

A New Extreme Learning Machine Optimized by Particle Swarm Optimization

Heng Zhang

College of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, P.R. China;

Corresponding author at: Kexue Road NO. 100, Zhengzhou 450001, P. R. China.

E-mail address: zhangheng@zzu.edu.cn

Keywords: parameters optimizing; particle swarm optimization; extreme learning machine; hidden neurons

Abstract. Extreme Learning Machine (ELM) is a new type of feedforward neural network (FNN). Compared with traditional single hidden layer FNN, ELM possesses higher speed and better performance. Due to the random determination of the input weights and hidden layer biases, ELM may need more hidden neurons to achieve a reasonable accuracy. In this paper, a novel ELM learning algorithm optimized by Particle Swarm Optimization (PSO) which called PSO-ELM is proposed. PSO-ELM uses PSO algorithm to select input weights and hidden layer biases. The experimental results based on SinC benchmarking function and a large number of classification experiments from UCI standard data set show that the new algorithm can produce better generalization performance with less hidden neurons than traditional ELM algorithms and conventional feedforward neural networks.

Introduction

In the past decades, neural networks based on gradient descent algorithms such as back propagation (BP) neural network have been widely applied in modeling, prediction, classification, and regression applications [1-4].

ELM has got an extensive concern of many scholars, because of faster learning speed, better performance and simpler parameters adjusting compared with conventional FNNs [6,7]. However ELM algorithm has a main drawback, the random chosen input weights and hidden layer biases of the SLFN with ELM are unlikely to be optimal. This has raised the question of how to create optimal input weights, hidden neurons and bias for a given task.

Particle swarm optimization (PSO) is a population-based optimization technique that emulates the social behavior of animals, such as the swarming of insects, the flocking of birds, and the schooling of fish, when searching for food in a collaborative manner. This technique was originally designed and introduced by Eberhart and Kennedy [7,8]. PSO has been widespread applied in applications such as optimal control and design [9], biomedical [10], clustering and classification [11].

To improve the generalization performance of ELM and to optimize its input weights and hidden layer bias, this paper introduces an optimized PSO-ELM that utilized the PSO algorithm to optimized the input weights and hidden layer bias of SLFN. The remainder of this article is organized as follows: Section 2 provides a brief overview of ELM. Section 3 presents a short review of the PSO algorithm. Section 4 presents the new PSO-ELM algorithm of this paper. Section 5 discusses the experimental results. Finally, Section 6 presents a brief conclusion.

Extreme learning machine

For N arbitrary distinct samples (x_i, y_i) , where $x_i = [x_{i1}, x_{i2}, \dots, x_{im}]^T \in R^n$ and

$y_i = [y_{i1}, y_{i2}, \dots, y_{im}]^T \in R^m$, A typical SLFN with L hidden neurons and activation function $f(x)$ can

be expressed as

$$\sum_{i=1}^L \beta_i f_i(x_j) = \sum_{i=1}^L \beta_i f_i(w_i \cdot x_j + b_i) = o_j, j = 1, 2, \dots, N \quad (1)$$

where $w_i = [w_{i1}, w_{i2}, \dots, w_{im}]^T$ represents the input weight matrices which connect i th hidden neuron and the input neuron, $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im}]^T$ represents the output connection matrices which connect i th hidden neuron and the output neuron, $o_i = [o_{i1}, o_{i2}, \dots, o_{im}]^T$ is the actual output of SLFN and b_i is the threshold of the i th hidden node. The output activation function in this paper is linear function.

A SLFN with L internal neurons can approximate these N samples with zero error, that is $\sum_{j=1}^L \|o_j - y_j\| = 0$, β_i , w_i and b_i should satisfy that

$$\sum_{i=1}^L \beta_i f(w_i \cdot x_j + b_i) = y_j, j = 1, 2, \dots, N \quad (2)$$

Equation (2) can be express compactly as

$$G\beta = Y \quad (3)$$

G is defined as the hidden layer output matrix of the SLFN. The output weights β can be obtained through simple smallest norm least squares solution as

$$\beta = \hat{G}Y \quad (4)$$

where \hat{G} is the Moore–Penrose generalized inverse of matrix G .

Particle swarm optimization

In a d -dimensional space, an n particles population can be expressed as $x = (x_1, x_2, \dots, x_n)$, $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ represents the position of the i^{th} particle. In this variation, the velocity and the position of each particle in a d -dimensional space can be modified with the following equations

$$v_{id}(t+1) = w \cdot v_{id}(t) + c_1 \cdot rand()(p_best_{it} - x_{id}) + c_2 \cdot rand()(g_best_d - x_{id}) \quad (5)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (6)$$

where $rand()$ is a random functions in the range $[0,1]$, p_best_{it} denotes the personal best of the it particle; g_best_d denotes global best of the d particle; c_1 and c_2 are positive constants; w is the inertia weight, and $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$ corresponds to the velocity for particle i . In order to prevent the aimless search of particles.

In [12], Shi and Eberhar introduced a new parameter into the original PSO optimizer, which can improve the performance of the particle swarm optimizer through an adaptive w adjustment strategy. In this paper, we use this new adaptive adjustment strategy to obtain w .

$$w = w_{min} + (w_{max} - w_{min}) \cdot e^{-3 \cdot (\frac{i}{n})^2} \quad (7)$$

Extreme learning machine optimized by particle swarm optimization

In order to optimize the input weight matrix and hidden layer bias, we propose a novel ELM learning algorithm optimized by PSO which called PSO-ELM in this paper. The procedure PSO-ELM algorithm can be summarized as follows:

1. Generate the initial population. Usually initial population size S is 20-40. The length of particle is $D=L \times (N+1)$, where L is the number of hidden layer neurons and N is the number of input vector dimensions.

2. For each particle θ^m , Use Eq (1)-(3) and training data compute the output weight β^m and training RMSE^m. Let the best fitness $RMSE^{best} = RMSE^1$ and the best personal particle is θ^1 . Then perform the following pseudo code:

2.1. For ($1 < i \leq S$)

```
{
    Calculate the fitness  $RMSE^i$  of  $i$ th particle  $\theta^i$ ;
    If  $RMSE^i > RMSE^{best}$ 
        Then  $pbest(i) = \theta^i$ 
         $RMSE^{best} = RMSE^i$ 
    Else  $pbest(i) = \theta^{i-1}$ 
```

```
}
```

Calculate the global best RMSE of the i th particle gbest;

2.2. For ($1 < i \leq itera$), where itera is the maximum iterations

```
{
    Update the particle velocity and position through eq (5) and (6);
    If  $i > itera$  or meet the best RMSE
        Break
    Else go to step 2.1
}
```

3. Perform the above pseudo code and then get the best particle which is the best input weight matrix and the hidden layer bias, then calculate the best output weight β^{best} through eq (1)-(3).

Performance evaluation

5.1 Approximation of ‘SinC’ function with noise

The ‘SinC’ function is a popular choice to illustrate SVM for regression in the literature.

$$y(x) = \begin{cases} \frac{\sin(x)}{x}, & x \neq 0 \\ 1, & x = 0 \end{cases}$$

We create 5000 training set (x_i, y_i) and 5000 testing set (x_i, y_i) respectively, where x_i are uniformly randomly distributed on the interval $(-10, 10)$.

10 hidden neurons are assigned in the PSO-ELM and BP algorithm and $itera = 1, 2, \dots, 7$. The average testing of 20 trails results with different max iterations are shown in Table 1. The testing performance with different hidden neurons of ELM and PSO-ELM are shown in Table 2.

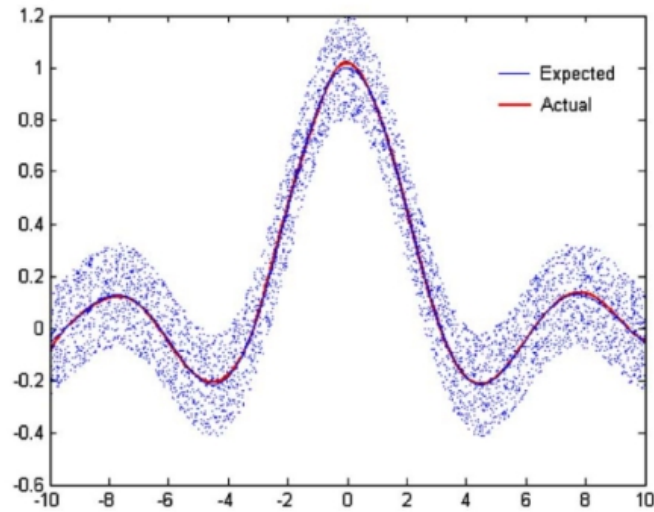
Table.1 The impact of the iteration number to PSO-ELM

itera	Training time (s)	Training RMSE	Testing RMSE
1	2.5156	0.1192	0.0331
2	3.3125	0.1165	0.0224
3	4.5000	0.1164	0.0182
4	5.5938	0.1151	0.0173
5	6.8281	0.1149	0.0147
6	8.0469	0.1142	0.0142
7	9.1875	0.1130	0.0140

Table.2 Testing performance with different hidden neurons of ELM and PSO-ELM

Hidden neurons	Training time (s)		Training RMSE		Testing RMSE	
	ELM	PSO-ELM	ELM	PSO-ELM	ELM	PSO-ELM
5	0.0301	6.8281	0.2791	0.1149	0.2532	0.0147
10	0.0313	6.6563	0.1199	0.1158	0.0391	0.0196
15	0.0425	10.2656	0.1198	0.1137	0.0388	0.0064
20	0.0669	14.7031	0.1139	0.1137	0.0088	0.0062
50	0.2188	45.0469	0.1138	0.1136	0.0084	0.0062
100	0.5625	140.9063	0.1138	0.1136	0.0082	0.0062
150	0.9688	256.1719	0.1138	0.1136	0.0082	0.0062
1000	39.6094	12551.12	0.1138	0.1135	0.0081	0.0060

According to Table 1, the training time increases as the iteration increasing. Training RMSE and testing RMSE decrease as the iteration increasing. The best performance of PSO-ELM can be gotten when the hidden neurons are 20. According to Table 1 and 2, the max iteration of PSO-ELM is set to 5 and the hidden neurons is set to 20. The regression performance of ‘SinC’ function is shown as Figure 1.

**Figure.1** The regression performance of ‘SinC’ function

To validate the performance of the proposed PSO-ELM, several methods, such as SVM, ELM, and BP are implemented for the comparison. The comparison of PSO-ELM and other methods is presented in Table 3.

Table.3 Comparison of PSO-ELM and other algorithms

Algorithm	Training time(s)	Training RMSE	Testing RMSE
PSO-ELM	14.7031	0.1137	0.0062
ELM	0.5625	0.1138	0.0082
SVM	9.7969	0.1130	0.0064
BP	36	0.0001	0.0091

According to Table 3, the performance of PSO-ELM and SVM is better than ELM and BP. PSO-ELM

5.2 Benchmarking with real classification applications

Without loss of generality, we selected part of standard classification data from standard UCI data. The data are tested by ELM and PSO-ELM respectively. 20 trials have been conducted for ELM and PSO-ELM algorithms and the average testing results are shown in Table 4.

Table 4. Result of classification problems

Data samples	Algorithms	Training time	Testing RMSE	Hidden neurons
Diabetes	ELM	0.0116s	0.7865	20
	PSO-ELM	14.105s	0.8005	10
LineBlobs	ELM	0.0313s	0.9798	100
	PSO-ELM	4.3594s	0.9907	5
Threecircles	ELM	0.0156s	0.9595	50
	PSO-ELM	4.3750s	0.9899	10
Twenty	ELM	0.0313s	0.9717	30
	PSO-ELM	8.6094s	0.9875	5

According to Table 4, ELM can achieve an high precision testing performance with less training time. However ELM need more hidden neurons to perform better. PSO-ELM can achieve a better performance than ELM with less hidden neurons (less than 20).

Conclusion

We proposed a novel PSO method to optimize the input weights matrix and the hidden layer bias of SLFN to achieve a better algorithm which called PSO-ELM. The PSO can be used to calculate the best input weights matrix and hidden layer bias. PSO-ELM combine the advantages of PSO and ELM. The testing RMSE performance of the proposed PSO-ELM approach is compared with some other methods. The comparisons confirm the superiority of our PSO-ELM approach in the ‘SinC’ function approximation and real classification applications dataset. PSO-ELM can achieve a better performance than ELM with less hidden neurons.

References

- [1] Pradhan B, Lee S. Regional landslide susceptibility analysis using back-propagation neural network model at Cameron Highland, Malaysia[J]. Landslides, 2010, 7(1): 13-30.
- [2] Liu M, Wang M, Wang J, et al. Comparison of random forest, support vector machine and back propagation neural network for electronic tongue data classification: Application to the recognition of orange beverage and Chinese vinegar[J]. Sensors and Actuators B: Chemical, 2013, 177: 970-980.
- [3] Wang J Z, Wang J J, Zhang Z G, et al. Forecasting stock indices with back propagation neural network[J]. Expert Systems with Applications, 2011, 38(11): 14346-14355.
- [4] Pradhan B, Lee S. Landslide susceptibility assessment and factor effect analysis: backpropagation artificial neural networks and their comparison with frequency ratio and bivariate logistic regression modelling [J]. Environmental Modelling & Software, 2010, 25(6): 747-759.
- [7] Deng W, Chen L. Color image water marking using regularized extreme learning Machine [J]. Neural Network World , 2010, 20(3):317–330.
- [8] Zong W.W, Huang G.B. Face recognition based on extreme learning machine[J]. Neurocomputing, 2011,74:2541-2551.

- [9] Eberhart R.C, Kennedy J, A new optimizer using particle swarm theory, in: Proceedings of the sixth International Symposium on Micro Machine and Human Science, 1995, 39–43.
- [10] Kennedy J, Eberhart R.C, Swarm Intelligence, Morgan Kaufman, 2001
- [11] M. Benedetti, R. Azaro, D. Franceschini, A. Massa, PSO-based real-time control of planar uniform circular arrays, *Antennas Wireless Propag. Lett.*, IEEE 5(2006) 545–548.
- [12] W. Cedefeto, D. Agrafiotis, Particle swarms for drug design, in: IEEE Congress on Evolutionary Computation, 2005, pp. 1218–1225.