much time than the conventional neural network algorithms and improved learning efficiency greatly. However, the conventional ELM algorithm is based on the empirical risk minimization, without considering the structural risk, which may cause over-fitting problems. Aiming at the deficiency of ELM algorithm, some ideas of parameter adjusting method emerged as the times require. Ding et al. [3] proposed an algorithm of a novel self-adaptive extreme learning machine, and Zhang et al. [4] proposed an algorithm researching for prediction algorithm of extreme learning machines based on rough sets. Yang et al. proposed hybrid chaos optimization ELM algorithm, which limited the number of useless neurons in incremental extreme learning machine model, and pointed out that the merits of the hidden layer neurons also affect the convergence speed of network greatly; therefore, they proposed a bidirectional extreme learning machine model [5, 6]. All these optimization ideas just changed the way of parameter optimization in conventional extreme learning machine

Published online: 09 May 2015

model. However, these ideas did not fully mine the information of input data, so that there is still much room for improvement. In this case, we decide to optimize the ELM algorithm by using kernel methods.

Kernel methods (KMs) are a pattern recognition algorithm, which aim at identifying and learning the relationship of the input data. Huang et al. proposed the kernel extreme learning machine (KELM). Similar to the conventional kernel methods, the selection of kernel functions is also the core problem of KELM. Kernel functions appropriate or not have great influence on the performance of KELM. The kernel functions commonly used are Gauss function, polynomial function, and so on. Ding et al. [7] tried to integrate different advantages of each function and proposed a KELM model based on the mixed kernel functions. The mix kernel ELM model can make full use of the different kernel functions and performs very well in classification problems.

Wavelet analysis technology has been widely used in the optimization of neural network in recent years. Wavelet analysis has the characteristics of multi-scale interpolation and sparse change and is suitable for local analysis and detection of signals. At present, the application of wavelet kernel in support vector machine (SVM) achieved many achievements. Zhou et al. [8] proposed wavelet kernel support vector machine model, which succeeds in signal detection. The wavelet–mix kernel functions have been used in SVM and obtain a good performance. However, if the number of samples in Stat ML is too small, perhaps the unbalanced samples cannot reflect the statistical characteristics of the input data, so that the learning ability of Stat ML will be influenced. To solve this question, we introduce the weighted method to KELM model.

From the results of the study of Wang et al. [9], wavelet kernel functions can widen the kernel function selection range of KELM and promote its further development. Based on the above points, to combine the advantages of different kernel functions, we propose the weighted WKELM algorithm, which combines the wavelet analysis and kernel extreme learning machine. Then we introduce the wavelet–mix kernel functions to weighted KELM model. According to the experiments, the wavelet KELM gets a better performance than conventional ELM in classification problems. At the same time, the wavelet kernel functions perform very well when the algorithm is compared with the conventional kernel functions in KELM model.

This paper is organized as follows: Sect. 2 is ELM model and introduces the principle and application of ELM and KELM. Section 3 introduces the wavelet analysis theory and the wavelet kernel functions. In Sect. 4, we propose the weighted wavelet–mix kernel extreme learning machine model. Section 5 is the analysis of the experiments. The conclusion is given in Sect. 6.

2 ELM and KELM model

2.1 The ELM model

ELM algorithm is based on the perceptron model with one hidden layer. By increasing the number of hidden layer nodes, we need not to adjust the input weights or hidden layer bias if we randomly assign the input weights and bias values. So the algorithm runs really fast. The network structure is shown in Fig. 1:

For N different training samples $(x_i, t_i) \in R^n \times R^m$ $(i = 1, 2, 3, \ldots, n)$, the number of hidden units is $\tilde{N}$. The SLFN model, which has activation function $f(x)$, can be expressed as:

$$\sum_{i=1}^{\tilde{N}} V_i f_i(x_j) = \sum_{i=1}^{\tilde{N}} V_i f(a_i \cdot x_j + b_i), \quad j = 1, \ldots, N \tag{1}$$

where $a_i = [a_{i1}, a_{i2}, \ldots, a_{in}]^T$ is the input weight vector connected to the hidden layer node i, b_i is the bias value of hidden layer nodes, $V_i = [V_{i1}, V_{i2}, \ldots, V_{im}]^T$ are the output weight vector connected to the hidden layer node i, $a_i \cdot x_j$ means inner product of a_i, and x_j, $f(x)$ can be “sigmoid,” “RBF,” “sine,” and so on.

Equation (1) can be rewritten as follows:

$$HV \tag{2}$$

where H is the output matrix of hidden layer and V is the output weight matrix. T is the label matrix.

$$H = \begin{bmatrix} f(a_1 \cdot x_1 + b_1) & \cdots & f(a_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & \cdots & \vdots \\ f(a_1 \cdot x_N + b_1) & \cdots & f(a_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}}$$

$$V = \begin{bmatrix} V_1^T \\ \vdots \\ V_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \qquad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m}$$

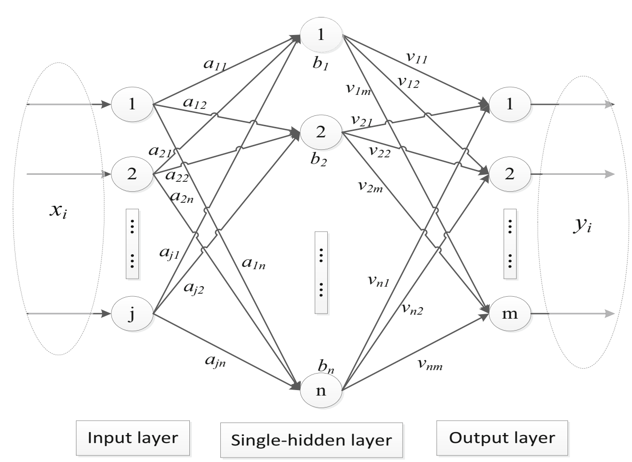

Fig. 1 ELM model

Not all parameters need to be adjusted, when the excitation function $f(x)$ is infinitely differentiable at any interval. At the start of training process, SLFNs assigned random values to the input weights a and hidden layer node bias b. When the input weights and hidden layer node biases are determined by random assignment methods, we can get the hidden layer output matrix H from the input samples. Therefore, training SFLNs are transformed into solving the least square solutions.

Introducing regularization theory to ELM model, the cost function can be expressed as:

$$\min L_{\mathrm{ELM}} = \frac{1}{2}\|V\|^2 + \frac{C}{2}\|T - HV\|^2 \tag{3}$$

The least squares solution of Eq. (3) is

$$V - CH^T(T - HV) = 0 \tag{4}$$

T is the label matrix, and V is the output weights matrix.

When the number of training samples is more than the number of hidden layer nodes,

$$V = \left(\frac{I}{C} + H^T H\right)^{-1} H^T T \tag{5}$$

When the number of training samples is less than the number of hidden layer nodes,

$$V = H^T \left(\frac{I}{C} + HH^T\right)^{-1} T \tag{6}$$

In ELM algorithm, when the number of hidden layer units is large enough, the classification accuracy of ELM algorithm is always stable.

2.2 The kernel function theory

The basic thinking of kernel function theory is as follows: Use a nonlinear map to map the original data to a high-dimensional feature space, and then make classification in this new space by a linear classifier. For $\overrightarrow{x}, \overrightarrow{z}$ in X, if a mapping of $\varphi()$ makes: $k(\overrightarrow{x}, \overrightarrow{z}) = (\varphi(\overrightarrow{x}), \varphi(\overrightarrow{z}))$, then $k(\cdot,\cdot)$ is called a kernel function. When introducing the kernel method to machine learning models, the nonlinear processing ability can be improved.

Theorem 1 *If function $k(\cdot,\cdot)$ is a mapping on $R^n \times R^n \to R$, $k(\cdot,\cdot)$ is an effective kernel function (also called Mercer Kernel Function), if and only if the kernel function matrix is semi positive definite and symmetric.*

The kernel thought has been developed into the kernel method. A proper kernel function must be based on the characteristics of the problem. In SVM, using different kernel functions can obtain different learning effects.

The kernel functions can be divided into global kernels such as sigmoid kernel function and local kernel functions such as Gaussian kernel function. The learning ability of local kernel functions is stronger; meanwhile, the generalization ability of global kernel functions is stronger. The local kernel functions and the global kernel functions can be combined to obtain a better learning ability.

The kernel functions also can be divided into the following two types:

1. Translation-invariant kernel

A translation-invariant kernel function is a function with the form $k(\overrightarrow{x}, \overrightarrow{z}) = f(\overrightarrow{x} - \overrightarrow{z})$, among which $f : X \to R$ is a function like: Gauss function $k(\overrightarrow{x}, \overrightarrow{z}) = \exp(-a||\overrightarrow{x} - \overrightarrow{z}||^2)$. To be a kernel function, $f(\overrightarrow{x} - \overrightarrow{z})$ should satisfy Theorem 2.

Theorem 2 *Let $f : X \to R$ is a function which is continuous and integrable, then $f : X \to R$ is a kernel function iff satisfies the necessary and sufficient condition: $f(0) > 0$ and its Fourier transform: $\tilde{f}(\omega) = \int_X f(\overrightarrow{x})e^{-j\omega x}dx \geq 0$*

2. Rotation-invariant kernels

A rotation-invariant kernel function is a function with the form: $k(\overrightarrow{x}, \overrightarrow{z}) = f((\overrightarrow{x}, \overrightarrow{z}))$, among which $f : D \to R \quad (D \subset R)$, such as the homogeneous polynomial kernel function: $k(\overrightarrow{x}, \overrightarrow{z}) = \left(\overrightarrow{x}, \overrightarrow{z}\right)^p$. To be a kernel function, $f()$ should satisfy Theorem 3.

Theorem 3 *If $f(t)$ is defined on $-r < t < r, \quad (0 < r < \infty)$, and $f^{(n)}(t) \geq 0 \quad 0 < t < r$, then $k(\overrightarrow{x}, \overrightarrow{z}) = f((\overrightarrow{x}, \overrightarrow{z}))$ is a kernel function.*

According to the research [10], the sum function of two kernel functions is still a kernel function. The combination of two different kernel functions has been widely used in KELM and SVM.

2.3 The KELM model

Section 2.1 part introduces the basic model of ELM. Now we apply the kernel functions to ELM model. The inner product of two mappings can be written as kernel functions.

$$HH^T = \begin{bmatrix} (h(\vec{x_1}), h(\vec{x_1})) & (h(\vec{x_1}), h(\vec{x_2})) & \dots & (h(\vec{x_1}), h(\vec{x_N})) \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ (h(\vec{x_N}), h(\vec{x_1})) & (h(\vec{x_4}), h(\vec{x_2})) & \dots & (h(\vec{x_N}), h(\vec{x_N})) \end{bmatrix} = \begin{bmatrix} k(\vec{x_1}, \vec{x_1}) & k(\vec{x_1}, \vec{x_2}) & \dots & k(\vec{x_1}, \vec{x_N}) \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ k(\vec{x_N}, \vec{x_1}) & k(\vec{x_N}, \vec{x_2}) & \dots & k(\vec{x_N}, \vec{x_N}) \end{bmatrix} \tag{7}$$

As we know:

So the output of ELM is:

$$y = \overrightarrow{h(\vec{x})}^T V = \overrightarrow{h(\vec{x})}^T H^T (HH^T)^{-1} T \tag{8}$$

Then introduce the regularization to balance the empirical risk and the structural risk minimization,

$$y = \overrightarrow{h(\vec{x})}^T V = \overrightarrow{h(\vec{x})}^T H^T \left(\frac{1}{C} I + HH\right)^{T-1} T \tag{9}$$

where C is the regularization parameter

So:

$$y = \overrightarrow{h(\vec{x})}^T H^T (HH^T)^{-1} T = \begin{bmatrix} k(\vec{x}, \vec{x_1}) \\ k(\vec{x}, \vec{x_2}) \\ \dots \\ k(\vec{x}, \vec{x_N}) \end{bmatrix} \cdot \left(\frac{1}{C} I + HH^T\right)^{-1} T \tag{10}$$

where I is unit matrix and T is the label matrix.Let $K = \begin{bmatrix} k(\vec{x_1}, \vec{x_1}) & k(\vec{x_1}, \vec{x_2}) & \dots & k(\vec{x_1}, \vec{x_N}) \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ k(\vec{x_N}, \vec{x_1}) & k(\vec{x_N}, \vec{x_2}) & \dots & k(\vec{x_N}, \vec{x_N}) \end{bmatrix}$, so:

$$y = \begin{bmatrix} k(\vec{x}, \vec{x_1}) \\ k(\vec{x}, \vec{x_2}) \\ \dots \\ k(\vec{x}, \vec{x_N}) \end{bmatrix} \cdot \left(\frac{1}{C} I + K^T\right)^{-1} T \tag{11}$$

In this way, the whole calculation process completely has been transformed into the kernel function form.

Kinds of kernel functions have been introduced to KELM algorithm, such as Gaussian kernel function, sigmoid kernel function, mix kernel function, and wavelet kernel function.

3 Wavelet kernel function theory

3.1 Wavelet theory

Wavelet analysis is a development of Fourier transform: Fourier transform is one of the most widely used methods in signal processing and abandons time information, but it also has a serious deficiency. Since the functions which used the Fourier transformation are periodic on infinite interval, so the Fourier transformation ignores the temporal characteristics of original functions and cannot capture the temporal information of signals [11–13]. At the same time, the sine and cosine waves are the basic functions of the original functions in Fourier transformation. In summary, the drastic changes of the original signal in a location cannot be well represented.

The wavelet is a finite-length waveform, which has the following characteristics: (1) Time domain is compactly supported or approximation compactly supported. (2) The DC component is 0. The advantages of wavelet analysis are integrated in time domain and frequency domain. The time domain characteristics of wavelet transform can be expressed by the wavelet functions which are translated from a wavelet basis function. At the same time, we use different wavelet functions after translation transformation to fit original signals. So, to some extent, the wavelet transformation is better than Fourier transformation.

The wavelet functions are obtained from a mother wavelet function by translation and size expansion. Wavelet analysis is to decompose the signal into the superposition of a series of wavelet functions. The wavelet functions are obtained by $\psi(t)$ through translation τ

$$f_x(a, \tau) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi\left(\frac{t - \tau}{a}\right) dt \quad a > 0 \tag{12}$$

The time domain expression is:

$$f_x(a, \tau) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(\omega) \psi(a\omega) e^{j\omega} dt \quad a > 0 \tag{13}$$

From the Formula (12) and Formula (13), we can see: Wavelet analysis can reflect the local characteristics of signals by wavelet transform. Therefore, wavelet analysis as a kind of analytical method caused wide attention [14–17].

3.2 Wavelet kernel function

If a function satisfies the Mercer theorem, then it is a kernel function [18], and it can be used as a kernel function of the KELM. Next, we will prove that wavelet kernel functions can be the kernel functions of KELM.

If $\psi(x)$ is a mother wavelet and $\vec{x}, \vec{z} \in R^d$, then the translation-invariant wavelet kernel is

$$k(\vec{x}, \vec{z}) = \prod_{i=1}^{d} \left(\psi\left(\frac{\vec{x_i} - \vec{z_i}}{a}\right)\right) \tag{14}$$

The rotation-invariant wavelet kernel function is

$$k(\vec{x}, \vec{z}) = \prod_{i=1}^{d} \left(\psi\left(\frac{\vec{x}_i - b}{a}\right) \cdot \psi\left(\frac{\vec{z_i} - b}{a}\right)\right) \tag{15}$$

Morlet wavelet kernel function has been used in SVM, and Mexican hat wavelet kernel function has been used in twin support vector machine (TWSVM). The mother wavelet of Morlet wavelet function is

$$\psi(x) = \cos(1.75 * x)\exp(-x^2/2) \tag{16}$$

The mother wavelet of Mexican hat wavelet function is

$$\psi(x) = (1 - x^2)\exp(-x^2/2)) \tag{17}$$

Therefore, Morlet wavelet kernel function is

$$\begin{aligned} k(\vec{x}, \vec{z}) &= \prod_{i=1}^{d} \cos(1.75 * (\vec{x_i} - \vec{z_i})/a) \\ &\quad \exp\left(-(\vec{x_i} - \vec{z_i})^2 \Big/ (2a^2)\right) \\ &= \prod_{i=1}^{d} \cos(1.75 * (\vec{x_i} - \vec{z_i}) \Big/ \gamma^{1/2}) \\ &\quad \exp\left(-(\vec{x_i} - \vec{z_i})^2 \Big/ (2\gamma)\right), \gamma > 0 \end{aligned} \tag{18}$$

Mexican hat wavelet kernel function is

$$\begin{aligned} k(\vec{x}, \vec{z}) &= \prod_{i=1}^{d} (1 - (\vec{x_i} - \vec{z_i})^2 \Big/ a^2) \\ &\quad \exp\left(-(\vec{x_i} - \vec{z_i})^2 \Big/ 2a^2)\right) \\ &= \prod_{i=1}^{d} (1 - (\vec{x_i} - \vec{z_i})^2 \Big/ \gamma) \\ &\quad \exp\left(-(\vec{x_i} - \vec{z_i})^2 \Big/ 2\gamma\right), \quad \gamma > 0 \end{aligned} \tag{19}$$

It has been proved in [19, 20] that Morlet wavelet kernel function and Mexican hat wavelet kernel function are Mercer kernel functions.

The wavelet–mix kernel functions have been introduced to SVM model. However, there is not much research about wavelet kernel functions in KELM model.

4 Wavelet kernel extreme learning machine model

In this part, we introduce the wavelet–mix kernel function to KELM model and introduce the thought of weighted method to KELM and then propose the weighted wavelet–mix kernel extreme learning machine (weighted WKELM) algorithm.

The sigmoid kernel function is a global kernel function and a translation-invariant kernel function as well. Wavelet kernel functions are local kernel functions. So, the combination of these two kernel functions can balance the learning ability and the generalization ability in theory and make full use of the characteristics of rotation invariant and translation invariant.

The weighted ELM model is proposed to deal with the samples which are unbalanced in probability distribution, and this method performs very well. Our weighted WKELM algorithm introduces the weighted method to cost function to obtain the similar effect of weighted ELM.

Case the KELM model comes from ELM model, and the weighted cost function can be expressed as follows:

$$\min L_{\mathrm{ELM}} = \frac{C}{2}\|\beta\|^2 + \frac{W}{2}\|T - HV\|^2 \tag{20}$$

$$\beta = \begin{cases} H^T(CI + WHH^T)^{-1}WT, & N < L \\ (CI + H^T WH)^{-1}H^T WT, & N \geq L \end{cases} \tag{21}$$

In KELM model, the output can be expressed as follows:

$$y = \begin{bmatrix} k(\vec{x}, \vec{x_1}) \\ k(\vec{x}, \vec{x_2}) \\ \cdots \\ k(\vec{x}, \vec{x_N}) \end{bmatrix} (CI + WK^T)^{-1}WT \tag{22}$$

where K is the kernel matrix, W is the weighted matrix, and C is the regularization parameter.

Then we introduce the wavelet–mix kernel functions to our KELM model to realize the weighted WKELM algorithm.

5 The analysis of WKELM algorithm

We use several commonly used data in UCI machine learning database to verify the weighted WKELM algorithm. Six datasets were used in the experiments: Ionosphere dataset, Glass dataset, Spect dataset, Diabetes dataset, Vowel-context dataset, and Breast cancer dataset. We use MATLAB 2012b to finish the experiments in the Windows 8.1 system (16 G memory, CPU core i7-4710hq).

The data characteristics of the six datasets are shown in Table 1.

The datasets we selected come from UCI, some of them are balanced, and others are unbalanced. We made WKELM and ELM experiments with six kinds of samples and then compared the experimental results.

In the experiments, we compared the wavelet KELM model with ELM and other commonly used KELM

Table 1 Basic features of the four datasets

Datasets	Number of training	Number of testing	Attributes	Categories
Ionosphere	200	151	34	2
Glass	163	51	10	7
Spect	80	187	23	2
Diabetes	576	192	8	2
Vowel context	264	264	10	10
Breast cancer	490	209	10	2

models. The experimental precisions of each dataset are presented in Table 2.

In weighted KELM, we have two parameters which are selected from [2^5, 2^4,…, 2^{-5}]. The mixed factor of the two kernel functions is selected from [0, 0.1, 0.2,…, 1]. The parameters in other KELM models are selected in the same way.

To get the highest classification accuracy, the classification model should make full use of the input information and perform some invariance of the input data. As we can see in Table 2, the wavelet KELM models perform very well when they are compared with the ELM and conventional kernel functions in KELM models. However, Gaussian KELM almost gets the best experimental results compared with the other KELM models which have single kernel function. The wavelet kernel functions match the non-smooth functions very well; however, for the smooth functions and the global characteristics of input data, wavelet kernel functions have no advantages. At the same time, the weighted WKELM performs very well. The combination of sigmoid kernel function and the wavelet kernel function exceeds the classification effects of any other models in classification problems. Therefore, we can make an assumption: The weighted WKELM combines the advantages of weighted method and wavelet analysis technology. In this way, weighted WKELM algorithm solves the problem of unbalanced input data and makes full use of the characteristics of wavelet kernel functions and conventional global kernel functions which are translation-invariant kernel functions as well.

However, we cost much time to select our parameters. The weighted method we used in this paper is a simple method, and also other weighted methods can be used and investigated.

Table 2 Testing accuracy of each algorithm

Testing accuracy	Weighted WKELM (%)	ELM (%)	Lin-KELM (%)	Gauss -KELM (%)	Morlet-KELM (%)	Mexico hat-KELM (%)	Sigmoid-KELM (%)
Ionosphere	100	94.44	95.36	98.01	96.69	97.35	92.05
Spect	94.12	69.63	90.91	79.18	79.68	79.14	93.58
Vowel context	28.20	21.97	24.90	27.10	26.55	26.41	24.90
Diabetes	80.73	77.60	68.23	80.73	80.21	79.69	78.65
Breast cancer	98.57	97.24	94.76	98.10	97.62	97.62	97.62
Glass	68.97	57.13	49.43	63.22	67.87	66.67	39.08

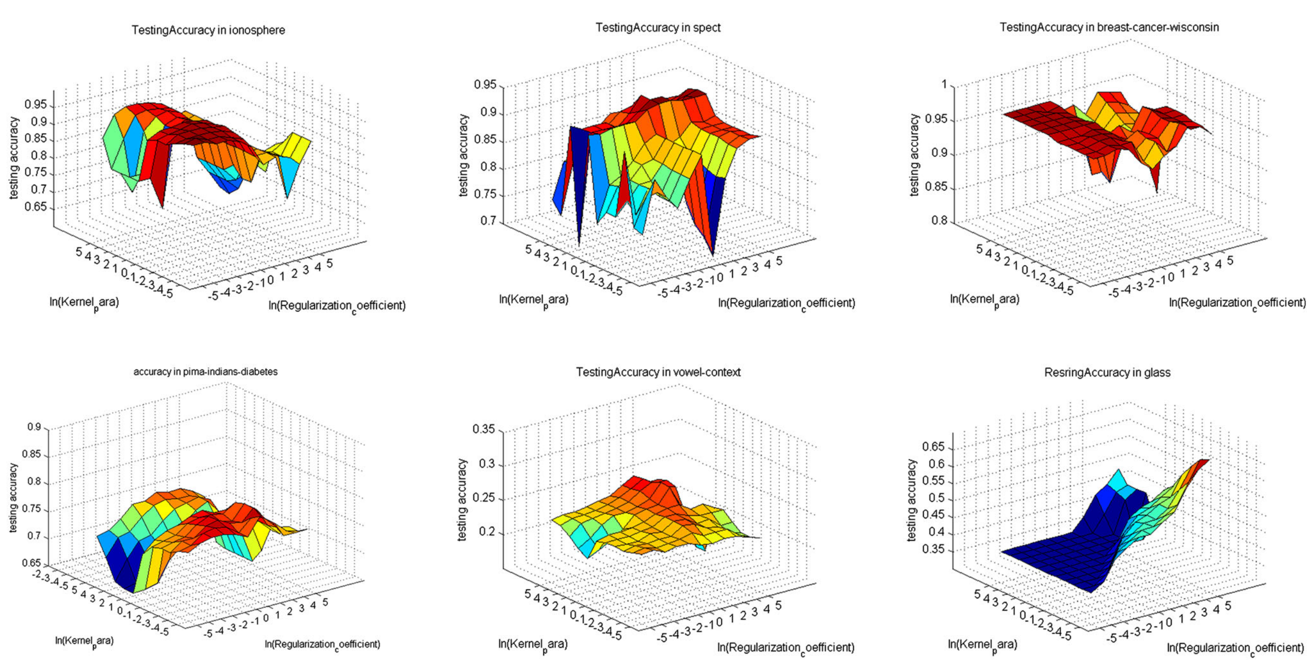

Fig.2 Wavelet–mix kernel ELM accuracy

Table 3 Training time of weighted WKELM and other algorithms

Training time (s)	Weighted WKELM	ELM	Lin-KELM	Gauss -KELM	Morlet-KELM	Mexico hat-KELM
Ionosphere	142.6904	7.7081	2.2031	3.4531	24.7188	13.3594
Spect	56.2188	3.2344	1.4531	1.7344	7.5621	4.8594
Vowel context	343.375	11.875	7.7188	13.8906	38.5	22.5
Diabetes	264.5781	18.3906	7.9063	10.8906	39.5855	20.2188
Breast cancer	176.399	14.2122	7.5753	10.1601	27.4307	19.9134
Glass	73.9445	5.5692	1.8978	2.1584	9.0325	5.8947

The following figures show the accuracy of weighted WKELM. The meaning of the *x*-axis and *y*-axis is shown in figures: they mean the regularization parameter and the kernel parameter; the *z*-axis means the testing accuracy. By the way, we use one parameter to simplify the wavelet kernel parameters. The testing accuracies are shown in Fig. 2.

As we can see in Fig. 2, our model is effective in classification problem. When we introduce the weighted method and the wavelet–mix kernel functions to KELM model, the testing accuracy is relatively dependent on the kernel parameter and the regularization parameter. If we select proper parameters, the testing accuracy is high and the algorithm is stable.

The training time is shown in Table 3:

As we can see, because the wavelet–mix KELM needs much more loops in training process, it takes more time than other models. Mexico hat-KELM performs better than Morlet-KELM in training time. The conventional kernel functions get the best performance in our experiments.

In theory, the training time depends on the complexity of the kernel functions. In our experiments, the training time includes parameter adjustment process. The complexity of wavelet kernel functions is higher than that of the conventional kernel functions. Because the size of kernel matrix is dependent on the number of samples and the calculation complexity of kernel methods is high, our algorithm is not a proper algorithm to deal with big data.

6 Conclusion

The weighted WKELM model based on wavelet–mix kernel combines wavelet–mix kernel functions and ELM algorithm. We select the appropriate kernel function to optimize KELM. The algorithm does wavelet transform of input signals according to wavelet kernel functions. Then we propose the weighted wavelet–mix KELM model, which gets the best experimental results in aspect of testing accuracy. However, the time complexity of weighted KELM is higher than the conventional KELM algorithms. The weighted method we used in this paper can be improved as well. So, there is still much work to do in our weighted WKELM model.

Acknowledgments This work is supported by the National Natural Science Foundation of China (No. 61379101), the National Key Basic Research Program of China (No. 2013CB329502), and the Basic Research Program (Natural Science Foundation) of Jiangsu Province of China (No. BK20130209).

References

1. Liang NY, Huang GB (2006) A fast and accurate online sequential learning algorithm for feed forward networks. IEEE Trans Neural Netw 17(6):1411–1423
2. 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
3. Ding SF, Ma G, Shi ZZ (2014) A novel self-adaptive extreme learning machine based on affinity propagation for radial basis function neural network. Neural Comput Appl 24(7–8):1487–1495
4. Zhang YN, Ding SF, Xu XZ et al (2013) An algorithm research for prediction algorithm of extreme learning machines based on rough sets. J Comput 8(5):1335–1342
5. Yang YM, Wang YN, Yuan XF (2012) Bidirectional extreme learning machine for regression problem and its learning effectiveness. IEEE Trans Neural Netw Learn Syst 23(9):1498–1505
6. Yang YM, Wang YN, Yuan XF (2013) Parallel chaos search based incremental extreme learning machine. Neural Process Lett 37(3):277–301
7. Ding SF, Zhang YN, Xu XZ (2013) A novel extreme learning machine based on hybrid kernel function. J Comput 8(8):2110–2117
8. Zhou Y, Hou TS (2013) Detection of line spectrum signal detection based on harmonic wavelet kernel support vector regression algorithm. Comput Simul 30(1):263–267
9. Wang J, Guo CL (2013) Wavelet kernel extreme learning classifier. Microelectron Comput 10:73–76+80
10. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge
11. Holland JH (1975) Adaptation in neural and artificial systems. MIT Press, Cambridge
12. Zhong JC (2004) Fault diagnosis study based on wavelet neural network. Ocean University of China, QingDao
13. Zhang ZG (2005) Research on fault diagnosis based on wavelet neural network. Daqing Petroleum Institute, DaQing
14. Ding SF, Qi BJ, Tan HY (2011) An overview on theory and algorithm of support vector machines. J Univ Electron Sci Technol China 40(1):2–10
15. Zhou XY, Ye YZ (2006) Application of wavelet analysis to fault diagnosis. Control Eng China 13(1):70–73

16. Chen Z (2001) A wavelet neural network based on BP algorithm learning. J Ocean Univ Qingdao 31(1):122–128
17. Zhu XL, Long YP, Jia L (2007) Traffic volume forecasting algorithm based on RBF neural network. J Shandong Univ 37(4):24–27
18. Zhang GB (2005) The improvement and application in wavelet neural network. Fuzhou University, Fuzhou
19. Zhang L, Zhou WD, Jiao LC (2004) Wavelet support vector machine. IEEE Trans Syst Man Cybern B Cybern 34(1):34–39
20. Ding SF, Wu FL, Shi ZZ (2014) Wavelet twin support vector machine. Neural Comput Appl 25(6):1241–1247