International Journal of Hybrid Information Technology
Vol. 9, No.12 (2016), pp. 27-36
http://dx.doi.org/10.14257/ijhit.2016.9.12.03

ELM-RBF Neural Networks using Micro-Genetic Algorithm for Optimization

Xianwei Xu and Sucheng Tian

Information Technology Department, Nanjing Forest Police College,210023,Nanjing,China
171918831@qq.com, 395668072@qq.com

Abstract

Thought Extreme Learning Machine-Radial Basis Function(ELM-RBF) can be used easily and can complete learning phase at very fast speed and provide more compact network than classical Extreme Learning Machine(ELM), it still has some room to improvement. Micro-Genetic Algorithms(uGA) improved calculated speed while inherits the Genetic Algorithms advantage of good for optimization and overall search. Considered on these, the paper designed a optimization strategy for ELM-RBF neural network based on uGA. In particular, based on classical RBF-ELM, we use real-uGA algorithm to optimize ELM-RBF hidden layer neurons center and biases value. Experiments results show that ELM-RBF-uGA has better recognition and prediction performance than classical ELM-RBF.

***Keywords**: Extreme Learning Machine; Optimization Strategy; Micro-Genetic Algorithms*

1. Introduction

People often build mathematical function to describe the task they facing, only after that, they can seek help from mathematic theory to solve problem. Now, in these fast-moving times, our task becoming more and more complex and need higher dimension, this situation making traditional polynomial algebra unable meet people's needs in practice.

Radial Basis Function (RBF) of its simple structure, high reliability and ease of project implementation, *etc*. has been widely researched. Broomhead [1] is the first researcher who applied RBF into neural network, and proposed a three levels structure of RBF neural network. Based on organism theory, Moody proposed a network with the characteristics of the local response [2], and in the essence, his network is same as Broomhead's network. After that, many approaches to improving the RBF network have been tried by researchers. In a word, given enough hidden layer neurons number, RBFNN can approximate to any continuous function.

Huang G.B[3~12] proposed an new learning algorithm named Extreme Learning Machine(ELM) for its ability of learn extremely fast, and has been a hot research area recently[13~17]. Based on excellent work of himself, he adopted RBF as ELM hidden nodes' activation functions, proposed ELM-RBF algorithm. This algorithm randomly set centers and widths value of ELM hidden layer neurons firstly; then, it calculates the linked weights between hidden layer neurons and output layer neurons. All this make ELM-RBF not only has RBF's advantage of not suffering from local minima, but also has ELM's fast learning speed. Further, to improve the generalization ability of network, literature[18~23] analyzed Differential Evolution algorithm (DE) and Particle Swarm Optimization algorithm(PSO), utilized them to optimize bias between hidden layer neurons and output layer neurons, and have better robustness.

ISSN: 1738-9968 IJHIT

Liu [24] proposed a new algorithm named as real-uGA, this algorithm is added a restart strategy based on traditional micro-genetic algorithm [25~28]. When premature happens, this restart strategy will produce a filial generation which has same size of current population, includes the best individual of current population, and randomly create rest individuals in search space. So, during the calculation process, whole real-uGA algorithm can be divided into several parallel sub-program, and every sub-program has its own local optima. When iteration increases, these sub-programs local optimas will approach the global optima, then, we will find the best answer.

Thought many algorithms have been proposed to improve the performance of RBF-ELM, for example, PSO-RBF-ELM, it is still exist room to improve this classical algorithm. So after studying a lot of literatures, based on their achievements ,this paper intend to adopt real-uGA to improve the generalization ability of RBF-ELM network, hope this new algorithm has merits of all three classical algorithm.

The remaining paper is organized as follows. Section 2 introduced related theories. Section 3 impressed our algorithm, including workflow. And section 4 presents the experiment results of proposed algorithm. Finally, Section 5 concludes this paper.

2. Related Theories

In literatures [3~5, 9, and 11], Professor Huang described his classical ELM-RBF network output as following formula (1):

$$\hat{y} = f(x) = \sum_{k=1}^{K} w_k \varphi_k(x) \tag{1}$$

$w_k = \left[w_{k,1}, w_{k,2}, \cdots w_{k,m}\right]$ is the linked weight between the *k*th neuron in hidden layer and the *m*th neuron in output layer. $\varphi_k(x)$ is hidden layer the *k*th neuron activation function which adopted RBF algorithm, and its specific form is shown as formula(2):

$$\begin{aligned}\varphi_k(x) &= \varphi(c_k, \sigma_k, x) \\ &= \exp\left(\frac{-\|x-c_k\|^2}{\sigma_k^2}\right)\end{aligned} \tag{2}$$

Here, $k = 1, 2, \cdots \mathrm{K}$, it represent the kth neuron of the hidden layer. x is input vector which has n dimensions. c_k is the kth neuron center of hidden layer. σ_k is the width of RBF.

Considered from formula (2), distance between input x and center c_k is represented by norm $\|\cdot\|$, that is, $\|x-c_k\|$ is the Euclid Norm between input x and center c_k. When $x = c_k$, $\varphi_k(x)$ has unique maximum. When $\|x-c_k\|$ increases, $\varphi_k(x)$ dies out to zero。 In whole input space, only data which close to center c_k can plays a part in whole algorithm, and that is why RBF-ELM has good receptivity to local data.

Existing *N* randomly training samples (x_i, y_i)

$$\begin{cases} x_i = \left[x_{i,1}, x_{i,2}, \cdots, x_{i,n}\right]^T \in R^n \\ y_i = \left[y_{i,1}, y_{i,2}, \cdots, y_{i,m}\right]^T \in R^m \end{cases}$$

The target of whole network is ELM-RBF network can approximate to these training samples with least error, when it has K hidden layer neurons, this task can be represented as following:

$$\min\left(\sum_{k-1}^{K} \left\| y_i - \hat{y}_i \right\|\right) \tag{3}$$

Transform it into formula (4):

$$\sum_{k-1}^{K}\left\| y_i - \hat{y}_i \right\| = 0 \tag{4}$$

Put it into formula (1), obtain formula (5)

$$\sum_{k=1}^{K} w_k \exp\left(\frac{-\left\|x-c_k\right\|^2}{\sigma_k^2}\right) = y_{i,}$$

$$i = 1,2,\cdots,N \tag{5}$$

Abbreviate formula(5),get formula(6)

HW=Y

$$\begin{cases} \mathrm{H} = \begin{bmatrix} \varphi(c_1,\sigma_1,x_1) & \cdots & \varphi(c_k,\sigma_k,x_1) \\ \vdots & \cdots & \vdots \\ \varphi(c_1,\sigma_1,x_N) & \cdots & \varphi(c_k,\sigma_k,x_N) \end{bmatrix}_{N\times \mathrm{K}} \\ \mathrm{W} = \begin{bmatrix} w_1^T \\ \vdots \\ w_k^T \end{bmatrix}_{\mathrm{K}\times m} \\ \mathrm{Y} = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N\times m} \end{cases} \tag{6}$$

H is output matrix of network hidden layer. When input vector is $x_1, x_2, \cdots, x_N$, the *k*th column in H is the output vector of the *k*th hidden layer neuron, and the *i*th row in H is the output vector of the x_i.

After simple derivation, to calculate W, we can get formula(7) as following:

$\mathrm{W} = \mathrm{H}^{+}\mathrm{Y}$

H^{+} is Moore-Penrose generalized inverse matrix of network hidden layer output matrix.

According to the proposed principle, ELM-RBF algorithm works flow is shown as following:

Input：[1].(x_i, y_i)：training sample set, and $x_i \in R^n, y_i \in R^m, i = 1,2,\cdots,N$

[2].K：the number of hidden layer neuron

Output：[1].ELM-RBF

Flow：

Step 1. Randomly set center value c_i and width value σ_i of

ELM-RBF hidden layer neurons, $i = 1,2,\cdots,\mathrm{K}$;

Step 2. Calculate network hidden layer output matrix H;

Step 3. Calculate linked weight matrix W between hidden layer and output layer $\mathrm{W} = \mathrm{H}^{+}\mathrm{Y}$, configure network, and use ELM-RBF to do recognition task we faced.

3. The Proposed Method (ELM-RBF-uGA)

The merit of ELM-RBF algorithm is extremely fast learning speed, and some generalization ability, but for its hidden layer neurons center value and width value is set randomly, obviously, they are not optimal[4~11]. In engineering, to improve generalization ability, we must add more hidden layer neurons, that leads to network's structure becomes more complex and computation is raised too. So, the problem becomes how to reduce hidden layer neuron numbers and simplify network structure when network performance can meet the request of our task. Then, optimization algorithms are introduced into ELM-RBF by researchers.

The starting point of our design is how to improve ELM-RBF generalization ability after its hidden layer neuron number is set, more broadly, its computational cost should not be increased obviously. Here, this paper presents a mixed algorithm which combines uGA algorithm and ELM-RBF algorithm, and we named it ELM-RBF-uGA. This algorithm adopts uGA algorithm to optimize ELM-RBF hidden layer neurons' center and width, improve the whole performance of ELM-RBF, it workflow is shown as following:

Input：[1]. (x_i, y_i) is training sample set；

$$\begin{cases} x_i = \left[x_{i,1}, x_{i,2}, \cdots, x_{i,n}\right]^T \in R^n \\ y_i = \left[y_{i,1}, y_{i,2}, \cdots, y_{i,m}\right]^T \in R^m \end{cases}$$

[2]. $val(i, j)$:the fitness of the ith generation and the jth population;

[3]. $\Delta val(i+1, j)$： the difference between same (jth) population's adjacent (ith) two generations' fitness.
$\Delta val(i+1, j)=val(i+1, j)-val(i, j)$;

[4]. $flag_j^r$:the rth reboot, the jth population's current optimal counter;

[5]. P_{op} :optimal population;

[6].K: hidden layer neuron number;
[7].m:reboot times;
[8].Rmax: max reboot time;
[9].Ts1:reboot threshold, in rth reboot process, only populations whose $\Delta val(i+1, j) \le Ts1$ or $flag_j^r \ge 1$ will be reserved in reboot step;
[11].Trj: the time of one population has been reserved;
[10].Ts2:if exist a $flag_{\text{j}}^r \ge Ts2$, algorithm over;

Output：Optimal ELM-RBF network.

Workflow：

Step 1：Randomly generate N initial populations $P_{1,j}$, $j=1,\cdots,N$;these populations contains information of networks hidden layer neurons center $c_{i,j}$ and width $\sigma_{i,j}$, and set all $flag_j^r=0$ $r=0$, $j=0, Ts_j=0$;

Step 2:Adopt Moore-Penrose generalized inverse theory to calculate every populations $\text{H}_{i,\ j}$, here i means the *i*th generation, j means the jth population. According to formula(5) and (6), calculate every populations output layer's linked weight matrix $\text{W}_{i,j}$ .Then, use populations' Root Mean Square Error(RMSE) as fitness function $val(i, j)$;

Step 4: Calculate the difference between same (*j*th) population's adjacent (ith) two generations' fitness.

$$\Delta val(i+1, j)=val(i+1, j)-val(i.j)$$

Step 5：If $\Delta val(i+1, j) < Ts1$,then, $flag_j^r = flag_j^r + 1$,goto Step6; If not, use traditional GA algorithm, except the optimal population in the *i*th generation，all rest population performs crossover, mutation, and so generate new generation $\text{P}_{i+1,j}$,and goto step2;If exceed traditional single GA iteration maximum, goto Step 12;

Step6: If *j*th populations' $flag_{\text{j}}^r \ge Ts2$, then , $r=r+1, Ts_j = Ts_j + 1, m = m+1$;

Step7: If exist $Ts_j \ge 2$, it means a population has been reserved in former reboot process, and it will be reserved in this reboot process again, goto Step 11;

Step8: If $m \ge R\max$, it means max iteration time is exceed, goto Step 11;

Step9: Reboot algorithm, only populations whose $flag_j^r \geq 1$ will be reserved, randomly generate new population, and all new generated populations' $flag_j^r = 0$; goto step2;

Step10: Warning, goto Step12;

Step 11：Output optimal population P_{op}, set optimal c_{op} and σ_{op}, then, use Adopt Moore-Penrose generalized inverse theory to calculate optimal W_{op}；Set optimal ELM-RBF by $c_{op}, \sigma_{op}, W_{op}$;

Step 12：Over.

Algorithm workflow is shown as Figure 1. During its evaluation, this algorithm calculates and reserve population with best fitness in each generation, then, in order to keep population diversity and avoid premature convergence of traditional genetic algorithm, it utilize a new reboot strategy ,randomly re-generate new populations when the best population is reserved. As a result, its ability to escape from local optimum is enhanced and the global search ability of the ELM-RBF algorithm is strengthened.

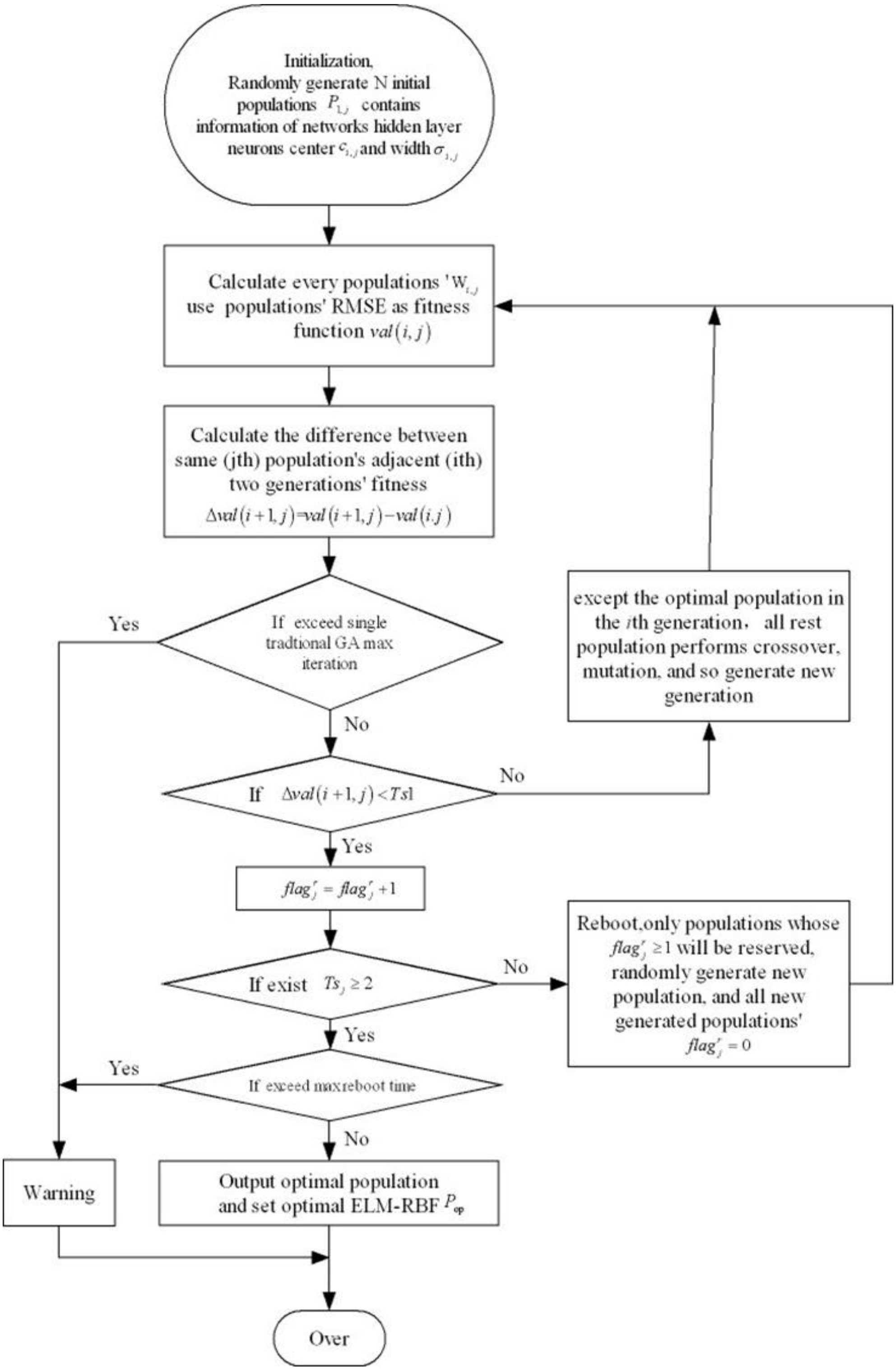
Initialization,
Randomly generate N initial populations $P_{1,j}$ contains information of networks hidden layer neurons center $c_{1,j}$ and width $\sigma_{1,j}$
Calculate every populations 'W$_{i,j}$ use populations' RMSE as fitness function $val(i,j)$
Calculate the difference between same (jth) population's adjacent (ith) two generations' fitness
$\Delta val(i+1,j)$=$val(i+1,j)-val(i.j)$
If exceed single tradtional GA max iteration
Yes
No
If $\Delta val(i+1,j)<Ts1$
No
Yes
except the optimal population in the ith generation， all rest population performs crossover, mutation, and so generate new generation
$flag_j^r=flag_j^r+1$
If exist $Ts_j\geq 2$
No
Yes
Reboot,only populations whose $flag_j^r\geq 1$ will be reserved, randomly generate new population, and all new generated populations' $flag_j^r=0$
Yes
If exceed maxreboot time
No
Warning
Output optimal population and set optimal ELM-RBF P_{op}
Over

Figure 1. ELM-RBF-uGA Workflow

4. Experiments Data

Test 1.We used the literature's[5] method to test recognition performance of proposed algorithm, we adopted test data from UCI Machine Learning Repository[29], worked by two pattern classification benchmark problems, Diabetes and Iris, calculate its Learning Set Accuracy(LSA), Test set Accuracy(TSA), Standard Deviation(SD), and see the experimental datum in Table 1.

Table 1. Comparisons of ELM-RBF-uGA and ELM-RBF

Algorithm	Diabetes		Iris	
	LSA/SD	TSA/SD	LSA/SD	TSA/SD
ELM-RBF-uGA	0.7512/ 0.0098	0.8211/ 0.0187	0.9472/ 0.0287	0.8672/ 0.0519
ELM-RBF	0.7163/ 0.0156	0.7052/ 0.0310	0.8382/ 0.1107	0.7753/ 0.1294

Here, our experiment's important parameters of ELM-RBF-uGA listed in Table 2.

Table 2. Parameters

Populations	Ts1	Ts2	Max iteration	Reboot times
50	0.001	3	1000	200

Analyzing result of simulated experiment data, we can see, no matter to learning dataset or test dataset, the accuracy and standard deviation of ELM-RBF-uGA is smaller than traditional ELM-RBF, so it is obviously that ELM-RBF-uGA has better recognition performance of object classification than traditional ELM-RBF.

Test 2. To test the prediction accuracy of proposed algorithm, we adopted the sunspots model prediction standard. This data set contains 307 groups of data, and we selected 199 data as input. In order to prove the prediction effect, we introduce the Mean Absolute Percentage Error (MAPE).

$$MAPE = \frac{1}{n}\sum_{t=1}^{1}\left|\frac{A_t - F_t}{A_t}\right|$$

Where A_t is the actual value and F_t is the forecast value.

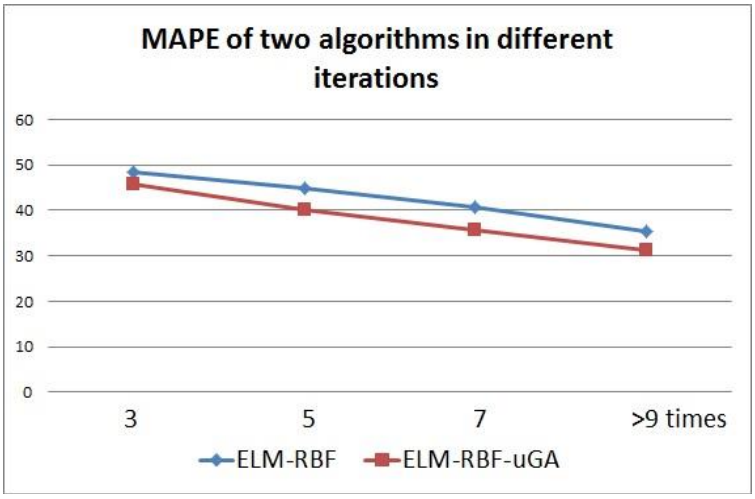

Figure 2. MAPE of Two Algorithms in Different Iterations

From Figure 2 we can see, data of ELM-RBF-uGA show better performance than data of ELM-RBF, so the prediction accuracy of proposed algorithm is better than classical ELM-RBF.

To further summarizes the prediction performance of proposed algorithm. It is clear that the performance prediction of proposed algorithm is better than ELM-RBF. Figure 3 presents the prediction results of sunspots data by these two algorithms.

It is obviously that data of ELM-RBF-uGA is much closer to Source data than data of ELM-RBF, so, it has better prediction performance than that.

The reason is that our optimization strategy plays a role, it is better than traditional ELM-RBF whose hidden layer neurons is randomly generated. But, it should be noted that ELM-RBF-uGA has much larger amount computation than traditional ELM-RBF.

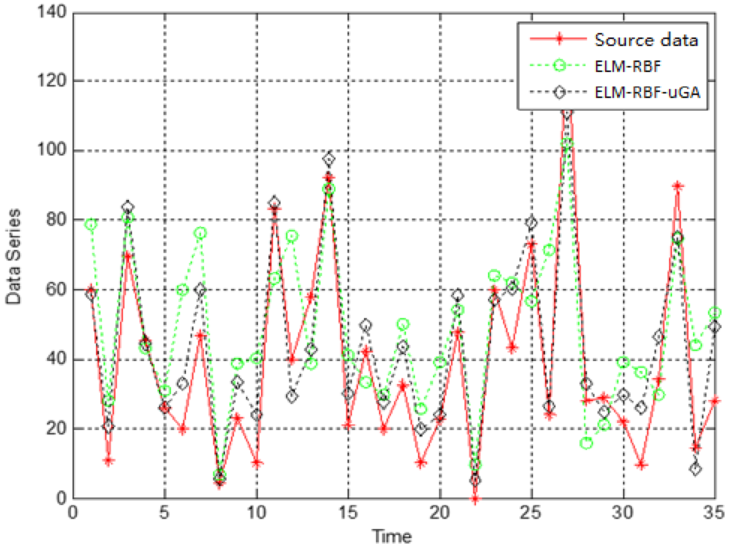

Figure 3. Data of Several Algorithms Prediction

5. Conclusion

This paper intends to adopt real-uGA to improve the generalization ability of RBF-ELM network, hope this new algorithm has merits of all three classical algorithms. The experiments data prove that, compared to traditional ELM-RBF, ELM-RBF-uGA shows stronger ability in recognition and prediction.

Acknowledgements

Whatever implementation we come up with needs to be based on classical ELM-RBF, and we cannot execute our work without Professor Huang’s earlier work and his open-source code online. Thanks to Professor Huang.

The authors would like to thanks the anonymous reviewers whose comments greatly improved the paper.This paper was funded under a grant from the Fundamental Research Funds for the Central Universities（No LGYB201410）.

Reference

[1] D. S. Broomhead and D. Lowe, “Multivariable functional interpolation and adaptive networks”, Complex Systems, vol. 2, **(1988)**, pp. 321-355.

[2] J. E. Moody and C. J. Darken, “Fast learning in networks of locally tuned processing units”, NeuralComputation, vol. 1, no. 2, **(1989)**, pp. 281-294.

[3] G. B. Huang, Q. Y. Zhu and C. K. Siew, “Extreme learning machine: theory and applications”, Neurocomputing, vol. 70, **(2006)**, pp. 489-501.

[4] G. B. Huang and C. K. Siew, “Extreme learning machine with randomly assigned RBF kernels”, International Journal of Information Technology, vol. 11, no. 1, **(2005)**, pp. 16-24.

[5] B. Li ，L. Bin and X. Rong, “Intelligent optimization strategy for ELM-RBF neural network”, Journal of Shandong University (Natural Science), vol. 45, no 5, **(2010)**, pp. 48-52.

[6] G. B. Huang, Q. Y. Zhu and C. K. Siew, “Extreme learning machine: theory and applications”, Neurocomputing, vol. 70, no. 1, **(2006)**, pp. 489-501.

[7] G. B. Huang, Q. Y. Zhu and C. K. Siew, “Extreme learning machine: a new learning scheme of feedforward neural networks”, Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on. IEEE, vol. 2, **(2004)**, pp. 985-990.

[8] G. B. Huang, H. Zhou and X. Ding, “Extreme learning machine for regression and multiclass classification”, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, vol. 42, no. 2, **(2012)**, pp. 513-529.

[9] G. B. Huang, X. Ding and H. Zhou, “Optimization method based extreme learning machine for classification”, Neurocomputing, vol. 74, no. 1, **(2010)**, pp. 155-163.

[10] G. B. Huang and C. K. Siew, “Extreme learning machine with randomly assigned RBF kernels”, International Journal of Information Technology, vol. 11, no. 1, **(2005)**, pp. 16-24.

[11] G. B. Huang and C. K. Siew, “Extreme learning machine: RBF network case”, Control, Automation, Robotics and Vision Conference, **2004**. ICARCV 2004 8th. IEEE, vol. 2, **(2004)**, pp. 1029-1036.

[12] K. Hu, A. Song and M. Xia, “An adaptive filtering algorithm based on genetic algorithm-backpropagation network”, Mathematical Problems in Engineering, 2013, **(2013)**.

[13] M. Xia and W.K. Wong, “A seasonal discrete grey forecasting model for fashion retailing”, Knowledge-Based Systems, vol. 57, **(2014)**, pp. 119-126.

[14] S. Aiguo and L. Jiren, “Evolving Gaussian RBF network for nonlinear time series modelling and prediction”, Electron Lett, vol. 34, no. 12, **(1998)**, pp. 1241-1243.

[15] K. Hu, A. G. Song and Y. C. Zhang, “Delay-range-dependent Stability Criteria of Neural Networks with Time-varying Discrete and Distributed Delays”, Int J Adv Robot Syst, vol. 11, **(2014)**, pp. 23.

[16] K. Hu, A.G. Song and W. L. Wang, “Fault detection and estimation for non-Gaussian stochastic systems with time varying delay”, Advances in Difference Equations, vol. 2013, no. 1, **(2013)**, pp. 1-12.

[17] X. Wen and A. Song, “An immune evolutionary algorithm for sphericity error evaluation”, International Journal of Machine Tools and Manufacture, vol. 44, no. 10, **(2004)**, pp. 1077-1084.

[18] A. Song, Y. Han and H. Hu, “Active perception-based haptic texture sensor”, Sensors and Materials, vol. 25, no. 1, **(2013)**, pp. 1-15.

[19] M. Xia, Y. Zhang, L. Weng and X. Ye, “Fashion retailing forecasting based on extreme learning machine with adaptive metrics of inputs”, Knowledge-Based Systems, vol. 36 **(2012)**, pp. 253–259.

[20] A. Song, Y. Han and H. Hu, “Active perception-based haptic texture sensor”, Sensors and Materials, vol. 25, no. 1, **(2013)**, pp. 1-15.

[21] M. Xia, Z. Wang and J. Fang, “Temporal association based on dynamic depression synapses and chaotic neurons”, Neurocomputing, vol. 74, **(2011)**, pp. 3242–3247.

[22] M. Xia, L. Weng and X. Ye, “Sequence Memory based on Ordered Pattern Interrelation”, Advanced science letters, vol. 5, no. 2, **(2012)**, pp. 547-551(5).

[23] Z. Xiong, “Some problems in approximation theory with radial basis function”, Fudan University, **(2007)** (In Chinese).

[24] G. Liu, “Multi-objective Optimization methods based on Micro Gentic Algorithm and Applications”, Hunan University, **(2008)** (In Chinese).

[25] J H. Holland, “Genetic algorithms and the optimal allocation of trials”, SIAM Journal on Computing, vol. 2, no. 2, **(1973)**, pp. 88-105.

[26] K. Krishnakumar, “Micro-genetic algorithms for stationary and non-stationary function optimization”, In SPIE Proceedings: Intelligent Control and Adaptive systems, **(1989)**, pp. 289-296.

[27] Y. G. Xu, G. R. Liu and Z. P. Wu, “A novel hybrid genetic algorithm using local optimizer based on heuristic pattern move”, Appl. Artif. Intelligence, vol. 15, no. 7, **(2001)**, pp. 601-613.

[28] G. R. Liu and H. J. Ma, “Material characterization of composite laminate using dynamic response and real parameter coded micro genetic algorithm”, Eng. Comput., **(2003)**.

[29] http://archive.ics.uci.edu/ml/.

Authors

Xianwei Xu, she was born in 1977, Yangzhou city, Jiangsu province, China, and got Master degree in Nanjing University of Information Science and Technoloy at 2006. Now she is a lecturer in Information Technology Department, Nanjing Forest Police College. Her research interests include computer crime and computer vision.

Sucheng Tian, he was born in 1970, Shandong province, China, and got Bachelor degree in Northeast Forestry University. Now he is an associate professor in Information Technology Department, Nanjing Forest Police College. His research interests include internet security and computer vision.