doi:10.21311/001.39.7.13

Parameters Selection of Kernel Based Extreme Learning Machine Using Particle Swarm Optimization

Liwei Zhang, Hongtao Wang, Cuiran Zhao
Northeast Dianli University, School of Electrical Engineering, Changchun 132012, Jilin, China

Yinghui Zhang
Jibei Electric Power Maintenance Company,102400Beijing, China
**Corresponding author(E-mail: zlwbd@126.com)*

Abstract
The generalization performance of kernel based extreme learning machine (KELM) with Gaussian kernel are sensitive to the parameters combination (C, γ). The best generalization performance of KELM with Gaussian kernel is usually achieved in a very narrow range of such combinations. In order to achieve optimal generalization performance, the parameters of KELM with Gaussian kernel were optimized by using particle swarm optimization (PSO) in this paper. To verify its effectiveness, the proposed method was tested on nine benchmark classification data sets compared with KELM optimized by Grid algorithm.

Key words: Extreme Learning Machine, Parameters Optimization, Particle Swarm Optimization.

1. INTRODUCTION

Conventional learningmethods on neural networks such as back-propagation (BP) and SVMmethods apparently face some drawbacks: (1) slow learning speed,(2) trivial human tuned parameters, and (3) trivial learning variantsfor different applications(Huang, Zhu, and Siew, 2006). Extreme learning machine (ELM) is an emerging learningtechnique proposed for generalized single-hidden layer feed forward networks (SLFNs) (Huang, Zhu, and Siew, 2004). ELM overcomes some major constraints faced byconventional learning methods and computational intelligence techniques.

Similar to SVM,kernels can be applied in ELM as well(Huang and Siew, 2005; Huang, Zhou, Ding, et al., 2012).Kernel based ELM (KELM) can beimplemented in a single learning step, so it runs fast.Similar to other kernel based methods, the parameters of KELM are usually assigned empirically or obtained by trials (Huang, Zhou, Ding, et al., 2012). Obviously, it is very time-consuming and the performance achieved with the chosen parameters is suboptimal.

Therefore, parameters optimization based on particle swarm optimization (PSO) isproposed for KELMin this paper. This paper is organized as follows.Section 2 reviews original ELM, equality constrained-optimization-based ELM and KELM. Section 3 introduces the parameters selection for KELM. In Section 4, the proposed KELM with PSO are described in detail. Section 5 discusses the comparison results of the proposed method with Grid algorithm.Conclusions are finally drawn in Section 6.

2. KERNEL BASED ELM

2.1. Original ELM

Extreme Learning Machine (ELM) was originally developedfor the single-hidden layer feedforward networks (SLFNs)and then extended to the "generalized" SLFNs. The hidden layer in ELM neednot be tuned. ELM randomly chooses theinput weights and the hidden neurons' biases and analytically determines the output weights of SLFNs.Input weights are the weights of the connectionsbetween input neurons and hidden neurons andoutput weights are the weights of the connectionsbetween hidden neurons and output neurons.

The output function of ELM for generalizedSLFNs (take one output node case as an example) is

$$f_L(\boldsymbol{x}) = \sum_{i=1}^{L} \beta_i h_i(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})\boldsymbol{\beta} \quad (1)$$

where $\boldsymbol{\beta} = [\beta_1, \cdots, \beta_L]^{\mathrm{T}}$ is the vector of the output weightsbetween the hidden layer of$L$nodes and the output node, and $\boldsymbol{h}(\boldsymbol{x}) = [h_1(\boldsymbol{x}), \cdots, h_L(\boldsymbol{x})]$ is the output (row) vector of thehidden layer with respect to the input$\boldsymbol{x}$.$\boldsymbol{h}(\boldsymbol{x})$ actually maps thedata from the$d$-dimensional input space to the$L$-dimensionalhidden-layer feature space (ELM feature space) **H**, and thus,$\boldsymbol{h}(\boldsymbol{x})$ is indeed a feature mapping.

Given a set of training data $\{(\boldsymbol{x}_i,\boldsymbol{t}_i)|\ \boldsymbol{x}_i\in\mathbf{R}^d,\boldsymbol{t}_i\in\mathbf{R}^m,i=1,\ldots,N\}$. Different from traditional learning algorithms, ELMtends to reach not only the smallest training error but alsothe smallest norm of output weights

$$\text{Minimize: } \|\boldsymbol{H}\boldsymbol{\beta}-\boldsymbol{T}\|^2 \text{ and } \|\boldsymbol{\beta}\| \tag{2}$$

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

$$\boldsymbol{H}=\begin{bmatrix}\boldsymbol{h}(\boldsymbol{x}_1)\\ \vdots\\ \boldsymbol{h}(\boldsymbol{x}_N)\end{bmatrix}=\begin{bmatrix}h_1(\boldsymbol{x}_1) & \cdots & h_L(\boldsymbol{x}_1)\\ \vdots & \ddots & \vdots\\ h_1(\boldsymbol{x}_N) & \cdots & h_L(\boldsymbol{x}_N)\end{bmatrix} \tag{3}$$

and $\boldsymbol{T}$is the expected output matrix

$$\boldsymbol{T}=\begin{bmatrix}\boldsymbol{t}_1^{\mathrm{T}}\\ \vdots\\ \boldsymbol{t}_N^{\mathrm{T}}\end{bmatrix}=\begin{bmatrix}t_{11} & \cdots & t_{1m}\\ \vdots & \ddots & \vdots\\ t_{N1} & \cdots & t_{Nm}\end{bmatrix} \tag{4}$$

The minimal norm least square method was used in the original implementation of ELM

$$\boldsymbol{\beta}=\boldsymbol{H}^{\dagger}\boldsymbol{T} \tag{5}$$

where $\boldsymbol{H}^{\dagger}$ is the*Moore–Penrose generalized inverse*of matrix$\boldsymbol{H}$.

The orthogonalprojection method can be used to calculate theMoore–Penrose generalized inverse of $\boldsymbol{H}$ in two cases: when$\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}$is nonsingular and $\boldsymbol{H}^{\dagger}=(\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})^{-1}\boldsymbol{H}^{\mathrm{T}}$, or when $\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$isnonsingular and$\boldsymbol{H}^{\dagger}=\boldsymbol{H}^{\mathrm{T}}(\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}})^{-1}$.

2.2. Equality Constrained-Optimization-Based ELM

According to the ridge regression theory, one can adda positive value to the diagonal of$\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}or\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$;theresultant solution is more stable and tends to have better generalizationperformance.

For multiclass classifier with multi-outputs, classifiers with *m*-class have*m*output nodes.If the original class label is*p*, the expected output vector of the *m*output nodes is $\boldsymbol{t}_i=[0,\cdots,0,\overset{p}{1},0,\cdots 0]^{\mathrm{T}}$.In this case, only the *p*th element of $\boldsymbol{t}_i=[t_{i1},\cdots,t_{im}]^{\mathrm{T}}$ is one, while the rest of the elements areset to zero.For the constrained-optimization-based ELMwith multi-output node, the classificationproblem can be formulated as

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

$$\text{Subject to: } \boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{\beta}=\boldsymbol{t}_i^{\mathrm{T}}-\boldsymbol{\xi}_i^{\mathrm{T}},\ \ i=1,\cdots,N \tag{6}$$

where $\boldsymbol{\xi}_i=[\xi_{i1},\cdots,\xi_{im}]^{\mathrm{T}}$ is the training error vector of the*m*output nodes with respect to the training sample$\boldsymbol{x}_i$,C is the costparameter.

Based on the Karush–Kuhn–Tucker (KKT) theorem, to train ELM is equivalent to solving thefollowing dual optimization problem:

$$L_{D_{ELM}}=\frac{1}{2}\|\boldsymbol{\beta}\|^2+\frac{C}{2}\sum_{i=1}^{N}\|\boldsymbol{\xi}_i\|^2-\sum_{i=1}^{N}\sum_{j=1}^{m}\alpha_{i,j}(\boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{\beta}_j-t_{i,j}+\xi_{i,j}) \tag{7}$$

We can have the KKTcorresponding optimality conditions as follows:

$$\frac{\partial L_{D_{\mathrm{ELM}}}}{\partial\boldsymbol{\beta}_j}=0\rightarrow\boldsymbol{\beta}_j=\sum_{i=1}^{N}\alpha_{i,j}\boldsymbol{h}(\boldsymbol{x}_i)^T\rightarrow\boldsymbol{\beta}=\boldsymbol{H}^{\mathrm{T}}\boldsymbol{\alpha} \tag{8}$$

$$\frac{\partial L_{D_{\mathrm{ELM}}}}{\partial\boldsymbol{\xi}_i}=0\rightarrow\boldsymbol{\alpha}_i=C\boldsymbol{\xi}_i,\qquad i=1,\cdots,N \tag{9}$$

$$\frac{\partial L_{D_{\mathrm{ELM}}}}{\partial\boldsymbol{\alpha}_i}=0\rightarrow\boldsymbol{h}(\boldsymbol{x}_i)\boldsymbol{\beta}-\boldsymbol{t}_i^{T}+\boldsymbol{\xi}_i^{T}=0,\qquad i=1,\cdots,N \tag{10}$$

By substituting (8) and (9) into(10), the aforementioned equations can be equivalently written as

$$\left(\frac{\boldsymbol{I}}{\boldsymbol{C}}+\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}\right)\boldsymbol{\alpha}=\boldsymbol{T} \tag{11}$$

From (8) and (11), we have

$$\boldsymbol{\beta}=\boldsymbol{H}^{T}\left(\frac{\boldsymbol{I}}{C}+\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}\right)^{-1}\boldsymbol{T} \tag{12}$$

The output function of ELM classifier is

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})\boldsymbol{\beta} = \boldsymbol{h}(\boldsymbol{x})\boldsymbol{H}^{\mathrm{T}}\left(\frac{\boldsymbol{I}}{C} + \boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}\right)^{-1}\boldsymbol{T} \tag{13}$$

2.3. Kernel Based Extreme Learning Machine

If a feature mapping$h(\boldsymbol{x})$is unknown to users,one can apply Mercer's conditions on ELM. We can define akernel matrix for ELM as follows:

$$\boldsymbol{\Omega}_{\mathrm{ELM}} = \boldsymbol{H}\boldsymbol{H}^{\mathrm{T}} : \Omega_{\mathrm{ELM}i,j} = h(\boldsymbol{x}_i)h(\boldsymbol{x}_j) = K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \tag{14}$$

Then, the output function of ELM classifier (13) can bewritten compactly as

$$\begin{aligned}\boldsymbol{f}(\boldsymbol{x}) &= \boldsymbol{h}(\boldsymbol{x})\boldsymbol{H}^{\mathrm{T}}\left(\frac{\boldsymbol{I}}{C} + \boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}\right)\boldsymbol{T} \\ &= \begin{bmatrix} K(\boldsymbol{x}, \boldsymbol{x}_1) \\ \vdots \\ K\left(\boldsymbol{x}, \boldsymbol{x}_N\right) \end{bmatrix}\left(\frac{\boldsymbol{I}}{C} + \boldsymbol{\Omega}_{\mathrm{ELM}}\right)^{-1}\boldsymbol{T}\end{aligned} \tag{15}$$

After ELM was trained, the given testing sample$\boldsymbol{x}$ was taken as the input of the classifier. The index of the outputnode with the highest output value is considered as thepredicted class label of the given testing sample. Let $f_i(\boldsymbol{x})$ denotethe output function of the ith output node, the predicted class label ofsample$\boldsymbol{x}$is

$$label\left(\boldsymbol{x}\right) = \arg\max_{i\in\{1,\cdots,m\}} f_i\left(\boldsymbol{x}\right) \tag{16}$$

3.USER-SPECIFIED PARAMETERS

In this study, the popular Gaussian kernel function $K(\boldsymbol{u},\boldsymbol{v})=\exp(-\gamma\|\boldsymbol{u}-\boldsymbol{v}\|^2)$ is used as the kernel functionin KELM.In order to achieve good generalization performance, the costparameter$C$and kernel parameter$\gamma$of KELM need to be chosen appropriately. Similar to SVM and LS-SVM, the values of Candγ are assignedempirically or obtained by trying a widerange of$C$andγ. As suggested in (Huang, Zhou, Ding, et al., 2012), 50 differentvalues of$C$and 50 different values of$\gamma$are used for each data set, resulting in a total of2500 pairs of(C, γ). The 50 different values of $C$and$\gamma$are$\{2^{-24},2^{-23}, \ldots,2^{24},2^{25}\}$.This parameters optimization method is called Grid algorithm.

Similar to SVM and LS-SVM, the generalizationperformance of KELM with Gaussian kernel are sensitiveto the combination of(C, γ)as well. Thebest generalization performance of ELMwith Gaussian kernel is usually achieved in a very narrow rangeof such combinations. Thus, the best combination of(C, γ)of KELM with Gaussian kernel needs to bechosen for each data set.

Take Diabetesdata set for example, the performance sensitivity ofKELMwithGaussian kernel on the user-specified parameters(C, γ) is shown in Figure 1.

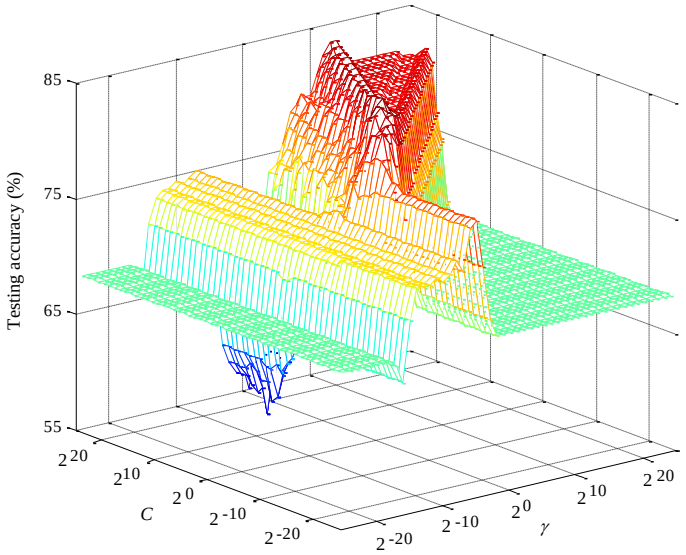

Figure 1. Performances of KELM with Gaussian kernel on Diabetes data set

The simulations were carriedout in MATLAB 7.0.1 environment running in Core 2 Duo 1.8GHZ CPU with 2GB RAM. Diabetesdata setis from the University of California, Irvine (UCI) machine learning repository, concluding 2 classes and 768 instances. In this case, 576 instances were selected randomly from Diabetesdata set as training data and the rest 192 instances were taken as testing data. As mentioned above, 50 differentvalues of$C$and 50 different values ofγ were used in this simulation. It can be seen from Fig. 1 that the performance of KELMwithGaussian kernel on Diabetesdata set is sensitiveto the user-specified parameters(C, γ) and the highest testing accuracy is obtained in a very narrow range ofthe combination of(C, γ). The time used for searching the best combination of these two parameters is 732.04 seconds; one of the best combinations of(C, γ) is (2^0, 2^1) and the corresponding testing accuracy is 83.85%.

4.OPTIMAL KELM WITH PSO

From practical point ofview, it may be time consuming and tedious for users to chooseappropriate kernel parameters(C, γ) by using the method mentioned in Section 3. What is more, the discretevalues of Candγ might result in suboptimaltesting accuracyalthough a widerange of$C$andγ have been tried (whichwill be discussed later in Section 5.1).In order to reduce time costs and achieve optimal generalization performance, the parameters in KELM with Gaussian kernel wereoptimized by using particle swarm optimization (PSO) in this paper.

4.1. Particle Swarm Optimization

Particle swarm optimization is a population-basedstochastic optimization technique developed by Eberhart andKennedy(Clerc and Kennedy, 2002). PSO simulates the social behavior of organisms,such as birds in a flock or fishes in a school, and can be describedas an automatically evolving system.

PSO works by initializing a flock of birds randomly over thesearching space, where every bird is called as a ‘‘particle’’. Theseparticles fly with a certain velocity and find the global bestposition after several iterations. During each iteration, each particleadjusts its velocity vector according to its momentum and theinfluence of its best position (P_b) as well as the best position of itsneighbors (P_g), and then a new position that the particle is to fly is obtained. Supposing the dimension of searching space is$D$, thetotal number of particles isn, the position of the ith particle canbe expressed as vector$\boldsymbol{X}_i=(x_{i1},x_{i2},\ldots,x_{iD})$; the best position of the$i$th particle searching until now is denoted as$\boldsymbol{P}_{ib}=(p_{i1},p_{i2},\ldots,p_{iD})$,and the best position of all particles searching until now isdenoted as vector$\boldsymbol{P}_g=(p_{g1},p_{g2},\ldots,p_{gD})$; the velocity of the $i$thparticle is represented as vector $\boldsymbol{V}_i=(v_{i1},v_{i2},\ldots,v_{iD})$. Then theoriginal PSO is described as

$$v_{id}(t+1)=\omega v_{id}(t)+c_1r_1[p_{id}(t)-x_{id}(t)]+c_2r_2[p_{gd}(t)-x_{id}(t)] \qquad (17)$$

$$x_{id}(t+1)=x_{id}(t)+v_{id}(t+1) \qquad i=1, 2, \ldots, n,\ d=1, 2, \ldots, D \qquad (18)$$

Wherec_1, $c_2$are the acceleration constants with positive values;r_1 and $r_2$are random number between 0 and 1. In addition to the$c_1$and$c_2$parameters, the implementation of the original algorithmalso requires to place a limit on the velocity ($v_{\max}$).The inertia weightω is used to balance the capabilities of global exploration andlocal exploration, which has a high value at the beginningand gradually lower later on. The following equation is used to determineω

$$\omega=\omega_{\min}+(\omega_{\max}-\omega_{\min})(t-1)\ /\ (T_e-1) \qquad (19)$$

where$\omega_{\max}$ is the initial inertia weight, $\omega_{\min}$ is the finalinertia weight, $t$is the current iteration and $T_e$is the epochparameter when inertial weight at final value.

4.2. Parameters Selection of KELM Using PSO

In PSO for parametersoptimization, the dimension of searching space is$D$=2 corresponding to the two parameters(C, γ) of KELM with Gaussian kernel, and the position of each particlerepresents the parametervalues of (C, γ) in Gaussian kernel. The aim of PSO for parametersoptimization is to obtain thebest generalization performance of KELM;thereforethe testing accuracy can be taken as the fitness function of PSO.

Theflowchart of this procedure is illustrated inFigure2.

The specific steps of PSO for KELM parametersoptimizationare described as follows.

Step1: Data preprocessing. All the attributes (except expected targets) of the classification dataset are normalized into the range [-1, 1] and then theclassification dataset is randomly divided into training and testing data in proportion.

Step2:Initialize the swarm size, maximum of iterations andvelocities. Generate randomly an initial velocity for eachparticle.

Step3:Evaluate each particle's fitness value according to the testing accuracy of KELM andset the best position from the particle with the maximal fitnessin the swarm.

Step4:Update the velocity and position for each candidateparticle by means of (17), (18) and (19) in each iteration.

Step5:Check the terminationcriterion. If the maximum numberof iterations is not yet reached, return to Step 3. Otherwisego to the next step.

Step6: Output the best combination of(C, γ)of KELM corresponding to the maximal fitness value.

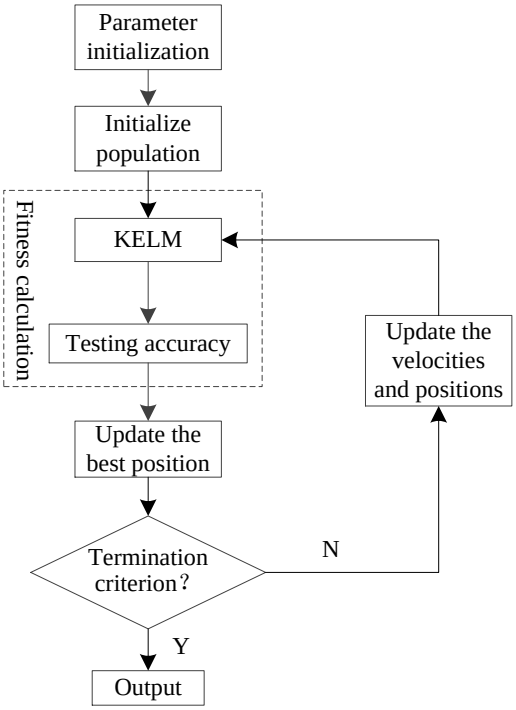

Figure1. Computational procedure of PSO for optimizing KELM

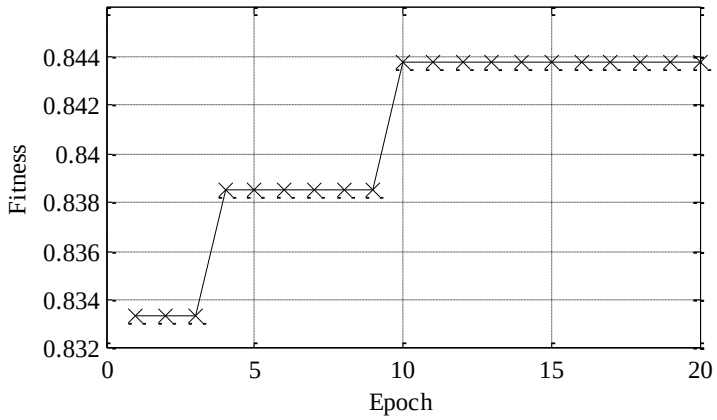

Figure 3. Fitness curve of PSO for KELM parameters optimization

6.EXPERIMENT RESULTS AND DISCUSSION

In this section, allsimulations on each data setsare carriedout in MATLAB 7.6 environment running in Core 2 Duo 1.8GHZ CPU with 2GB RAM. For KELM with PSO in all experiments,the range of costparameter$C$and kernel parameterγ were also [2^{-24}, 2^{25}] as mentioned in Section 3; population size was set to 24; maximum number of iterations (epochs) to train was set to 1000; acceleration constants$c_1$and$c_2$were set to 2; max particle velocityv_{max} was set to2^{10}; initial inertia weightω_{max} was set to 0.9 and final inertia weightω_{min} was set to 0.4; epochparameter T_e when inertial weight is at final value was set to 750. The training and testing data of all datasets are fixed for all trials of simulations.

For comparison with Grid algorithm, KELM with PSO was also tested on Diabetes data set. The training and testing data were the same as Grid algorithm mentioned in Section 3. The fitness curve of PSO for KELM parametersoptimization is shown in Figure 3.

It can be found from Fig. 3that the best fitness is obtained after 10 iterations. The best testing accuracy was 84.38% and the correspondingparameters(C, γ) were ($2^{1.913}$, $2^{0.8986}$). The best testing accuracy of KELM with Gaussian kerneloptimized by PSO is higher than that derived by Grid algorithm mentioned in Section 3. The time consumed by PSO for KELM parametersoptimization is 270.31 seconds, far less than 732.04 seconds spent by Grid algorithm. Compared with Grid algorithm,KELM with PSO achieves better generalization performance.

To verify the performance of KELM with PSO, simulations are also conducted on other eight benchmark classification datasets from the Universityof California, Irvine (UCI) machine learning repository. Specifications of the nine classification datasets (including Diabetes data set) are shown in Table 1.

Table 1. Specifications of classification datasets

Datasets	#classes	#attributes	#instances	#train	#test
Diabetes	2	8	768	576	192
Balance	3	4	625	376	249
Glass	7	9	214	139	75
Iris	3	4	150	102	48
Wine	3	13	178	121	57
Liver	2	6	345	231	114
Vowel	11	10	990	528	462
Waveform	3	21	5000	3002	1998
Breast-can	2	30	569	301	268

Table 2. Performance comparison of KELM optimized by different algorithms

Datasets	Grid algorithm				PSO			
	time (s)	C	γ	Testing accuracy (%)	time (s)	C	γ	Testing accuracy (%)
Diabetes	732.05	2^{0}	2^{1}	83.85	**270.31**	$2^{1.913}$	$2^{0.8986}$	**84.38**
Balance-scale	297.19	2^{21}	2^{14}	91.97	**118.02**	$2^{20.63}$	$2^{13.02}$	91.97
Glass	105.53	2^{13}	2^{-3}	73.33	**62.469**	$2^{16.57}$	$2^{-1.556}$	**74.67**
Iris	65.75	2^{-24}	2^{-20}	100	**21.297**	$2^{17.72}$	$2^{-11.78}$	100
Wine	103.5	2^{13}	2^{-5}	100	**46.641**	$2^{12.03}$	$2^{-5.635}$	100
Liver	132.81	2^{17}	2^{9}	75.44	**45.828**	$2^{13.62}$	$2^{6.672}$	75.44
Vowel	766.94	2^{4}	2^{0}	96.97	**525.98**	$2^{18.8}$	$2^{-0.3934}$	**97.19**
Waveform	31123	2^{-1}	2^{9}	86.49	**7454.3**	$2^{-2.764}$	$2^{9.305}$	**86.54**
Breast-can	541.52	2^{10}	2^{16}	97.01	**209.72**	$2^{19.77}$	2^{25}	97.01

The correspondingperformance of KELM optimized byGrid algorithm and PSO on the nine classification problems islisted in Tables 2, including the time consumed, parametersC andγ, and the best testing accuracy.

It can be found fromTables 2 that the computational time of PSO is far less than that of Grid algorithm and the best testing accuracies obtained by PSO is even higher than that derived by Grid algorithm.

Finally, KELM with PSO achieves better performance and is less time-consuming compared withGrid algorithm.

6.CONCLUSIONS

In this paper, the parameters of KELM are optimized by using PSO to improve the performance of KELM. Experimental results show that, compared with Grid algorithm on nine benchmark classification data sets, KELM optimized by PSO achieves better performance and is less time-consuming.

Acknowledgements

This paper was supported by the Doctoral Scientific Research Foundation of Northeast Dianli University (no. BSJXM-201401), China.

REFERENCES

G.B. Huang, Q.Y. Zhu, C.K. Siew (2006)"Extreme learning machine: Theory and applications",*Neurocomputing*, 70(1-3), pp.489-501.

G.B. Huang, Q.Y. Zhu, C.K. Siew(2004) "Extreme learning machine: A new learning scheme of feedforward neural networks",*Proc. Conf. on Neural Networks*, 2(25–29), pp.985-990.

G.B. Huang, C.K. Siew (2005)"Extreme learning machine with randomly assigned RBF kernels",*International Journal of Information Technology*, 11(1), pp.16-24.

G.B. Huang, H. Zhou, X. Ding, R. Zhang(2012)"Extreme learning machine for regression and multiclass classification",*IEEE Transactions on Systems, Man, and Cybernetics - Part B*: *Cybernetics*, 42(2), pp.513-529.

M. Clerc, J. Kennedy(2002)"The particle swarm-explosion, stability, and convergence in a multidimensional complex space",*IEEE Transactions on Evolutionary Computation*, 6(1), pp.58-73.