2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science

Improved Extreme Learning Machine Based on Artificial Bee ColonyAlgorithm

Li Mao
School of Internet of Things, Jiangnan University
Wuxi, Jiangsu 214122, China
missliyang@live.com

Yang Li, Yu Mao
School of Internet of Things, Jiangnan University
Wuxi, Jiangsu 214122, China
372902817@qq.com

***Abstract*—Artificial bee colony *(ABC)* algorithm has good performance in discovering the optimal solutions to difficult optimization problems. In this paper, we introduce an improved extreme learning machine method based on artificial bee colony optimization with the method, the defect of worse results of the traditional extreme learning machine in classification and regression is overcomed, and effectively improves the results of classification and regression.**

Key words: Artiicial bee colony; Extreme learning machine; Group intelligence;

I. Introduction

Artificial neural networks (ANN) are algorithm-oriented mathematical models for simulating the behavior characteristics of biological neural networks for distributed parallel calculation processing[1]. Therein, single-hidden layer feed forward neutral networks (SLFN) have been extensively applied to many fields due to their good learning ability. However, the values of hidden nodes are corrected with a gradient descent method in most of the traditional feed forward neutral networks, therefore, disadvantages such as slow training speed, easy coverage to local minima, and requirements for setting more parameters may easily occur.

To overcome this defect, some scholars have achieved a good effect by using an intelligent optimization algorithm in combination with the extreme learning machine. An evolutionary extreme learning machine (E-ELM) is proposed by Zhu et al[2]., where a differential evolutionary algorithm is used to optimize the parameters of hidden nodes of ELM to thereby improve the performance of ELM, but with more parameters required to be set and complex experimental process; a self-adaptive evolutionary extreme learning machine (SaE-ELM) is proposed by Cao et al[3]., where a self-adaptive evolutionary algorithm and the extreme learning machine are combined to optimize the hidden nodes, with fewer parameters set, which improves the accuracy and stability of the extreme learning machine regarding the issues of regression and classification, however, this algorithm has the defects of overlong used time and worse practicability; an extreme learning machine based on particle swarm optimization (PSO-ELM) is proposed by Wang Jie et al[4], where a particle swarm optimization algorithm is used to optimize and choose the input layer weight and hidden layer bias of the extreme learning machine to obtain an optimal network and a novel hybrid intelligent optimization algorithm (DEPSO-ELM) based on a differential evolution algorithm and a particle swarm optimization algorithm is proposed by Lin Meijin et al[5], with reference to the memetic evolution mechanism of a frog-leaping algorithm for parameter optimization, where the extreme learning machine algorithm is used to solve an output weight of SLFNs, but with excessive dependency on experimental data and worse robustness[6].

With respect to the defects occurring when the traditional extreme learning machine is applied to classification and regression, in this paper proposes an improved extreme learning machine method based on artificial bee colony optimization (DECABC-ELM) in view of the traditional extreme learning machine, which effectively increases the effects of classification and regression. The structure of this paper is organized as follows: introduction to ELM in section 2, description of proposed DECABC-ELM in section 3, experimental results and performance assessment in section 4, and finally conclusion in section 5.

II. Traditinal Elm Algorithm and ABC

For N arbitrary distinct different training sample sets (x_i, y_i) , one feed forward neural network having L hidden nodes has an output as follow: $t_i = \sum_{j=1}^{L} \beta_j g(\omega_j * x_i + b_j)$(1)In Formula (1), $\omega_j \epsilon R^d$is a connection weight from an input layer to a hidden node, $b_j \epsilon R$ is a neural threshold of the hidden node, $g()$ is an activation function of the hidden node, $g(\omega_j * x_i + b_j)$ is an output of the ith sample at the hidden node, $\omega_j * x_i$ is an inner product of a vector, and β_j is a connection weight between the hidden node and an output layer. solve a least square solution of a linear equation below to obtain an output weight $\hat{\beta}$:$\min \sum_{i=1}^{N} ||y_i - t_i|| = 0$ (2)the solution of Equation (2) is as follows:$\hat{\beta} H^{+} T$(3), In Formula (3), H^+ stands for a Moore-Penrose (MP) generalized inverse of a hidden layer output matrix substitute $\hat{\beta}$ solved in Formula (3) into Formula (1) to possibly obtain a calculation result.

The traditional artificial bee colony (ABC) optimization algorithm has the following steps: generation of an initial solution, where an initial solution is generated for SN individuals at an initialization phase, with a formula as follows: $x_{i,j} = x_j^{min} + rand[0,1](x_j^{max} - x_j^{min})$ (4)In Formula (4), $i \epsilon \{1,2,\dots,N\}$indicates the number of the initial solution, $j = 1,2,\dots,D$ indicates that each initial solution is a D-dimensional vector, rand [0,1] indicates that a random number ranging from 0 to 1 is chosen, $x_j^{max}$and x_j^{min} indicate an upper bound and a lower bound of the jth dimension of the solution, respectively. searching phase of employed bees,

978-1-5386-7445-1/18/$31.00 ©2018 IEEE
DOI 10.1109/DCABES.2018.00053

where each employed bee individual searches a new nectar source nearby a current position from an initial position, with an updating formula as follows: $v_{i,j} = x_{i,j} + rand[-1,1](x_{i,j} - x_{k,j})(5)v_{i,j}$ indicates position information of a new nectar source, $x_{i,j}$indicates position information of an original nectar source, rand [-1, 1] indicates that a random number ranging from -1 to 1 is chosen, and $x_{k,j}$indicates the jth dimension information of the kth nectar source, $k\epsilon\{1,2,\dots,SN\}$, with k≠i. a changed position is generated based on the employed bees and the new nectar source is searched. The choice probability calculation: $p_i = fitness(x_i / \sum_{j=1}^{SN} fitness(x_j))$(6)

III. PROPOSED ALGORITHM:DECABC-ELM

An improved extreme learning machine method based on artificial bee colony optimization comprises the following steps: given a training sample set (x_i, y_i) $(i = 1,2,\dots,N), x_i \in R^d, y_i \in R^m$, with an activation function of $g()$, and a number of hidden node of L, generating an initial solution for SN individuals as $x_{i,j} = x_j^{min} + rand[0,1](x_j^{max} - x_j^{min})$ wherein each individual is encoded in a manner as shown below $\theta_G = [\omega_1^T G, \dots, \omega_L^T G, b_{1,G}, \dots, b_{L,G}]$ wherein during an encoding, $\omega_j (j = 1, \dots L)$ is a D-dimensional vector, with each dimension being a random number ranging from -1 to 1, b_j is a random number ranging from 0 to 1, and G indicates an iteration number for a bee colony.then, globally optimizing a connection weight ω_j and a threshold b for an extreme learning machine: $v_{i,j} = x_{i,j} rand[-1,1](X_{best,j} - X_{k,j} + X_{l,j} - X_{m,j})$(8)wherein in the formula (8), $X_{best,j}$ stands for a currently best individual in the bee colony, $X_{k,j}$, $X_{l,j}$ and $X_{m,j}$ are another three different individuals chosen randomly except the current individual, i.e, i ≠ k ≠ l ≠ m; whenever employed bees reach a new position, a training sample set is verified by means of the connection weight ω and threshold b of the extreme learning machine and a fitness value is obtained, and under the condition of a high the fitness value is high, a new position information is used to substitute an old position information. the fitness probability calculation formula is formula(6) a concentration probability calculation

$$\begin{cases} P_d(x_i)=\frac{1}{SN}(1-\frac{HN}{SN}) & if \frac{N_i}{SN}>T \\ P_d(x_i)=\frac{1}{SN}(1+\frac{HN}{SN}*\frac{HN}{SN-HN}) & if \frac{N_i}{SN}\le T \end{cases} \quad (10)$$

wherein in Formula (10), N_i indicates the number of the onlooker bees having a fitness value approximate to the ith onlooker bee, $\frac{N_i}{SN}$ indicates a quantitative proportion of the onlooker bees approximate in fitness in the colony, T is a concentration threshold, and HN indicates the number of the onlooker bees having a concentration greater than T; the choice probability calculation formula is as follows: $P_{choose}(x_i) = \alpha P_i(x_i) + (1-\alpha) P_d(x_i)$ (11)an onlooker bee colony is chosen according to Formula (11) in a roulette form, and the first SN onlooker bees with a maximal fitness function are chosen to create a new food source information, under the condition that the iteration number reaches a set value or a mean square error value reaches an accuracy of 1e^{-4}, extracting the connection weight ω and threshold b of the extreme learning machine from best individuals.

IV. EXPERIMENTS AND PERFORMANCE ASSESSMENT

we present the performance comparison in regression data set is about 4 real-world regression data sets from the Machine Learning Library of University of California Irvine[7] were used to compare the performances of the four algorithms. The names of the data sets are Auto MPG (MPG)[8], Computer Hardware(CPU), Housing and Servo respectively[9]. In this experiment, the data in the data sets are randomly divided into a training sample set and a test sample set, with 70% as the training sample set and 30% remained as the test sample set[10]. To reduce the impacts from large variations of all the variables, we perform normalizing on the data before the algorithm is executed, i.e., an input variable normalized to [-1, 1], and an output variable normalized to [0, 1]. Across all the experiments, the hidden nodes gradually increase, and the experiment results having the mean best RMSE are recorded into Tables 1 to Table 4.

Table 1 Comparison of fitting results of Auto MPG

Algorithm Name	Test Set RMSE	Test Set Std.Dev.	Training Time (s)	Number of Hidden Nodes
SaE-ELM	0.0726	0.0019	6.6517	20
PSO-ELM	0.0739	0.0033	4.7803	20
DEPSO-ELM	0.0741	0.0043	3.7441	17
ABC-ELM	0.0745	0.0039	5.2760	21
DECABC-ELM	0.0702	0.0032	5.2039	19

Table 2 Comparison of fitting results of Computer Hardware

Algorithm Name	Test Set RMSE	Test Set Std.Dev.	Training Time (s)	Number of Hidden Nodes
SaE-ELM	0.0412	0.0148	4.2279	15
PSO-ELM	0.0386	0.0116	2.4960	13
DEPSO-ELM	0.0461	0.0120	2.0137	11
ABC-ELM	0.0516	0.0248	1.8319	11
DECABC-ELM	0.0259	0.0170	2.4466	10

Table 3 Comparison of fitting results of Computer Hardware

Algorithm Name	Test Set RMSE	Test Set Std.Dev.	Training Time (s)	Number of Hidden Nodes
SaE-ELM	0.0412	0.0148	4.2279	15

PSO-ELM	0.0386	0.0116	2.4960	13
DEPSO-ELM	0.0461	0.0120	2.0137	11
ABC-ELM	0.0516	0.0248	1.8319	11
DECABC-ELM	0.0259	0.0170	2.4466	10

Table 4 Comparison of fitting results of Housing

Algorithm Name	Test Set		Training	Number of
	RMSE	Std.Dev.	Time (s)	Hidden Nodes
SaE-ELM	0.0720	0.0049	42.4382	69
PSO-ELM	0.0642	0.0072	28.5984	67
DEPSO-ELM	0.0656	0.0064	26.7162	70
ABC-ELM	0.0748	0.0050	25.4063	68
DECABC-ELM	0.0567	0.0046	30.8024	66

Table 5 Comparison of fitting results of Servo

Algorithm Name	Test Set		Training	Number of
	RMSE	Std.Dev.	Time (s)	Hidden Nodes
SaE-ELM	0.1785	0.0094	6.4484	30
PSO-ELM	0.1877	0.0166	3.1621	22
DEPSO-ELM	0.1959	0.0090	3.1918	25
ABC-ELM	0.1958	0.0136	3.9710	30
DECABC-ELM	0.1740	0.0075	4.0030	26

V. CONCLUSION

As can be seen from the tables, DECABC-ELM obtains the minimal RMSE among all the data set fitting experiments, however, DECABC-ELM has the standard deviation worse than those of other algorithms in Auto MPG and Computer Hardware, that is, its stability needs to be improved. From the view of training time and number of hidden nodes, PSO-ELM and DEPSO-ELM have higher convergence rate and less number of used hidden nodes, but with the accuracy worse than that of DECABC-ELM. Based on the overall consideration, DECABC-ELM, i.e. the algorithm as described in the present invention, has a superior performance.

ACKNOWLEDGMENT

This research is supported in part by the National Science and Technology Support Program of China (no. 2015 BAD17B02), Industry, Education and Research Program of Jiangsu Province（no. BY2015019-30）.

REFERENCES

[1] Artificial Neural Networks[M]. Springer International Publishing, 2017.

[2] Zhu Q Y, Qin A K, Suganthan P N, et al. Evolutionary extreme learning machine[J]. Pattern Recognition, 2005, 38(10):1759-1763.

[3] Cao J, Lin Z, Huang G B. Self-Adaptive Evolutionary Extreme Learning Machine[J]. Neural Processing Letters, 2012, 36(3):285-305.

[4] Pacifico L D S, Ludermir T B. Evolutionary extreme learning machine based on particle swarm optimization and clustering strategies[C]// International Joint Conference on Neural Networks. IEEE, 2014:1-6.

[5] Lin M J, Fei L, Cai-Hong S U, et al. An improved hybrid intelligent extreme learning machine[J]. Control & Decision, 2015, 30(6):1078-1084.

[6] Jalilzadeh M S S, Sharifi E. Optimal Capacitor Placement in a Radial DistributionSystem using Shuffled Frog Leaping and ParticleSwarm Optimization Algorithms[J]. International Journal of Network Security (2152-5064), 2011.

[7] Abramavicius D, Ahmed M, Ahn C B, et al. University of California at Irvine[J].

[8] Wang J, Dong Z, Zhang W, et al. Rule extraction of support vector machine based on regression tree and application[J]. Computer Engineering & Applications, 2017.

[9] Kai J. FEM analysis of the clearance between the housing body and servo control valve for the main pump of 22T hydraulic excavator[J]. 2014, 42:36-41.

[10] Liu K Y, Smith M R, Fear E C, et al. Evaluation and amelioration of computer-aided diagnosis with artificial neural networks utilizing small-sized sample sets[J]. Biomedical Signal Processing & Control, 2013, 8(3):255-262.