



Local Coupled Extreme Learning Machine Based on Particle Swarm Optimization

Hongli Guo¹, Bin Li^{1,*}, Wei Li¹, Fengjuan Qiao¹, Xuewen Rong² and Yibin Li²

- ¹ School of Mathematics and Statistics, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China; guohongli0127@163.com (H.G.); 15106900205@163.com (W.L.); qiao_fj@126.com (F.Q.)
- ² School of Control Science and Engineering, Shandong University, Jinan 250061, China; scrobot@163.com (X.R.); liyb@sdu.edu.cn (Y.L.)
- * Correspondence: ribbenlee@126.com

Received: 20 August 2018; Accepted: 29 October 2018; Published: 1 November 2018



Abstract: We developed a new method of intelligent optimum strategy for a local coupled extreme learning machine (LC-ELM). In this method, both the weights and biases between the input layer and the hidden layer, as well as the addresses and radiuses in the local coupled parameters, are determined and optimized based on the particle swarm optimization (PSO) algorithm. Compared with extreme learning machine (ELM), LC-ELM and extreme learning machine based on particle optimization (PSO-ELM) that have the same network size or compact network configuration, simulation results in terms of regression and classification benchmark problems show that the proposed algorithm, which is called LC-PSO-ELM, has improved generalization performance and robustness.

Keywords: extreme learning machine; LC-ELM; particle swarm optimization; LC-PSO-ELM

1. Introduction

The mathematical model of single-hidden layer feed-forward neural networks (SLFNs) has been widely used in many domains because of its ability to approximate strongly nonlinear input-output mappings. However, traditional learning methods are usually much slower than required while few faster learning algorithms for SLFNs are generated [1]. In 2006, a novel learning algorithm for SLFNs called extreme learning machine (ELM) [1,2] was presented by Huang et al. for decreasing the training time of SLFNs.

Different from the existing learning algorithms of SLFNs, the weights and biases between the input layer and hidden layer of the ELM were chosen randomly, then the weights between the hidden layer and the output layer were determined based on the ordinary least squares. The ELM learning algorithm has fast learning speed and good generalization performance with little human intervention, which makes the algorithm applicable to many areas, such as stock prediction [3], image classification [4], fault diagnosis [5], etc.

In the ELM, the number of hidden neurons is required to be greater than or equal to the number of the training samples so as to guarantee the convergence of the algorithm. Therefore, there will be quite a lot of input-hidden weights when the number of input neurons is large [6], which may reduce the generalization performance of SLFNs. The original ELM model has been equipped with various extensions to make it more suitable and efficient for specific applications [7]. For example, based on the structure of the local coupled feed-forward neural network (LCFNN) [8,9] and the learning mechanism of the ELM algorithm, the local coupled extreme learning machine (LC-ELM) learning algorithm was proposed by Qu in 2014 [10]. The algorithm could decrease the researching complexity of the weights between the input layer and the hidden layer by means of assigning the addresses to



the hidden neurons [10]. The advantage of the LC-ELM on image watermarking was examined by Mehta et al. [11].

In the LC-ELM learning algorithm, the addresses and radiuses were generally preset empirically or randomly. And, thus, those parameters might not be optimal for the LC-ELM, and the algorithm may yield an inappropriate underlying model. In 2015, Qu et al. presented an evolutionary local coupled extreme learning machine (ELC-ELM). In the ELC-ELM, the differential evolutionary (DE) algorithm was used to optimize the addresses and the radiuses of the fuzzy membership functions in hidden neurons for improving the generalization performance [12]. However, it should be noted that the hidden biases and input weights in the ELC-ELM were also set randomly.

The DE algorithm has good global converge property by means of utilizing the differential information of the population. However, the instability performance of DE can also be caused because of the above reason and the algorithm may be trapped in local optima [13,14]. Moreover, three parameters of the DE algorithm should be controlled manually [15]. In 1995, the particle swarm optimization (PSO) algorithm was presented by Eberhart et al. [16] and has been used in many optimization fields as it can converge to the global minima quickly. Compared with other stochastic optimization techniques, the advantages of the PSO algorithm are that it is easy to be implemented in practice and few parameters need to be adjusted [17,18]. The PSO algorithm and its improved variants, such as APSO (Adaptive PSO) and PSOGSA (The hybrid PSO and gravitational search algorithm), were used to select the optimal parameters between the input layer and the hidden layer (input weights and biases) of the ELM [19,20].

Therefore, in order to overcome the limitation of the DE, a new method combining the LC-ELM with an improved PSO called LC-PSO-ELM is proposed in this paper. In the proposed algorithm, the improved PSO algorithm is used to optimize the address and window radius of the local coupled parameters. In addition, the input weights and hidden layer biases of the ELM are also optimized to further improve the generalization performance of the LC-ELM, and the MP generalized inverse is used to calculate the weights between the hidden layer and the output layer analytically. In order to prove the superiority of the proposed algorithm, we compared the computer simulation results from our developed algorithm to those from the ELM, LC-ELM and PSO-ELM algorithms, respectively. The comparison results demonstrated that the newly developed algorithm exhibits improved generalization performance with the highest accuracy.

The rest of this paper is organized as follows. The local coupled extreme learning machine (LC-ELM) and the improved particle swarm optimization algorithm are given in Section 2. The local coupled extreme learning machine based on the PSO algorithm is introduced in Section 3. Section 4 includes different simulation results and analysis of the proposed algorithm in regression and classification benchmark problems. Finally, the conclusions are summarized in Section 5.

2. Theoretical Background

2.1. Local Coupled Extreme Learning Machine

The ELM learning algorithm is a simple, fast and efficient method. For further improving the generalization performance of the ELM, the LC-ELM learning algorithm was proposed by Qu [10] in which the efficiency of LC-ELM in terms of classification and regression benchmark problems was investigated.

In the LC-ELM, due to the utilization of the fuzzy membership function $F(\cdot)$ and the similarity relation $S(x, d_i)$, the complexity of the weight searching space was reduced and the generalization performance was correspondingly improved in terms of the simple neural networks structure. The mathematical formulation of the LC-ELM is presented as follows:

For *M* arbitrary distinct examples $(\mathbf{x}_i, \mathbf{t}_i)$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \cdots, x_{ip}] \in \mathbb{R}^p$ is the input and $\mathbf{t}_i = [t_{i1}, t_{i1}, \cdots, t_{iq}] \in \mathbb{R}^q$ is the expected output, $i = 1, \dots, M$. The output of the hidden layer neurons $g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i)$ for the ELM is modified with the help of fuzzy membership function as

 $g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i)F(S(\mathbf{x}_j, \mathbf{d}_i))$. Therefore, the network output of the LC-ELM with *N* hidden neurons are mathematically modeled by

$$f(\mathbf{x}_j) = \sum_{i=1}^N \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) F(S(\mathbf{x}_j, \mathbf{d}_i)), i = 1, \dots, N, j = 1, \dots, M$$
(1)

where $g(\cdot)$ denotes the activation function of the ELM, which can not only be sigmoid functions, however also other functions such as sin, cos, cubic, etc. β_i denotes the weight vector connecting the *i*th hidden neuron and the output neurons, \mathbf{w}_i is the weight vector connecting the *i*th hidden neuron and the input neurons. b_i is the bias of the *i*th hidden neuron. \mathbf{d}_i is the address of the *i*th hidden node.

In the LC-ELM learning algorithm, the similarity relation $S(\mathbf{x}, \mathbf{d}_i)$ is the distance between the input \mathbf{x} and the *i*th hidden node with address \mathbf{d}_i . Various forms of fuzzy membership functions $F(\cdot)$, such as Gaussian function, sigmoid function and reverse sigmoid function [21,22], are utilized. In addition, the underlying radius parameter r is kept in $F(\cdot)$ for adjusting the width of the activation area, which is also an optimized parameter, to the same as the address parameter \mathbf{d} . Combining the structure of the LCFNN with the learning mechanism of the ELM, the LC-ELM also is a three step learning algorithm and the parameters (input weights \mathbf{w} and biases b between the input layer and hidden layer, the address \mathbf{d} of the hidden neurons) of the networks are assigned randomly, which is the same as the ELM [10].

The standard LC-ELM learning algorithm can approximate these *M* examples with zero error means $\sum_{j=1}^{N} ||o_j - t_j|| = 0$, where \mathbf{o}_j is the actual output of the LC-ELM. i.e., the corresponding relation is defined by

$$f(\mathbf{x}_i) = \mathbf{t}_i, j = 1, \dots, M \tag{2}$$

The above *M* equations can be written compactly as a linear system:

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{T}, j = 1, \dots M \tag{3}$$

where H is the output matrix of the hidden layer and can be expressed as

$$\mathbf{H} = \{\mathbf{h}_{ji} = g(\mathbf{w}_i \mathbf{x}_j + b_i) F(S(\mathbf{x}, \mathbf{d}_i))\} \ i = 1, \dots, N, j = 1, \dots, M.$$
(4)

in the above Equation (4), $\mathbf{h}_{ji} = g(\mathbf{w}_i \mathbf{x}_j + b_i)$ denotes the output of the *i*th hidden neuron with respect to \mathbf{x}_j . $\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_N]_{N \times q}^T$ is the matrix of the output weights and $\boldsymbol{\beta}_i$ denotes the weight vector connecting the *i*th hidden node and the output layer. $\mathbf{T} = [\mathbf{t}_1, \dots, \mathbf{t}_M]_{M \times q}^T$ is the matrix of the target of the LC-ELM.

The smallest norm least squares solution of Equation (3) is

$$\widehat{\boldsymbol{\beta}} = \mathbf{H}^{+}\mathbf{T}$$
(5)

where **H**⁺ is the Moore-Penrose generalized inverse of the hidden layer output matrix **H** [23]. Based on the above discussion, the LC-ELM algorithm can be summarized in Algorithm 1.

Algorithm 1. The algorithm flow of LC-LEM

- (1) Input weights **w**, hidden bias *b* and the node address **d** are allocated randomly.
- (2) The output matrix of the hidden layer H is computed using Equation (4).
- (3) Calculate the output weights β between the hidden layer and the output layer based on Equation (5):

$$\beta = H^+T.$$

2.2. Particle Swarm Optimization

In 1995, a particle swarm methodology was proposed for nonlinear function optimization by Kennedy and Eberhart [16], which was called the PSO algorithm. It belongs to a population-based, heuristic optimization algorithm. The PSO algorithm is simple, easy to be realized and has a fast convergence rate. It has been widely applied in the fields of scientific research and engineering application [20].

As a swarm-based algorithm, the particles of the PSO algorithm may flow through the searching space depending on the best position information of their own and their neighbors'. The initial values of the particles in the population are set randomly [24].

In the PSO algorithm, suppose *D* is the dimension of searching space and *N* is the number of particles, respectively. Then, x_i^t and v_i^k are denoted by the current position and the current velocity of the *i*th particle at iteration *t*, respectively [25]. Therefore, the new velocity and the particle position in the next iterative time are described as:

$$v_i^k(t+1) = w \cdot v_i^k(t) + c_1 \cdot rand()(p_i^k(t) - x_i^k(t)) + c_2 \cdot rand()(g_i^k(t) - x_i^k(t))$$
(6)

$$x_i^k(t+1) = x_i^k(t) + v_i^k(t+1) \ 1 \le i \le \hat{N}, 1 \le k \le D$$
(7)

where *w* denotes the inertia weight. c_1 and c_2 stand for the different acceleration coefficient, respectively. *rand()* denotes a constant value in the interval [0, 1] and is set randomly. p_i^k is the best position of the *i*th particle in the search stage at present, g_i^k represents the global best position, which constitutes the best position found in the population at present.

In the PSO algorithm, the initial parameter w plays the role of balancing the global search and the local search. Therefore, in order to ensure higher exploring ability in the early iteration and fast convergence speed in the last part iteration, w is not a constant and can be expressed as a nonlinear function of time [17,26]:

$$w(t) = w_{max} - iter \times \frac{(w_{max} - w_{min})}{max_iter}$$
(8)

where w_{max} and w_{min} are the initial and terminal values of inertia weight in the iteration process, respectively. The parameter max_iter is the maximum iteration number of the algorithm and *iter* is the current iteration time of the algorithm.

In addition, in order to enhance the global search in the early part iteration, to encourage the particles to converge to the global optimal solution and to improve the convergence speed in the final iteration period [27], the acceleration parameters c_1 and c_2 are described as:

$$c_1 = (c_{1min} - c_{1max})\frac{iter}{max_iter} + c_{1max}$$
⁽⁹⁾

$$c_2 = (c_{2max} - c_{2min})\frac{iter}{max_iter} + c_{2min}$$
(10)

where c_{1max} and c_{1min} , c_{2max} and c_{2min} are constants. Based on the Equation (6), the searching ability of the cognitive and social components can be changed by changing the values of c_1 and c_2 , which can improve the convergence rate of the PSO algorithm.

3. Local Coupled Extreme Learning Machine Based on the PSO Algorithm

Based on the optimization technique of the above PSO algorithm with self-adaptive parameters w and c, the parameter values w, b, d and r of the LC-ELM are optimized for improving the generalization performance in this work.

In the LC-ELM learning algorithm, the decoupling of the input layer and the hidden layer is determined by the address parameter \mathbf{d} and the radius parameter r. However, these parameter values are randomly determined. In other words, they might not be suitable for the algorithm, resulting

in the poor performance of the algorithm. In addition, the hidden biases b and input weights **w** are also set randomly in the LC-ELM. Therefore, for improving the performance of the LC-ELM algorithm, the four parameters (**w**, b, **d**, r) of the LC-ELM are optimized based on the above adaptive PSO algorithm simultaneously. When the optimal parameters of the LC-ELM algorithm are established, the t weights between the hidden layer and the output layer of the LC-ELM are determined analytically based on the Equation (5) of the ELM, which is called the LC-PSO-ELM algorithm in this paper.

Therefore, the particles in the searching space of the LC-PSO-ELM are composed of a set by the parameter values of input weights, hidden biases, address and radius, which can be defined as:

$$\boldsymbol{\theta} \in [w_{11}, w_{12}, \dots, w_{1N}, w_{21}, w_{22}, \dots, w_{2N}, \dots, w_{p1}, w_{p2}, \dots, w_{pN}, \dots, b_1, b_2, \dots, b_N d_{11}, d_{12}, \dots, d_{1N}, d_{21}, d_{22}, \dots, d_{p1}, d_{p2}, \dots, d_{pN}, \dots, r_1, r_2, \dots, r_N]$$
(11)

where $\mathbf{w} = {\{\mathbf{w}_i | \mathbf{w}_i \in R^p, i = 1, ..., N\}}, b \in R^N, \mathbf{d} = {\{\mathbf{d}_i | \mathbf{d}_i \in R^p, i = 1, ..., N\}} and r \in R^N.$

Based on the global searching capability of the above PSO algorithm and the universal approximation performance of the LC-ELM learning algorithm, the detailed steps of the LC-PSO-ELM algorithm (Algorithm 2) are described as follows:

The parameters in the algorithm are defined as: the training set is denoted as $\{(x_i, t_i)x_i \in \mathbb{R}^p, t_i \in \mathbb{R}^q, i = 1, ..., M\}$, g(x) is the output function $g(\mathbf{w}_i \mathbf{x}_j + b_i)$ of the hidden neuron, N is the number of the hidden neurons, F and S are fuzzy membership and similarity function, respectively. *max_iter* is the preset maximum learning epoch of the PSO algorithm. w_{max} and w_{min} are the initial and terminal values of inertia weight in the iterative stage. c_{max} and c_{min} are the initial and final values of the acceleration constants.

Algorithm 2. The algorithm flow of LC-PSO-ELM

Each particle in the generation is composed of a set of the input weights \mathbf{w} , biases b, address \mathbf{d} and radius r, as is shown in Equation (11). The initialization value of all of the components of the particle are set from -1 to 1 randomly.

- (2) Iter = 1
- (3) While $Iter \leq max_{iter}$

(4) (1) Evaluate the fitness function of each particle (the root means standard error for regression problems and the classification accuracy for classification problems).

(2) Modify the position of the particle according to Equations (4)–(8).

- (3) Iter = Iter + 1
- (5) end while
- (6) The optimal parameters of the LC-ELM can be determined. Then, based on the optimized parameters:
- (1) The output matrix **H** of the hidden layer is computed based on Equation (4).
- (2) The weight β is calculated based on Equation (5).

Similar to the LC-ELM, the combinational function $F(S(\mathbf{x}, \mathbf{d}_i))$ between the similarity relation S(x) and the fuzzy membership F(x) in the LC-PSO-ELM also has many selection strategies. For example, the similarity relation function could be selected by the fuzzy similarity function, Gaussian kernel and wave kernel functions, etc. Meanwhile, the fuzzy membership Equations (12)–(14) can be also chosen in the LC-PSO-ELM learning algorithm.

$$F(x) = \exp(-\frac{x^2}{r}) \tag{12}$$

$$F(x) = \frac{2}{1 + \exp(x/r)}$$
 (13)

⁽¹⁾ Initiate the population (particle).

$$F(x) = \tanh(-\frac{x}{r}) + 1. \tag{14}$$

4. Simulations and Performance Verification

In this section, the proposed LC-PSO-ELM learning algorithm and three alternative ELM algorithms in the aspect of four function approximation (regression) and four classification benchmark problems, the original ELM, LC-ELM [10] PSO-ELM [17], are conducted in the MATLAB R16a environment running with 3.4 GHz CPU and 16 G RAM. The parameters specification of the benchmarks problems is shown in Table 1. The experimentally well-characterized datasets were chosen for good comparison in this paper [28,29], in which the Box and Jenkins gas furnace data were sourced from the reference [30], the Calhousing data came from the StatLib dataset [31] and the other dataset was derived from the UCI (University of California, Irvine, CA, USA) Machine Learning Repository [32], respectively. For each dataset, the input sequence of the data was changed randomly and then the data were divided into two groups of training data and testing data for experiments based approximately on a 70–30 ratio. The number of the two groups is shown in Table 1.

Problem	Dataset	Attributes	Class	Training Data	Testing Data
	Box and Jenkins gas furnace data	10	1	203	87
Regression	Autompg	7	1	279	119
	Abalone	8	1	2923	1254
	Calhousing	8	1	14,448	6192
	Wine	13	3	124	54
	Iris	4	3	95	42
Classification	Diabetes	8	2	537	231
	Satimage	36	6	4504	1931

Table 1. Parameters specification of the benchmark problems.

The number of the population of the PSO algorithm is 200 and the maximum iterative number is 50. The configurations of the ELM, PSO-ELM, LC-ELM and the LC-PSO-ELM are listed in Table 2. For simplicity, RN is the abbreviation for random number and NDRN is the abbreviation for normally distributed random numbers.

As shown in Table 2, the sigmoid function is selected as the activation function of the four learning algorithms. The wave kernel $S(x, y) = (\theta/||x - y||) \sin(||x - y||/\theta)$ is selected as the similarity function and the reversed sigmoid function Equation (13) $F(x) = \frac{2}{1 + \exp(x/r)}$ is selected as the fuzzy membership function in the LC-ELM and the LC-PSO-ELM algorithms, respectively.

Table 2. Configurations of the ELM (extreme learning machine), PSO-ELM (extreme learning machine based on particle optimization), LC-ELM (local coupled extreme learning machine) and LC-PSO-ELM algorithms.

Configurations	ELM	PSO-ELM	LC-ELM	LC-PSO-ELM
Input weight and hidden layer biases	RN in [-1, 1]	RN in [-1, 1]	RN in [-1, 1]	NDRN (normally distributed random numbers)
Activation function	sigmoid	sigmoid	sigmoid	sigmoid
Hidden node address and window radius	_	_	RN in [0,1] & 0.4	NDRN
Similarity	-	-	Wave kernel	Wave kernel
Fuzzy membership function	-	_	Equation (13)	Equation (13)

In order to increase the persuasion of different algorithms in terms of validity, 10 trials of the average simulation results (root mean square error (RMSE) is the abbreviation) for regression benchmarks and classification accuracy for classification (pattern classification) are given in the following tables. The training and testing subsets of each experiment in the 10 trials are created by randomly choosing samples of the datasets based on a 70–30 ratio renewedly, the robustness of the algorithms is compared using the standard deviation (STD is the abbreviation) of the 10 trials. The CPU time of training is used to evaluate the computational complexity of the algorithms. The testing error and the CPU time of testing are used to evaluate the generalization performance and application value of the algorithms, respectively. On the other hand, in all of the tables of the simulation results, symbols in bold represent the comparatively best value of the corresponding algorithms. The control parameters of the PSO that were used in different algorithms of PSO-ELM and LC-PSO-ELM are listed in Table 3.

Table 3. Control parameters used in the different algorithms of the PSO-ELM and LC-PSO-ELM.

Algorithm	<i>w</i> max	wmin	<i>C_{max}</i>	C _{min}	c_1	c_2
PSO-ELM/LC-PSO-ELM	0.9	0.4	2.5	0.5	2	2

Besides the parameters of input weights and hidden biases, address parameter and the radius parameter (\mathbf{w} , b, d, r), the generalization performance of the algorithms is affected mainly by the number of hidden nodes (neurons). In order to simply the analysis and comparison, all the figures in this paper illustrating the generalization curves of different algorithms based on different hidden neurons in function approximation and classification problems are the simulation results in one run of the experiments. As shown in Figure 1, in the function approximation problems, with the increasing of the hidden nodes from one to some determined value, the testing RMSE of the algorithms first rapidly decreases, then the curves become stable with a fluctuating value, except for the LC-ELM learning algorithm. From the figures, we can also conclude that the proposed LC-PSO-ELM algorithm has less testing RMSE error in most cases, which means that the proposed algorithm in terms of generalization performance is better than the other algorithms in one run.



Figure 1. Cont.



Figure 1. The generalization curves of different algorithms based on different hidden neurons on function approximation problem: (**a**) Box and Jenkins gas furnace; (**b**) Autompg; (**c**) Abalone; (**d**) Calhousing.

Figure 2 shows the generalization curves of classification problems in one run of the experiments. The testing classification accuracy is gradually bigger with the increasing of the hidden neurons, which also show the superiority of the proposed algorithm and the instability of the LC-ELM algorithm in one run.

For the sake of comparison, based on the generalization curves of different algorithms in terms of different hidden neurons on function approximation and classification problems, the selection of hidden neurons for the proposed algorithm is equal or less than the other algorithms. Meanwhile, a good number of hidden neurons of different algorithms in terms of generalization performance are also considered in the selection process of hidden neurons. Finally, the number of hidden neurons in the algorithms for different benchmark problems is shown in Table 4.



Figure 2. Cont.



Figure 2. The generalization curves of different algorithms based on different hidden neurons on classification problems: (a) Wine; (b) Iris; (c) Diabetes; (d) Satimage.

Dataset Algorithms	Box and Jenkins Gas Furnace Data	Autompg	Abalone	Calhousing	Wine	Iris	Diabetes	Satimage
ELM	15	72	25	10	15	10	38	30
LC_ELM	15	57	25	10	15	10	26	30
PSO-ELM	15	40	25	10	15	10	38	30
LC-PSO-ELN	A 15	27	25	10	15	10	17	30

Table 4. The number of hidden neurons in the algorithms for different benchmark problems.

4.1. Performance Comparison of Regression Benchmark Problems

This section mainly shows the comparison results of the original ELM, LC-ELM, PSO-ELM and LC-PSO-ELM four algorithms on the function approximation datasets. The average simulation results of 10 experiments are shown in Tables 5 and 6. From these tables, we may see that the training time of the proposed algorithm consumed much more than the other ones, which means that the adaptive PSO algorithm needs more time for searching the global optimal solution of the parameters (\mathbf{w} , b, \mathbf{d} , r) in the LC-PSO-ELM algorithm.

Although the training error is higher than the other algorithms in the proposed algorithm in terms of the Autompg problem, the proposed algorithm in this paper focuses on superiority in terms of improved generalization performance, the fact that the testing time of all of them is almost equivalent and the proposed algorithm has better generalization performance with fewer parameters and compact network configuration, which shows that the proposed algorithm has good generalization value and real applicability.

Moreover, the proposed LC-PSO-ELM and PSO-ELM learning algorithms have relatively less value of STD in the experiments, which means that the algorithms have stable performance with parameters optimized by means of the PSO algorithm, although searching the optimal parameters needs much time in the training process.

Except for the STD value of the Autompg, the other problems of LC-ELM are bigger than the ELM, PSO-ELM and LC-PSO-ELM algorithms. The results show that the LC-ELM is the most unstable learning algorithm out of the four, and they are also the same as the simulation results in Figures 1 and 2.

	Box a	nd Jenkins	Gas Furnace	Data	Autompg			
Algorithms	Training Time (s) STD	Testing Time (s) STD	Training Error STD	Testing Error STD	Training Time (s) STD	Testing Time (s) STD	Training Error STD	Testing Error STD
TIM	0	0	0.0187	0.0214	0.0125	0.0062	0.0533	0.0866
ELM	0	0	${6.9674 imes 10^{-4}}$	0.0015	0.0263	0.0197	0.0030	0.0108
LOFIN	0.0094	0.0035	0.0183	0.0262	0.0187	0.0156	0.0635	0.0885
LC-ELM	0.0211	0.0075	0.0019	0.0060	0.0301	0.0255	0.0031	0.0074
	116.2472	0.0156	0.0161	0.0184	221.9707	0	0.0601	0.0662
PSO-ELM	2.3068	0.0337	0.0011	$\begin{array}{c} 8.8292 \times \\ 10^{-4} \end{array}$	2.8030	0	0.0018	0.0027
	155.2323	0.0312	0.0178	0.0182	296.2370	0.0203	0.0663	0.0653
LC-F 3O-ELM	4.6814	0.0353	0.0017	0.0018	5.1307	0.0148	0.0027	0.0043

Table 5. Performance comparison of different algorithms on regression problems of Box and Jenkins gas furnace data and Autompg.

		Aba	lone			Calhousing				
Algorithms	Training time (s) STD	Testing Time (s) STD	Training Error STD	Testing Error STD	Training Time (s) STD	Testing Time(s) STD	Training Error STD	Testing Error STD		
ELM	0.0078	0.0094	0.0756	0.0771	0.0172	0.0187	0.1441	0.1453		
ELM	0.0198	0.0197	$9.3903 imes 10^{-4}$	0.0019	0.0289	0.0301	0.0030	0.0033		
IC FIM	0.0593	0.0125	0.0758	0.0849	0.0577	0.0421	0.1449	0.1589		
LC_ELM	0.0498	0.0218	7.2119×10^{-4}	0.0067	0.0255	0.0221	0.0022	0.0140		
DCO ELM	308.3157	0.0312	0.0747	0.0761	486.9368	0	0.1386	0.1395		
PSO-ELM	7.2990	0.0504	$9.4281 imes 10^{-4}$	0.0022	6.2410	0	0.0015	0.0023		
LC-PSO-ELM	1.8441×10^{3}	0.1576	0.0743	0.0748	8.2110×10^{3}	0.7653	0.1260	0.1317		
	55.4315	0.0307	0.0011	$9.9337 imes 10^{-4}$	70.2587	0.0689	0.0052	0.0023		

Table 6. Performance comparison of different algorithms on regression problems of Abalone and Calhousing.

4.2. Performance Comparison of Classification Problems

Performance comparison among ELM, LC-ELM, PSO-ELM and LC-PSO-ELM algorithms is given in Tables 7 and 8. The generalization performance of the problems is justified by testing classification accuracy (testing accuracy). The simulation results in the tables show that the LC-PSO-ELM algorithm is obviously superior to the other algorithms in terms of generalization performance, except for the Iris dataset. From the Tables 7 and 8, we can also conclude that the PSO-ELM algorithm and the LC-PSO-ELM algorithm have the comparable generalization performance in the Iris dataset. From the subgraph of Figure 1, there are 16 times to 100% in the testing classification accuracy of the proposed algorithm in 20 trials with the increasing of the number of hidden neurons and the PSO-ELM learning algorithm has 15 times to 100%, which also shows the same conclusion. Therefore, the preferable performance of the proposed algorithm illustrates that the selection of optimized parameters in these specific problems is suitable for improving the generalization performance of the model.

Table 7. Performance comparison of different algorithms on classification problems of Wine and Iris.

		W	ine		Iris			
Algorithms	Training	Testing	Training	Testing	Training	Testing	Training	Testing
	Time (s)	Time (s)	Accuracy	Accuracy	Time (s)	Time (s)	Accuracy	Accuracy
	STD	STD	STD	STD	STD	STD	STD	STD
ELM	$0.0047 \\ 0.0148$	0.0125 0.0263	0.9952 0.0068	0.9741 0.0096	0.0390 0.0287	0 0	0.9642 0.0173	0.9334 0.0219
LC-ELM	0.0156	0	0.9911	0.9537	0.0218	0.0062	0.9779	0.9500
	0.0255	0	0.0097	0.0180	0.0287	0.0197	0.0092	0.0176
PSO-ELM	33.7149	0	0.9895	0.9888	24.1443	0	0.9880	0.9842
	0.3132	0	0.0054	0.0122	0.7071	0	0.0078	0.0141
LC-PSO-ELM	139.0843	0.0281	0.9917	0.9889	126.8756	0.0234	0.9882	0.9860
	6.3945	0.0301	0.0099	0.0129	1.7935	0.0437	0.0060	0.0150

		Dial	oetes		Satimage			
Algorithms	Training Time (s) STD	Testing Time (s) STD	Training Accuracy STD	Testing Accuracy STD	Training Time (s) STD	Testing Time (s) STD	Training Accuracy STD	Testing Accuracy STD
FLM	0.0234	0.0125	0.8004	0.7835	0.0187	0.0047	0.8299	0.8248
ELM	0.0305	0.0263	0.0075	0.0153	0.0242	0.0148	0.0071	0.0064
LCEIM	0.0125	0.0062	0.7933	0.7779	0.0154	0.0234	0.8307	0.8238
LC-ELM	0.0230	0.0197	0.0079	0.0117	4.9453×10^{-4}	0.0247	0.0082	0.0081
DCO ELM	114.5703	0.0109	0.7937	0.7912	384.5331	0.0094	0.8513	0.8522
PSO-ELM	4.8491	0.0345	0.0118	0.0103	9.3345	0.0296	0.0035	0.0041
LC-PSO-ELM	299.1179	0.0250	0.7966	0.7918	3.9371×10^{3}	0.3401	0.8583	0.8685
	0.6804	0.0168	0.0171	0.0198	42.8024	0.0161	0.0032	0.0062

Table 8. Performance comparison of different algorithms on classification problems of Diabetes and Satimage.

Moreover, the STD value of the PSO-ELM learning algorithm is the least in the four algorithms, which shows that it is more easily obtained from the global solution in terms of searching two parameters than four parameters for the PSO algorithm. In addition, the LC-ELM is also the most unstable learning algorithm in most cases.

In summary, by analyzing all of the obtained results, the following conclusions can be drawn:

- (1) The generalization performance of the ELM algorithm can be improved by means of the parameter optimization based on the PSO.
- (2) The improvement of the generalization performance has been made at the expense of the consumption of the training time of CPU for searching the optimal parameters of the model.
- (3) The proposed algorithm in this paper has the best generalization ability for real applications.

4.3. Performance Comparison of LC-ELM Based on Two Different Optimization Methods of DE and PSO

Performance comparison results of the ELC-ELM [12] and the LC-PSO-ELM algorithms on regression or classification problems are listed in Table 9. Here, in the ELC-ELM algorithm, the differential evolution (DE) optimization algorithm is used for improving the generalization performance of the ELC-ELM (evolution local coupled extreme learning machine) algorithm, in which the parameters of the hidden neuron address and the radiuses of the fuzzy membership functions are optimized; otherwise, the input weights and hidden biases are still preset randomly in this algorithm.

		Auto	mpg		Iris			
Algorithms	Training Time (s)	Training Error STD	Testing Error STD	Number of Hidden Neurons	Training Time (s)	Training Accuracy STD	Testing Accuracy STD	Number of Hidden Neurons
ELC-ELM	37.1688	0.0805 0.0059	0.0769 0.0048	15	11.8374	97.40 0.52	97.20 2.70	15
LC-PSO-ELM	[191.3243	0.0681 0.0035	0.0696 0.0049	15	126.8756	0.9882 0.0060	0.9860 0.0150	10

Table 9. Performance comparison of the ELC-ELM and LC-PSO-ELM algorithms.

The function approximation problem of Autompg and the classification problem of the Iris data sets are used for comparing the generalization performance of the two algorithms. The number of hidden neurons in the LC-PSO-ELM algorithm is the same as or less than that in the ELC-ELM

algorithm. As can be seen from Table 9 (the data of simulation results in the ELC-ELM algorithm came from reference [12]), compared with the ELC-ELM algorithm, although the learning speed of the LC-PSO-ELM is slower than the ELC-ELM, the generalization performance of the LC-PSO-ELM algorithm for optimizing four parameter values is better than the ELC-ELM algorithm for optimizing two parameter values.

4.4. Performance Comparison of the LC-PSO-ELM Based on Different Fuzzy Membership Functions

The choice of activation (basis) functions of the ELM learning algorithm is problem dependent [33], which means that different fuzzy membership function in the LC-ELM and the LC-PSO-ELM algorithms will affect the generalization performance. Meanwhile, Yu pointed out that the window function that is used in the LC-ELM does not satisfy the necessary conditions of window function that are required by LCFNN. As a result, it is possible that the improper window function can cause the LC-ELM to have the same discriminant with the basic ELM [34]. For this reason, three different fuzzy membership functions of Gaussian function, reversed sigmoid function and reversed tanh function were used to verify the results. The simulation results of 10 trials with the three different fuzzy membership functions in the LC-PSO-ELM algorithms on regression and classification problems are listed in Table 10.

As can be seen from Table 10, the simulation results demonstrate that the LC-PSO-ELM learning algorithm has different generalization performance with different fuzzy membership functions, and the better test accuracy can be obtained in the LC-PSO-ELM algorithm using the reversed sigmoid function.

Fuzzy	Box	and Jenkins	Gas Furnac	e Data	Diabetes			
Membership Function	Training Time (s)	Training Error STD	Testing Error STD	Number of Hidden Neurons	Training Time (s)	Training Accuracy STD	Testing Accuracy STD	Number of Hidden Neurons
Gaussian function	163.3120 1.7935	0.0175 0.0010	$0.0186 \\ 3.4140 \times 10^{-4}$	15	325.6693 5.7551	0.7753 0.0151	0.7637 0.0293	17
reversed sigmoid function	155.2323 4.6814	0.0178 0.0017	0.0182 0.0018	15	299.1179 0.6804	0.7966 0.0171	0.7918 0.0198	17
reversed tanh function	162.3003 4.7447	0.0242 0.0024	0.0243 0.0035	15	332.4445 7.1548	0.7263 0.0498	0.7133 0.0668	17

Table 10. Performance comparison of different fuzzy membership function in the LC-PSO-ELM algorithms on regression or classification problems.

5. Conclusions

In this study, a novel learning algorithm, named LC-PSO-ELM, was proposed by means of the frame structure of LC-ELM and the parameter optimization strategy of the PSO algorithm. The parameters of input weights, hidden biases, addresses and radiuses were all adjusted by the PSO for searching the optimal solution in the model.

Based on the function approximation and classification benchmarks problems, the performance of the LC-PSO-ELM utilizing different fuzzy membership functions was conducted. Meanwhile, the generalization performance of the four algorithms of ELM, LC-ELM, PSO-ELM and LC-PSO-ELM were compared, which showed that the proposed algorithm can produce better generalization performance in most cases, compared with the other alternative ELM-based approaches.

Although the LC-PSO-ELM can obtain a significantly improved generalization performance, the training time of the algorithm was much longer than the others due to the fact that four parameter values should be optimized in the algorithm. In future, it is necessary to propose a parallel training mechanism for the proposed method for improving the efficiency to solve problems with very large datasets. Correspondingly, it is also necessary to exploit the sensitivities of these chosen activation functions in theory in the future.

Author Contributions: Conceptualization, B.L., X.R. and Y.L.; Methodology, H.G. and B.L.; Software, W.L. and F.Q.; Validation, H.G., W.L. and F.Q.; Formal Analysis, X.R. and Y.L.; Writing-Review & Editing, H.G. and B.L.

Funding: This work was funded by the National Natural Science Foundation of China (No. 61773226) and Key Research and Development Program of Shandong Province (No. 2018GGX103054).

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
- 2. Huang, G.B.; Wang, D.H.; Lan, Y. Extreme learning machines: A survey. *Int. J. Mach. Learn. Cybern.* **2011**, *2*, 107–122. [CrossRef]
- Lv, W.; Mao, Z.; Jia, M. ELM Based LF Temperature Prediction Model and Its Online Sequential Learning. In Proceedings of the 24th China Conference on Control and Decision-Making, Taiyuan, China, 23–25 May 2012; pp. 2362–2365.
- 4. Yu, J.; Song, W.; Li, M.; Jianjun, H.; Wang, N. A Novel Image Classification Algorithm Based on Extreme Learning Machine. *China Commun.* **2015**, *s2*, 48–54.
- 5. Wang, X.; Wang, Y.; Ji, Z. Fault Diagnosis Algorithm of Permanent Magnet Synchronous Motor Based on Improved ELM. J. Syst. Simul. 2017, 29, 646–668.
- 6. Li, F.; Sibo, Y.; Huanhuan, H.; Wu, W. Extreme Learning Machine with Local Connections. *arXiv* 2018, arXiv:1801.06975.
- 7. Albadra, M.A.A.; Tiuna, S. Extreme Learning Machine: A Review. Int. J. Appl. Eng. Res. 2017, 12, 4610–4623.
- 8. Sun, J. Local coupled feed forward neural network. Neural Netw. 2010, 23, 108–113. [CrossRef] [PubMed]
- 9. Sun, J. Learning algorithm and hidden node selection scheme for local coupled feedforward neural network classifier. *Neurocomputing* **2012**, *79*, 158–163. [CrossRef]
- 10. Yanpeng, Q. Local coupled extreme learning machine. Neural Comput. Appl. 2016, 27, 27–33.
- 11. Mehta, R.; Vishwakarma, V.P. LC-ELM-Based Gray Scale Image Watermarking in Wavelet Domain. In *Quality, IT and Business Operations;* Springer: New York, NY, USA, 2018; pp. 191–202.
- 12. Yanpeng, Q.; Ansheng, D. The optimisation for local coupled extreme learning machine using differential evolution. *Math. Probl. Eng.* **2015**. [CrossRef]
- Hao, Z.F.; Guo, G.H.; Huang, H. A Particle Swarm Optimization Algorithm Diffrential Evolution. In Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, China, 19–22 August 2007; pp. 1031–1035.
- 14. Price, K.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces; Kluwer Academic Publishers: Norwell, MA, USA, 1997; pp. 341–359.
- 15. Xu, Y.; Shu, Y. Evolutionary Extreme Learning Machine—Based on Particle Swarm Optimization. In *Advances in Neural Networks*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2006; pp. 644–652.
- Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the 1995 IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; IEEE: Piscataway, NJ, USA, 1995; pp. 1942–1948.
- 17. Han, F.; Yao, H.F.; Ling, Q.H. An improved evolutionary extreme learning machine based on particle swarm optimization. *Neurocomputing* **2013**, *116*, 87–93. [CrossRef]
- 18. Du, J.; Liu, Y.; Yu, Y.; Yan, W. A prediction of precipitation data based on support vector machine and particle swarm optimization (pso-svm) algorithms. *Algorithms* **2017**, *10*, 57. [CrossRef]
- 19. Li, B.; Li, Y.; Rong, X. A Hybrid Optimization Algorithm for Extreme Learning Machine. In Proceedings of the China Intelligent Automation Conference, Fuzhou, China, 8–10 May 2015; pp. 297–306.
- 20. Li, B.; Li, Y.; Liu, M. A parameter adaptive particle swarm optimization algorithm for extreme learning machine. In Proceedings of the IEEE 27th Chinese Control and Decision Conference, Qingdao, China, 23–25 May 2015; pp. 1146–1160.
- 21. Liu, C.; Wang, B.; Wang, X.; He, Y.; Ashfaq, R. An Improved Local Coupled Extreme Learning Machine. *J. Softw.* **2016**, *8*, 745–755. [CrossRef]

- 22. Dasa, M.; Chakrabortyb, M.K.; Ghoshalc, T.K. Fuzzy tolerance relation, fuzzy tolerance space and basis. *Fuzzy Sets Syst.* **1998**, *97*, 361–369. [CrossRef]
- 23. Rao, C.R.; Mitra, S.K. Generalized inverse of matrices and its applications. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Theory of Statistics;* University of California Press: Oakland, CA, USA, 1972; pp. 601–620.
- 24. Li, W.T.; Shi, X.W.; Hei, Y.Q. An Improved Particle Swarm Optimization Algorithm for Pattern Synthesis of Phased Arrays. *Prog. Electromagn. Res.* **2008**, *82*, 319–332. [CrossRef]
- 25. Li, B.; Rong, X.; Li, Y. An improved kernel based extreme learning machine for robot execution failures. *Sci. World J.* **2014**. [CrossRef] [PubMed]
- 26. Shi, Y.; Eberhart, R.C. A modified particle swarm optimizer. In Proceedings of the IEEE World Conference on Computation Intelligence, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
- 27. Ratnaweera, A.; Halgamuge, S.K.; Watson, H.C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans. Evol. Comput.* **2004**, *8*, 240–255. [CrossRef]
- 28. Manavalan, B.; Shin, T.H.; Kim, M.O.; Lee, G. PIP-EL: A New Ensemble Learning Method for Improved Proinflammatory Peptide Predictions. *Front. Pharmacol.* **2018**. [CrossRef] [PubMed]
- 29. Manavalan, B.; Shin, T.H.; Kim, M.O.; Lee, G. AIPpred: Sequence-Based Prediction of Anti-inflammatory Peptides Using Random Forest. *Front. Pharmacol.* **2018**. [CrossRef] [PubMed]
- 30. Box, G.E.P.; Jenkins, G.M. Time Series Analysis, Forecasting and Control; Holden Day: San Francisco, CA, USA, 1970.
- 31. California Housing. Available online: https://www.dcc.fc.up.pt/~ltorgo/Regression/cal_housing.html (accessed on 25 October 2017).
- 32. Blake, C.L.; Merz, C.J. *Repository of Machine Learning Databases;* Department of Information and Computer Science, University of California: Irvine, CA, USA, 1998.
- 33. Li, B.; Li, Y.; Rong, X. The extreme learning machine learning algorithm with tunable activation function. *Neural Comput. Appl.* **2013**, *22*, 531–539. [CrossRef]
- Wanguo, Y.; Xu, Z.; Ashfaq, R. Comments on "Local coupled extreme learning machine". Neural Comput. Appl. 2017, 28, 631–634.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).