An Extreme Learning Machine Based on Quantum Particle Swarm Optimization and its Application in Handwritten Numeral Recognition

Xin Sun, Liangxi Qin

School of Computer, Electronics and information

Guangxi University

Nanning, China

E-mail: sunxinhb@126.com

Abstract—**The Extreme Learning Machine algorithm was proposed by Prof. Guangbin Huang in 2004. It is a single hidden layer feedforward neural network. It has attracted extensive research of many scholars because of its fast speed, simple implementation and good generalization performance. In this paper, Quantum Particle Swarm Optimization was introduced to extreme learning machine to solve the problem of complex network structure which is caused by random assignments to the input weights and biases of hidden nodes .The QPSO is used in the process to select the input weights and biases instead of random assignment. Then extreme learning machine uses the result produced by QPSO to train the network. Thus can improve the prediction accuracy and response speed to unknown data and gain a more compact network structure. The proposed method is used in handwritten numeral recognition application in the end. And it gets an approving performance.**

Keywords-extreme learning machine； network structure； Quantum Particle Swarm Optimization； prediction accuracy; Handwritten Numeral Recognition

I INTRODUCE

The Extreme Learning Machine is a single hidden layer feedforward neural network which was proposed by Prof. Guangbin Huang in Nanyang Technological University in 2004[1]. Compared to the traditional gradient-based learning algorithms, it has faster learning speed, better generalization performance and simpler implementation. It also can avoid trapping in the local optimal value. These all because that it randomly assign values to the input weights and biases of **hidden nodes**, calculates the output weights by using the Moore Processor generalized inverse of hidden output weights. It needs computation only once. ELM is widely used in classification and regression. It is found that the extreme learning machine have a more complex network structure and slower response speed than BP algorithm because it assigns the values of input weights and biases randomly [2].

Quantum Particle Swarm Optimization (QPSO) was proposed by Jun Sun in 2004[3].It is a swarm intelligence optimization algorithm which is combining the quantum evolutionary perspective with swarm particle optimization (PSO). It has stronger global search capability, higher search precision, less control parameters than PSO algorithm. It is widely used in the fields of clustering and classification [4], neural network [6-7] and combination optimization.

In this paper, it is proposed an new extreme learning machine model which combine QPSO with extreme learning machine. It used QPSO in the process of selecting the input weights and biases instead of random selection. It can gain more effective input weights and biases to improve ELM performance. The method is based on comprehensive study of the QPSO algorithm and ELM. Then using the method to construct a classifier for handwritten numeral recognition problem to prove it can get a better performance in real-world application.

The rest of the paper is organized as follows. In section 2, it gave a brief introduction to ELM and QPSO. The Extreme Learning Machine based on Quantum Particle Swarm Optimization(QPSO_ELM) was proposed in Section 3. In Section 4, it applied the method to solve the handwritten numeral recognition problem. Finally, the conclusion was given in Section 5.

II PRELIMINARIES

A Extreme Learning Machine

According to the paper [5], the thoughts and steps of extreme learning machine are shown as follow: Suppose the ELM has L hidden nodes and activation function f(x).we use it to training N distinct samples as $\{(x_i,t_i)|\, x_i \in R^n, t_i \in R^m, i=1,2,\cdots N\}$.

1. Randomly assign a value to each input weight a_i and bias b_i, i=1, 2 …L;

2. Calculate the hidden output weights matrix H according to equation (1);

$$\mathrm{H}=\begin{bmatrix} f(a_1x_1+b_1)\cdots f(a_Lx_1+b_L) \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ f(a_1x_N+b_1)\cdots f(a_Lx_N+b_L)\end{bmatrix}_{N\times L} \tag{1}$$

978-1-4799-3279-5 /14/$31.00 ©2014 IEEE

3. Calculate the network output weights β according to equation (2);

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

Where H^+ is the MP generalized inverse of matrix H and T is indicating the expect output as $T = \{t_1, t_2, \cdots t_N\}^T$.

We can learn that the only parameter which is need user to set is the hidden nodes number.

B Quantum Particle Swarm Optimization

QPSO is a new particle swarm optimization based on PSO and DELTA trap. Assuming that there are M particles of potential solutions in an n-dimensions target space as $X(t)=\{X_1(t),X_2(t), \cdots \cdots ,X_M(t)\}$,where $X_i(t)$ indicates the position of the i-th particle at t-th iteration. The best individual optimal of i-th particle at t-th iteration is represented as $X_{ibest}(t)=\{X_{ibest1}(t), X_{ibest2}(t),\cdots\cdots,X_{ibestn}(t)\}$ and the best global optimal at t-th iteration is represented as $X_{gbest}(t)=\{X_{gbest1}(t), X_{gbest2}(t), \cdots \cdots ,X_{gbestn}(t)\}$. Considering the minimization problem, the smaller the objective function value is, the better the corresponding fitness value is. Thus the best individual optimal position can be determined by equation (3):

$$p_{ibest}(t+1) = \begin{cases} x_i(t) & f[x_i(t)] < f[p_{ibest}(t+1)] \\ p_{ibest}(t) & f[x_i(t)] \geq f[p_{ibest}(t+1)] \end{cases} \tag{3}$$

Where f() represents the fitness function. The global optimal position is represented as:

$$\begin{aligned} g &= \arg\min_{1 \le i \le M} \{f[P_i(t)]\} \\ P_{gbest} &= P_g(t) \end{aligned} \tag{4}$$

The position of particles is updated by formula (5) ~ (8)

$$x_{id}(t+1) = p + b \times |mbest_d - position_d| \times \ln\left(\frac{1}{u}\right) \tag{5}$$

$$p = a \times x_{ibestd}(t) + (1-a) \times x_{gbestd} \tag{6}$$

$$mbest = \frac{1}{M} \times \sum_{i=1}^{M} x_{ibest} \tag{7}$$

$$b = 1 - 0.5 \times \frac{t}{Maxgen} \tag{8}$$

Where p represents a random position between best individual optimal position and best global optimal position, the mbest represents the average position of all best individual optimal position, a and u are random numbers between 0 and 1, Maxgen stands the maximum number of iterations and t stands the current iteration.

III AN EXTREME LEARNING MACHINE BASED ON QUANTUM PARTICLE SWARM OPTIMIZATION

In order to solve the problem that the structure of extreme leaning machine is more complex than ordinary neural network, in the paper [8], it was proposed an ELM based on Differential Evolution(E_ELM), and in [9], it was proposed an ELM based on Particle Swarm Optimization (PSO_ELM). They are introduced to make the input weights and biases more effective.

In this paper, it was proposed an ELM based on QPSO (QPSO_ELM) because of the advantages of QPSO, such as fewer parameters, stronger optimization capability and avoiding falling into local optimal value. The basic thought is using QPSO to gain input weights and biases before ELM training, then uses the optimal result to train and test the ELM performance.

QPSO_ELM learning steps are as follows:

1. Randomly generate particles. The dimension of each particle is determined by equation (9);

$$Dimension = (NumberofInuput + 1) * L \tag{9}$$

Where NumberofInput represents the number of input features, L represents the number of hidden nodes. The i-th particle is represented as $p_i = [w_{11}, w_{12}, \cdots, w_{1n}, w_{21}, w_{22}, \cdots, w_{2n}, \cdots, w_{H1}, w_{H2}, \cdots, w_{Hn}, b_1, b_2, \cdots, b_{H,}]$, where n represents the number of training samples .

2. Calculating fitness of each particle. According to paper [10], in order to avoid over-fitting, dividing the testing dataset into two distinct parts as testing dataset and validating dataset. It is taken the root mean square error(RMSE) as the fitness function on regression problem, and taken the sample identification error as the fitness function on classification problem.

$$fit = \begin{cases} \dfrac{MissClassification\_training}{n_v} & classification \\ \sqrt{\dfrac{\sum_{j=1}^{n_v} \left\| \sum_{i=1}^{H} \beta_i * (w_i \cdot x_j + b_j) - t_j \right\|_2^2}{n_v}} & regression \end{cases} \tag{10}$$

Where MissClassification_training indicates the sample number of wrong identification, n_v represents the number of validating data, β_i indicates output weights of i-th sample.

3. According to paper [10] and [11], the introduced structural risk can enable the network to have better generalization performance. Thus we introduce structural risk to calculate the best global optimal and individual optimal as equation (11) and (12).

$$p_{ibest}(t+1) = \begin{cases} p_i(t) & \begin{array}{l} (f(p_{ibest}(t)) - f(p_i(t))) > \eta * f(p_{ibest}(t)) \| \\ |f(p_{ibest}(t)) - f(p_i(t))| < \eta * f(p_{ibest}(t)) \\ \&\& \|\beta_{p_i(t)}\| < \|\beta_{p_{ibest}(t)}\| \end{array} \\ p_{ibest}(t) & else \end{cases} \tag{10}$$

$$p_{gbest}(t+1)=\begin{cases} & \left(f(p_{gbest}(t))-f(p_i(t))\right)>\eta * f(p_{gbest}(t))\| \\ p_i(t) & \left|f(p_{gbest}(t))-f(p_i(t))\right|<\eta * f(p_{gbest}(t)) \\ & \&\&\|\beta_{p_i(t)}\|<\|\beta_{p_{gbest}(t)}\| \\ p_{gbest}(t) & else \end{cases} \quad (11)$$

Where η >0 is a tolerance rate.

4. Update particles position by using (5)~ (8).

5. Boundary processing. According to the literatures [1, 5, 12], all dimensions in the particle should be limited to [-1, 1], as (13).

$$x_{id}=\begin{cases} x_{\max} & x_{id}>x_{\max} \\ x_{\min} & x_{id}<x_{\min} \end{cases} \quad d\in[1,\dim ension] \quad (12)$$

6. Fitness value of each particle is calculated according to (10), and in accordance with (11) to update best individual optimal position and best global optimal position;

7. Determining whether the intended accuracy or max number of iterations is reached, it is then go to step 8, otherwise go to step 4 to continue the iterations and add one to t.

8. Using the best particle position to train ELM and output the training time, testing time, training accuracy and testing accuracy.

We compare the result of QPSO_ELM, PSO_ELM and E_ELM on sinc regression.

A training dataset(x_i,y_i) and testing dataset(x_j,y_j) with 1000 data, where xi's and x_j's are uniformly randomly distributed on the interval(-10,10). We add the noise distributed in [-0.2, 0.2] to training samples. All input feature normalized into [-1,1].The validation data set is taken the half data from the test data, and the rest data is used for testing performance. The result of the four algorithms is shown on TABLE I.

TABLE I THE RESULT ON SINC FUNCTION

Algorithm	Training time	Training error	Testing error	hidden node
ELM	0.0243	0.1138	0.0197	30
E_ELM	14.8713	0.1150	0.0137	10
PSO_ELM	14.4404	0.1150	0.0123	10
QPSO_ELM	14.5511	0.1151	0.0117	10

From table 1, we can see that E_ELM, PSO_ELM and QPSO_ELM all have a long training time than ELM. But they all have a smaller RMSE, a more compact network structure and a faster response speed to unknown data than ELM. In the three evolutionary ELM, QPSO_ELM have a less RMSE than others. It is because QPSO have a stronger Optimization ability.

IV APPLICATION IN HANDWRITTEN NUMERAL RECOGNITION

In this paper, the 1000 training data and the 2000 testing sample of the experiment are respectively selected from training and testing dataset of the USPS handwriting numeral. The dimensions of input feature is 256 and the output categories is 10. So the network has 256 input nodes and 10 output nodes. TABLE II shows the number of training sample and testing sample for every category.

TABLE II THE NUMBER OF TRAINING SAMPLE AND TESTING SAMPLE FOR EVERY CATEGORY

Class	0	1	2	3	4	5	6	7	8	9
Training sample	163	140	116	94	101	68	95	77	73	73
Testing sample	250	201	174	135	147	109	133	112	120	119

In this experiment, all programs are run in the MATLAB2009 environment and all inputs are normalized into [-1, 1]. It uses logsig function as the activation function of hidden layer, that is $f(x)=\frac{1}{1+\exp(-x)}$.The input weights and biases of ELM are set into [-1, 1]. We had run 50 times for every algorithm in this experiment to avoid generating occasional results, and the experimental results are the average of the 50 times results.

We mainly compare the method of this paper proposed with ELM and BP neural network in the handwriting digital recognition application. We uses the BP neural network tools box of MATLAB as BP neural network. The max iterations is set to 5000, learning speed is 0.05 per step, the expected error is 0.01. The parameters of QPSO_ELM are set as follows: the particles' number is set to 20, the max iterations is set to 80, the tolerance of structural risk is set to 0.02, validation set are half data extracted from testing set. Table 2 shows the training accuracy and testing accuracy of these three algorithms.

TABLE III THE CLASSIFICATION RESULT ON HANDWRITTEN NEURAL RECOGNITION

digit	Testing accuracy			Training accuracy		
	ELM (%)	*QPSO_ELM (%)*	*BP (%)*	*ELM (%)*	*QPSO_ELM (%)*	*BP*

0	96.16	94.92	93.43	99.95	95.80	99.55
1	97.61	99.70	96.39	100	99.47	99.83
2	78.14	90.14	72.93	99.29	86.62	99.52
3	79.95	89.80	78	99.72	91.38	99.55
4	90.22	92.04	92.57	98.93	86.38	99.17
5	77.76	82.29	78.54	99.38	70.18	98.03
6	97.26	98.53	96.21	99.28	89.07	99.71
7	78.57	90	78.46	98.7	89.53	98.99
8	75.44	73.57	74.55	97.75	72.41	98.08
9	84	84.26	82.22	99.37	78.71	99.89
Total result	86.65	90.17	85.2	99.36	88.06	99.33

From TABLE III, we can see that the testing accuracy of ELM is higher than BP neural network at the same training accuracy, except 4 and 5. That is because ELM can effectively avoid falling into local optimum and over-fitting , so that it can obtain better testing performance. Although all the training accuracies of QPSO_ELM are lower than ELM and BP neural network, the testing accuracies of it are higher than BP neural network, except 4. Comparing with ELM, the testing accuracies of QPSO_ELM is higher except 1 and 8.That is because QPSO_ELM optimized the input weights and biases by using Quantum Particle Swarm Optimization to get efficient input weights and biases. From the total classification result, we can see the testing accuracies of ELM is higher than BP neural network while QPSO_ELM is higher than ELM. QPSO_ELM can obtain the best testing performance.

TABLE IV THE TRAINING TIME, TESTING TIME AND THE NUMBER OF HIDDEN NODES OF THREE ALGORITHMS

Algorithm	Training time	Training dev	Testing time	Testing dev	Hidden node
ELM	1.1070	0.2075	0.0574	0.0239	300
QPSO_ELM	306.6197	34.8975	0.0075	0.0131	50
BP	257.6812	48.2755	0.0390	0.0286	20

TABLE IV shows the training time, testing time and the number of hidden nodes of three algorithms. From the table we can see the training time of ELM is far less than BP and QPSO_ELM, and QPSO_ELM is the longest. That is because ELM only needs one calculation, while BP neural network needs to train repeatedly and QPSO_ELM needs to spend most time on the optimization algorithm. But the testing time is completely opposite. The longest is ELM, the second is BP neural network, and QPSO_ELM is the shortest. The reason why the testing time of QPSO_ELM is shorter than BP neural network is that testing data of QPSO_ELM is half of BP. The reason why the testing time of QPSO_ELM is shorter than ELM is that it needs fewer hidden nodes than ELM. The fewer nodes it needs, the faster response speed it is.

Generally, comparing with ELM, although the training time of QPSO_ELM is longer, it can obtain higher testing accuracy and more compact network structure, and it has faster response speed to unknown data. Comparing with BP neural network, it can avoid falling into local optimal and has higher testing accuracy for unknown data.

V CONCLUSION

In this paper, it is proposed a new extreme leaning machine model which is combining QPSO with ELM. It not only can avoid falling into local optimal value and over-fitting and has a better generation performance, but also has a more compact structure and a faster response speed to unknown data. It is a new method to solve handwritten numeral recognition problem.

REFERENCES

[1] G.-B.Huang, Q.-Y. Zhu, Siew CK.Extreme learning machine:A new learning scheme of feedforward neural networks[J].2004 IEEE INTERNATIONAL JOINT CONFERENCE ON NEURALNETWORKS, 2004, 1(4):985-990.

[2] Q.-Y. Zhu,A.K. Qin,P.N. Suganthan,G.-B. Huang.Evolutionary extreme learning machine[J]. PatternRecognition,2005,38(10):1759-1763.

[3] Sun J,Feng B,XuWB.Particle Swarm Optimization with Particles Having Quantum Behavior[A]. Proceedings of 2004 Congress on Evolutionary Computation, 2004: 325-331.

[4] Tao Liu,Feng Yin,Jianying Chen.Obtaining the Classification Rules of Particle Swarm Optimization Algorithm Based on Quantum Behavior [J].Application Research Of Computers, 2009, 26(2):496-499.

[5] G.-B. Huang,Q.-Y. Zhu,C.-K. Siew.Extreme learning machine : theory and applications[J]. Neurocomputing,2006 ,70:489-501.

[6] Ruhui M, Yuan L, Xing L. Network anomaly detection using RBF neural network with hybrid QPSO[C]. IEEE International Conference on Networking,Sensing and Control,2008,1284-1287.

[7] Sun J,Xu W,Liu J.Training RBF neural network via quantum-behaved particle swarm optimizaiton[C]. In:Neural Information Processing,2006,1156-1163.

[8] Q.-Y. Zhu, A.K. Qin, P.N. Suganthan, G.-B. Huang, Evolutionary extreme learning machine, Pattern Recognition , 2008, 38(10):1759–1763.

[9] Fei Han, Hai-FenYao, Qing-HuaLing. An improved evolutionary extreme learning machine based on particle swarm optimization [J]. Neurocomputing , 2013, 116 : 87–93 .

[10] R. Ghosh, B. Verma.A hierarchical method for finding optimal architecture and weights using evolutionary least square based learning.Int. J. Neural Syst., 2003, 12(1):13-24.

[11] 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[J].IEEE Trans. Inf. Theory, 1998, 44(2):525-536.

[12] You Xu, Yang Shu.Evolutionary extreme learning machine-based on particle swarm optimization, International Symposium on Neural Networks 2006 (ISNN2006),2006, 3971:644–652.