

Hindawi
Scientific Programming
Volume 2018, Article ID 2943290, 7 pages
<https://doi.org/10.1155/2018/2943290>



Research Article

Developing a Novel Hybrid Biogeography-Based Optimization Algorithm for Multilayer Perceptron Training under Big Data Challenge

Xun Pu,¹ ShanXiong Chen,¹ XianPing Yu,¹ and Le Zhang^{1,2} 

¹College of Computer & Information Science, Southwest University, Chongqing, China

²College of Computer Science, Sichuan University, Chengdu, China

Correspondence should be addressed to Le Zhang; zhangle06@scu.edu.cn

Received 24 August 2017; Revised 8 December 2017; Accepted 18 January 2018; Published 1 March 2018

Academic Editor: Anfeng Liu

Copyright © 2018 Xun Pu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A Multilayer Perceptron (MLP) is a feedforward neural network model consisting of one or more hidden layers between the input and output layers. MLPs have been successfully applied to solve a wide range of problems in the fields of neuroscience, computational linguistics, and parallel distributed processing. While MLPs are highly successful in solving problems which are not linearly separable, two of the biggest challenges in their development and application are the local-minima problem and the problem of slow convergence under big data challenge. In order to tackle these problems, this study proposes a Hybrid Chaotic Biogeography-Based Optimization (HCBBO) algorithm for training MLPs for big data analysis and processing. Four benchmark datasets are employed to investigate the effectiveness of HCBBO in training MLPs. The accuracy of the results and the convergence of HCBBO are compared to three well-known heuristic algorithms: (a) Biogeography-Based Optimization (BBO), (b) Particle Swarm Optimization (PSO), and (c) Genetic Algorithms (GA). The experimental results show that training MLPs by using HCBBO is better than the other three heuristic learning approaches for big data processing.

1. Introduction

The term big data [1–3] had been developed to describe the phenomenon of the increasing size of massive datasets in scientific experiments, financial trading, and networks. Since big data is always of big volume and has multiple varied types and fast update velocity [4], it is urgent for us to develop such a tool that can extract the meaningful information from big data. Neural networks (NNs) [5, 6] are one of popular machine learning computational approaches, which are composed of several simple and interconnected processing elements and good at loosely modeling the neuronal structures of the human brain. A neural network can be represented as a highly complex nonlinear dynamic system [5], which has some unique characteristics: (a) high dimensionality, (b) extensive interconnectivity, (c) adaptability, and (d) ability to self-organize.

In the last decade, feedforward neural networks (FNNs) [6] have gained popularity in various areas of machine learning [7] and big data mining [1] to solve classification and

regression problems. While the two-layered FNN is the most popular neural network used in practical applications, it is not suitable for solving nonlinear problems [7, 8]. The Multilayer Perceptron (MLP) [9, 10], a feedforward neural network with one or more hidden layers between the input and the output layers, is more successful in dealing with nonlinear problems such as pattern classification, big data prediction, and function approximation. Previous research [11] shows that MLPs with one hidden layer are able to approximate any continuous or discontinuous function. Therefore, the study of MLPs with one hidden layer has gained a lot of attention from the research community.

Theoretically, the goal of the learning process of MLPs is to find the best combination of weights and biases of the connections in order to achieve minimum error for the given train and test data. However, one of the most common problems of training an MLP is that there is a tendency for the algorithm to converge on a local minimum. Since an MLP can consist of multiple local minima, it is easy to be trapped in

one of them rather than converging on the global minimum. This is a common problem in most gradient-based learning approaches such as backpropagation (BP) based NNs [12]. According to Mirjalili's research [13], the initial values of the learning rate and the momentum can also affect the convergence in case of BP based NNs, with unsuitable values for these variables resulting in their divergence. Thus, many studies focus on using novel heuristic optimization methods or evolutionary algorithms to resolve the problems of MLP learning algorithms [14]. Classical applied approaches are Particle Swarm Optimization (PSO) algorithms [15, 16], Ant Colony Optimization (ACO) [17], and Artificial Bee Colony (ABC) [18]. However, the No Free Lunch (NFL) theorem [19, 20] states that no heuristic algorithm is best suited for solving all optimization problems. Most of them have their own side effects and overall there has been no significant improvement [13] using these approaches. For example, Genetic Algorithms (GA) may reduce the probability of getting trapped in a local minimum, but they still suffer from slow convergence rates.

Recently, a novel optimization method called Biogeography-Based Optimization (BBO) [21] has been proposed. It is based on the motivation that geographical distribution of biological organisms can be represented by mathematical equations. It is a distributed paradigm, which seeks to simulate the collective behavior of unsophisticated individuals interacting locally with their environment to efficiently identify optimum solutions in complex search spaces. There are many related works of research [22–25] which show that the BBO algorithm is a type of evolutionary algorithm which can offer a specific evolutionary mechanism for each individual in a population. This mechanism makes the BBO algorithm more successful and robust on nonuniform training procedures than gradient-based algorithms. Moreover, compared with the PSO or ACO, the mutation operator of the BBO algorithm can enhance their exploitation capability. This allows the BBO algorithm to outperform PSOs in training MLPs. This has led to a great interest in applying the efficiency of BBO in training MLPs. In 2010, Ovreiu and Simon [24] trained a neuro-fuzzy network with BBO for classifying P-wave features for the diagnosis of cardiomyopathy. Research [13] used 11 standard datasets to provide a comprehensive test bed for investigating the abilities of the BBO algorithm in training MLPs. In this paper, we propose a hybrid BBO with chaotic maps trainer (HCBBO) for MLPs. Our approach employs chaos theory to improve the performance of the BBO with very little computational burden. In our algorithm, the migration and mutation mechanisms are combined to enhance the exploration and exploitation abilities of BBO, and a novel migration operator is proposed to improve BBO's performance in training MLPs.

The rest of this paper is organized as follows. In Section 2, a brief review of the MLP notation and a simple first-order training method are provided. In Sections 3 and 4, the HCBBO framework is introduced and analyzed. In Section 5, the computational results to demonstrate the effectiveness of the proposed improved hybrid algorithm are provided. Finally, Section 6 provides concluding remarks and suggests some directions for future research.

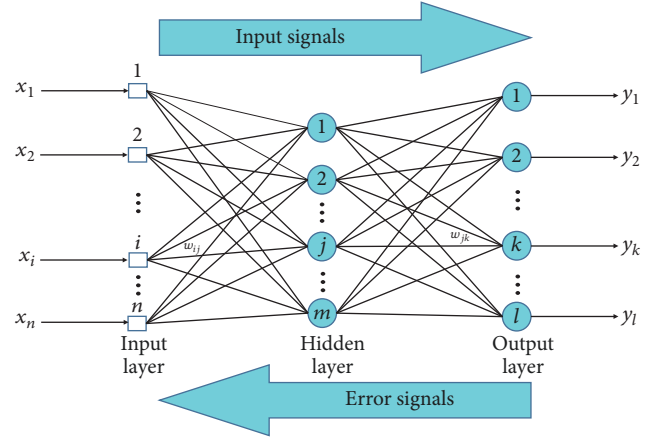


FIGURE 1: An MLP with one hidden layer.

2. Review of the MLP Notation

The notation used in the rest of the paper represents a fully connected feedforward MLP network with a single hidden layer (as shown in Figure 1). This MLP consists of an input layer, an output layer, and a single hidden layer. The MLP is trained using a backpropagation (BP) learning algorithm. Let n denote the number of input nodes, m denote the number of hidden nodes, and l denote the number of output nodes. Let the input weights $w_{i,j}$ connect the i th input to the j th hidden unit and output weights $w_{out(j,k)}$ connect the j th hidden unit to the k th output. The weighted sums of inputs are first calculated by the following equation:

$$s_j = \sum_{i=1}^n (w_{ij}x_i) - \theta_j, \quad j = 1, 2, \dots, m, \quad (1)$$

where n is the number of the input nodes, w_{ij} is the connection weight from the i th node in the input layer to the j th node in the hidden layer, x_i indicates the i th input, and θ_j means the threshold of the j th hidden node.

The output of each hidden node is calculated as follows:

$$f(j) = \frac{1}{(1 + \exp(-s_j))} \quad j = 1, 2, \dots, m. \quad (2)$$

After calculating outputs of the hidden nodes, the final output can be defined as follows:

$$o_k = \sum_{j=1}^m W_{jk} \cdot f(j) - \theta'_k \quad k = 1, 2, \dots, l, \quad (3)$$

where W_{jk} is the connection weight from the j th hidden node to the k th output node and θ'_k is the bias of the k th output node.

The learning error E (fitness function) is calculated as follows:

$$E_k = \sum_{i=1}^l (o_i^k - d_i^k)^2, \quad (4)$$

$$E = \sum_{k=1}^q \frac{E_k}{q}$$

where q is the number of training samples, l is the number of outputs, d_i^k is the desired output of the i th input unit when the k th training sample is used, and o_i^k is the actual output of the i th input unit when the k th training sample is used.

From the above equations, it can be observed that the final value of the output in MLPs depends upon the parameters of the connecting weights and biases. Thus, training an MLP can be defined as the process of finding the optimal values of the weights and biases of the connections in order to achieve the desirable outputs from certain given inputs.

3. The Proposed Hybrid BBO for Training an MLP

Biogeography-Based Optimization (BBO) is a population-based optimization algorithm inspired by evolution and the balance of predators and preys in different ecosystems. Experiments show that results obtained using the BBO are at least competitive with other population-based algorithms. It has been shown to outperform some well-known heuristic algorithms such as PSO, GA, and ACO on some real-world problems and benchmark functions [21].

The steps of the BBO algorithm can be described as follows. In the beginning, the BBO generates a random number of search agents named habitats, which are represented as vectors of the variables in the problem (analogous to chromosomes in GA). Next, each agent is assigned emigration, immigration, and mutation rates which simulate the characteristics of different ecosystems. In addition, a variable called HSI (the habitat suitability index) is defined to measure the fitness of each habitat. Here, a higher value of HSI indicates that the habitat is more suitable for the residence of biological species. In other words, a solution of the BBO with a high value of HSI indicates a superior result, while a solution with a low value of HSI indicates an inferior result.

During the course of iterations, a set of solutions is maintained from one iteration to the next, and each habitat sends and receives habitants to and from different habitats based on their immigration and emigration rates which are probabilistically adapted. In each iteration, a random number of habitants are also occasionally mutated. That makes each solution adapt itself by learning from its neighbors as the algorithm progresses. Here, each solution parameter is denoted as a suitability index variable (SIV).

The process of BBO is composed of two phases: migration and mutation. During the migration phase, immigration (λ_k) and emigration (μ_k) rates of each habitat follow the model as depicted in Figure 2. A high number of habitants in a habitat increase the probability of emigration and decrease the probability of immigration. During the mutation phase, the mutation factor in BBO keeps the distribution of habitants in a habitat as diverse as possible. In contrast with the mutation factor in GA, the mutation factor of BBO is not set randomly; it is dependent on the probability of the number of species in each habitat.

The mathematical formula of immigration (λ_k) and emigration (μ_k) can be written as follows:

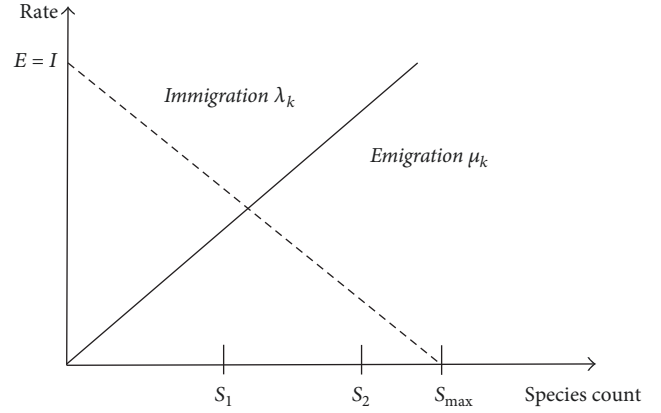


FIGURE 2: Species model of a habitat.

$$\begin{aligned}\lambda_k &= I \left(1 - \frac{S_k}{S_{\max}} \right), \\ \mu_k &= E \left(\frac{S_k}{S_{\max}} \right),\end{aligned}\quad (5)$$

where I is the maximum immigration rate, E is the maximum emigration rate, S_{\max} is the maximum number of habitats, and S_k is the habitat count of k .

The mutation of each habitat, which improves the exploration of BBO, is defined as follows:

$$m(s) = m_{\max} \times \left(1 - \frac{P_n}{P_{\max}} \right). \quad (6)$$

Here m_{\max} is the maximum value of mutation defined by user, P_{\max} is the greatest mutation probability of all the habitats, and P_n is the mutation probability of the n th habitat, which can be obtained as

$$\begin{aligned}\dot{P}_n &= \begin{cases} -(\lambda_n + \mu_n) P_n + \mu_{n+1} P_{n+1}, & n = 0; \\ -(\lambda_n + \mu_n) P_n + \mu_{n+1} P_{n+1} + \lambda_{n-1} P_{n-1}, & 1 \leq n \leq S_{\max} - 1 = 0; \\ -(\lambda_n + \mu_n) P_n + \lambda_{n-1} P_{n-1}, & n = S_{\max}. \end{cases} \quad (7)\end{aligned}$$

The complete process of the BBO algorithm is described in Algorithm 1; here $I : \phi \rightarrow \{H^n, \text{HSI}^n\}$ initializes an ecosystem of habitats and computes each corresponding HSI and $\Gamma = (n, m, \lambda, \tau, \Omega, M)$ is a transition function which modifies the ecosystem from one optimization iteration to the next. The elements of the 6-tuple can be defined as follows: n is the number of habitats; m is the number of SIVs; λ is the immigration rate; τ is the emigration rate; Ω is the migration operator; and M is the mutation operator.

4. The Proposed Hybrid CBBO Algorithm for Training an MLP

There are three different approaches for using heuristic algorithms for training MLPs. In the first approach, heuristic algorithms are employed to find a combination of weights

```

I :  $\phi \rightarrow \{H^n, \text{HSI}^n\}$ 
While (condition = T)
   $\Gamma = (n, m, \lambda, \tau, \Omega, M)$ 
end

```

ALGORITHM 1: Pseudocode of BBO for optimization problems.

and biases to provide the minimum error for an MLP. In the second approach, heuristic algorithms are utilized to find the proper architecture for an MLP to be applied to a particular problem. In the third approach, heuristic algorithms can be used to tune the parameters of a gradient-based learning algorithm.

Mirjalili et al. [13] employed the basic BBO algorithm to train an MLP using the first approach, and the results demonstrate that BBO is significantly better at avoiding local minima compared to PSO, GA, and ACO algorithms. However, the basic BBO algorithm still has some drawbacks, such as (a) the large number of iterations needed to reach the global optimal solution and (b) the tendency to converge to solutions which may be locally the best. Many methods have been proposed to improve the capabilities for the exploration and exploitation of the BBO algorithm.

4.1. Chaotic Systems. Chaos theory [26] refers to the study of chaotic dynamical systems, which is embodied by the so-called “butterfly effect.” As nonlinear dynamical systems, chaotic systems are highly sensitive to their initial conditions, and tiny changes to their initial conditions may result in significant changes in the final outcomes of these systems.

In this paper, chaotic systems are applied to BBOs instead of random values [25–27] for their initialization. This means that chaotic maps substitute the random values to provide chaotic behaviors to heuristic algorithms. During the processing of the BBO algorithm, the most important random values are calculated to choose a habitat for emigrating the new habitants during the migration phase. We utilize chaotic maps, which use the logistic model in (8), and choose a value from the interval of $[0, 1]$, whenever there is a need for a random value.

$$x_{n+1} = f(x_n) = \frac{\mu}{4} \sin(\pi x_n); \quad (8)$$

here $x_{n+1} \in [0, 1]$ and μ are named logistic parameters. When μ equals 4, the iterations produce values which follow a pseudorandom distribution. This means that a tiny difference in the initial value of x_1 will give rise to a large difference in its long-time behavior. We employ this feature to avoid a local convergence of the BBO algorithm.

4.2. Habitat Suitability Index (Fitness Function). During the training phase of an MLP, each training data sample should be involved in calculating the HIS of each candidate solution.

In this work, the Mean Square Error (MSE) is utilized for evaluating all training samples. The MSE is defined as follows:

$$E = \sum_{k=1}^q \frac{\sum_{i=1}^l (o_i^k - d_i^k)^2}{q}; \quad (9)$$

here q is the number of training samples, l is the number of outputs, d_i^k is the desired output of the i th input unit when the k th training sample is used, and o_i^k is the actual output of the i th input unit when the k th training sample is used. Thus, the HSI value for the i th candidate is given by $\text{HSI}(c_i) = E(c_i)$.

4.3. Opposition-Based Learning. To improve the convergence of BBO algorithm during the mutation phase, a method named opposition-based learning (OBL) has been used in [22]. The main idea of opposition-based learning is to consider an estimate and its opposite at the same time to achieve a better approximation of the current candidate solution.

Assuming that $X = (x_1, x_2, \dots, x_n)$ represents a vector of the weights and biases in the MLP, with $x_i \in R$ and $x_i \in [\min_i, \max_i] \forall i \in \{1, 2, \dots, n\}$, then the definition of the opposite vector is $X' = (x'_1, x'_2, \dots, x'_n)$ with its elements as $x'_i = \min_i + \max_i - x_i$. The algorithm for the OBL method can be described as follows:

- (1) Generate a vector $X = (x_1, x_2, \dots, x_n)$ and its opposite $X' = (x'_1, x'_2, \dots, x'_n)$, in an n -dimensional search space.
- (2) Evaluate the fitness of both points, $\text{HSI}(X)$ and $\text{HSI}(X')$.
- (3) If $\text{HSI}(X) \leq \text{HSI}(X')$, then replace X with X' ; otherwise, continue with X .

Thus, the vector and its opposite vector are evaluated simultaneously to obtain the fitter one.

4.4. Outline of HCBBO for MLP. In this section, the main procedure of HCBBO is described. To guarantee an initial population with a certain quality and diversity, the initial population is generated using a combination of the chaotic system and the OBL approach. By fusing the local search strategies with the migration and mutation phases of the BBO algorithm, the exploration and exploitation capabilities of the HCBBO can be well balanced. The main procedure of our proposed HCBBO to train an MLP can be described as Algorithm 2.

5. Experimental Analysis

This study focuses on finding an efficient training method for MLPs. To evaluate the performance of the proposed HCBBO algorithm in this paper, a series of experiments were developed using the Matlab software environment (V2009). The system configuration is as follows: (a) CPU: Intel i7; (b) RAM: 4 GB; (c) operating system: Windows 8. Based on the works described in [13, 28, 29], we choose four publicly available classification big datasets to benchmark our system: (1) balloon, (2) iris, (3) heart, and (4) vehicle. All these datasets are freely available from the University of California at Irvine (UCI) Machine Learning Repository [30], thus ensuring replicability. And the characteristics of these datasets are listed in Table 1.

- (1) **input:** habitat size n , maximum migration rate E and I (emigration and immigration rate), the maximum mutation rate M_{\max} ;
- (2) Initialize set of MLPs (habitats) by chaos maps on formula Eq. (8);
- (3) For each habitat, calculate its mean square error by relative parameters based on formulas (9). And the basic rule of fitness function is the better performance maintains the smaller value of MSE. Then elite habitats are identified by the values of HSI.
- (4) Combing MLPs according to immigration and emigration rates based on Eq. (6)
Probabilistically use immigration and emigration to modify each non-elite habitat based on Eq. (7).
- (5) Select number of MLPs and recomputed (mutate) some of their weights or biases by chaos maps.
- (6) Save some of the MLPs with low MSE;
- (7) This loop will be terminated if a predefined number of generations are reached or an acceptable problem solution has been found, otherwise go to step (3) for the next iteration.
- (8) **output:** the MLP with minimum MSE (HSI).

ALGORITHM 2: The framework of HCBBO algorithm.

TABLE 1: Classification datasets.

Classification datasets	Number of attributes	Number of training samples	Number of test samples	Number of classes
Balloon	4	16	16 as training samples	2
Iris	4	150	150 as training samples	3
SPECT Heart	22	80	187	2
Vehicle	18	400	446	4

TABLE 2: The main parameters of BBO and HCBBO.

Maximum number of generations: $T = 300$	Maximum mutation rate: $M_{\max} = 0.005$
Elitism parameter: $e = 5$	Maximum possible emigration rate: $E = 1$
Population size: $P_{\max} = 200$	Maximum possible immigration rate: $I = 1$

TABLE 3: MLP structure parameters.

	Balloon	Iris	SPECT Heart	Vehicle
Input	4	4	22	18
Hidden	9	9	45	52
Output	1	3	1	4

In this paper, we compare the performances of 4 algorithms, BBO, PSO, GA, and HCBBO, over the benchmark datasets described in Table 1. Since manually choosing appropriate parameters for each of these algorithms is time-consuming, the initial parameters and property structures for both the classical BBO algorithm and HCBBO algorithms (which were adjusted as Table 2) were chosen as in paper [13].

In order to increase the accuracy of the experiment, each algorithm was run 20 times, and different MLP structures will be used to deal with different datasets, which were listed in Table 3.

The running time (RT) and convergence curves of each algorithm are shown in Figures 3–7. From Figure 3, it can be observed that the average computational time of HCBBO is 8 to 13% lower than the best time obtained for the BBO. It is also lower than the computational time of all the other algorithms compared in this experiment. This decrease in the running time can be attributed to the fact that the HCBBO’s search ability was enhanced by OBL.

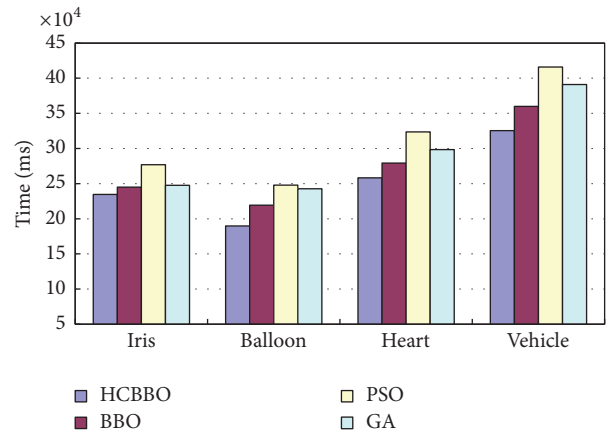


FIGURE 3: Total running time of each algorithm.

The convergence curves in Figures 4–7 show that, among all the algorithms, HCBBO has the fastest convergence behavior on all the datasets. In Figure 4, under the same experimental conditions, HCBBO achieved the optimal values for its parameters after 150 generations while BBO could not converge to an optimal value even after 200 generations. The same pattern in faster convergence for the HCBBO was observed for the other classical problems (Figures 5–7). Statistically speaking, HCBBO performs the best on all the classification datasets, since it is able to avoid local minima better than any

TABLE 4: Experimental results for classification rate.

Algorithm	Iris Classification rate	Heart Classification rate	Balloon Classification rate	Vehicle Classification rate
HCBBO	93%	81.2%	100%	76.2%
BBO	90%	75.4%	100%	71.7%
PSO	38%	66.5%	100%	56.8%
GA	88.2%	56.9%	100%	59.9%

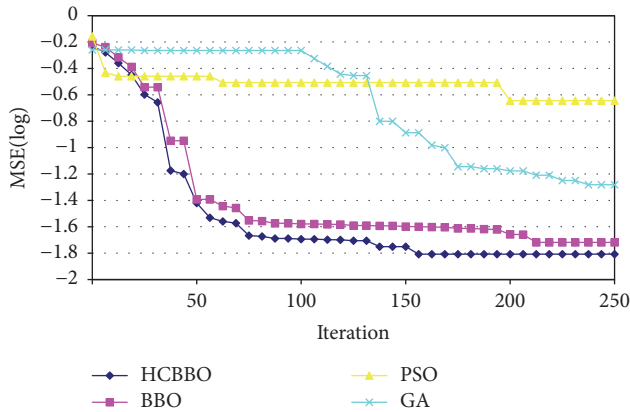


FIGURE 4: Convergence curves of algorithms for iris dataset.

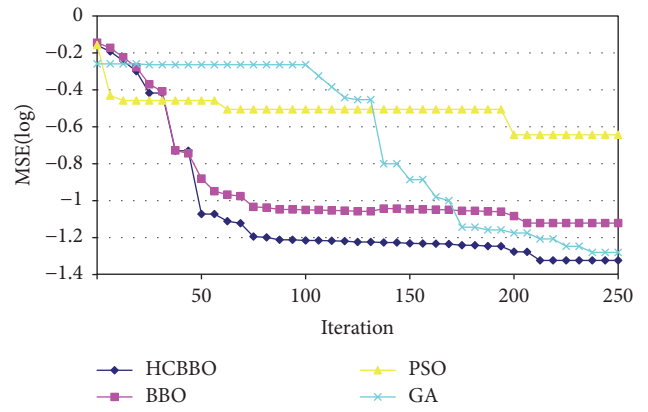


FIGURE 7: Convergence curves of algorithms for vehicle dataset.

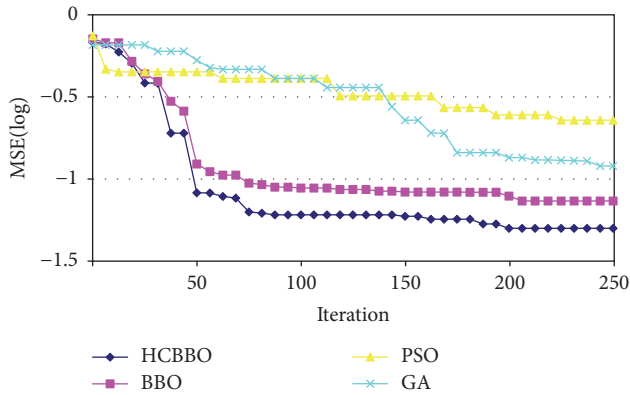


FIGURE 5: Convergence curves of algorithms for heart dataset.

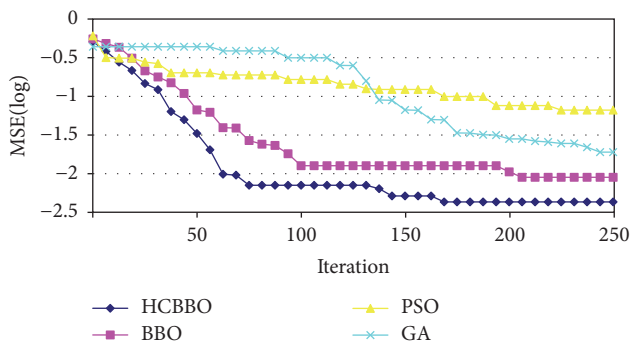


FIGURE 6: Convergence curves of algorithms for balloon dataset.

other algorithm. And the classification results obtained by HCBBO are better than all other algorithms for the chosen datasets.

The experimental results of mean classification rate are provided in Table 4. Statistically speaking, HCBBO has the best results in all of the classification datasets because it avoids local minima better.

6. Discussion and Conclusions

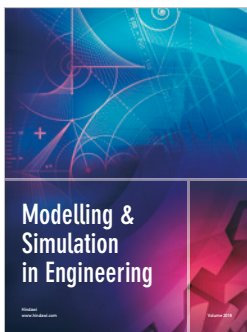
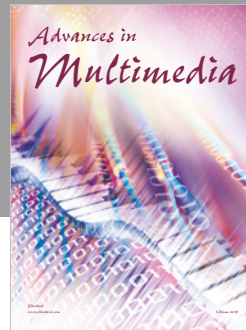
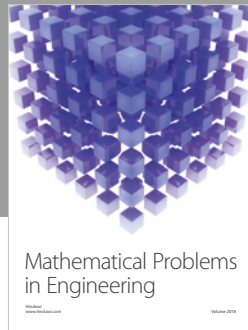
In this paper, a HCBBO algorithm was presented for training an MLP. Four benchmark big datasets (balloon, iris, heart, and vehicle) were employed to investigate the effectiveness of HCBBO in training MLPs. The performance results were statistically compared with three state-of-the-art algorithms: BBO, PSO, and GA. The main contributions and innovations of this work are summarized as follows: (a) this is the first research work combining a hybrid chaos system with the BBO algorithm to train MLPs; (b) the method named OBL was used in the mutation operator of HCBBO to improve the convergence of the algorithm; and (c) the results demonstrate that HCBBO has better convergence capabilities than BBO, PSO, and GA. In the future, we will apply the trained neural networks to analyze the big medical data and integrate more novel data mining algorithms [29, 31–35] into HCBBO.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] C. Kacfar Emani, N. Cullot, and C. Nicolle, "Understandable big data: a survey," *Computer Science Review*, vol. 17, pp. 70–81, 2015.
- [2] F. J. Alexander, A. Hoisie, and A. Szalay, "Big data [Guest editorial]," *Computing in Science & Engineering*, vol. 13, no. 6, Article ID 6077842, pp. 10–12, 2011.
- [3] D. Boyd and K. Crawford, "Critical questions for big data: provocations for a cultural, technological, and scholarly phenomenon," *Information Communication and Society*, vol. 15, no. 5, pp. 662–679, 2012.
- [4] P. Hitzler and K. Janowicz, *Linked Data, Big Data, and the 4th Paradigm*, IOS Press, 2013.
- [5] B. Irie and S. Miyake, "Capabilities of three-layered perceptrons," in *Proceedings of 1993 IEEE International Conference on Neural Networks (ICNN '93)*, pp. 641–648, San Diego, CA, USA, 1988.
- [6] T. L. Fine, "Feedforward neural network methodology," *Information Science & Statistics*, vol. 12, no. 4, pp. 432–433, 1999.
- [7] C. W. Deng, G. B. Huang, J. Xu, and J. X. Tang, "Extreme learning machines: new trends and applications," *Science China Information Sciences*, vol. 58, no. 2, pp. 1–16, 2015.
- [8] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*, MIT Press, 1995.
- [9] R. C. Odom, P. Paul, S. S. Diocee, S. M. Bailey, D. M. Zander, and J. J. Gillespie, "Shaly sand analysis using density-neutron porosities from a cased-hole pulsed neutron system," in *Proceedings of the SPE Rocky Mountain Regional Meeting*, Gillette, Wyoming, 1999.
- [10] N. A. Mat Isa and W. M. F. W. Mamat, "Clustered-hybrid multilayer perceptron network for pattern recognition application," *Applied Soft Computing*, vol. 11, no. 1, pp. 1457–1466, 2011.
- [11] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [12] D. Rumelhart and J. McClelland, *Learning Internal Representations by Error Propagation*, MIT Press, 1988.
- [13] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Let a biogeography-based optimizer train your multi-layer perceptron," *Information Sciences*, vol. 269, pp. 188–209, 2014.
- [14] A. Van Ooyen and B. Nienhuis, "Improving the convergence of the back-propagation algorithm," *Neural Networks*, vol. 5, no. 3, pp. 465–471, 1992.
- [15] I. A. A. Al-Hadi, S. Z. M. Hashim, and S. M. H. Shamsuddin, "Bacterial foraging optimization algorithm for neural network learning enhancement," in *Proceedings of the 2011 11th International Conference on Hybrid Intelligent Systems, HIS 2011*, pp. 200–205, Malaysia, 2011.
- [16] V. G. Gudise and G. K. Venayagamoorthy, "Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks," in *Proceedings of the IEEE Swarm Intelligence Symposium (SIS '03)*, pp. 110–117, Indianapolis, Ind, USA, 2003.
- [17] C. Blum and K. Socha, "Training feed-forward neural networks with ant colony optimization: an application to pattern classification," in *Proceedings of the 5th International Conference on Hybrid Intelligent Systems (HIS '05)*, pp. 233–238, 2005.
- [18] M. Karacor, K. Yilmaz, and F. Erfan Kuyumcu, "Modeling MCSRM with artificial neural network," in *Proceedings of the 2007 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) and Electromotion '07*, pp. 849–852, Bodrum, Turkey, 2007.
- [19] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, no. 237, pp. 82–117, 2013.
- [20] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [21] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [22] M. Ergezer and D. Simon, "Oppositional biogeography-based optimization for combinatorial problems," in *Proceedings of the 2011 IEEE Congress of Evolutionary Computation, CEC 2011*, pp. 1496–1503, New Orleans, LA, USA, 2011.
- [23] S. S. Malalur, M. T. Manry, and P. Jesudhas, "Multiple optimal learning factors for the multi-layer perceptron," *Neurocomputing*, vol. 149, pp. 1490–1501, 2015.
- [24] M. Ovreiu and D. Simon, "Biogeography-based optimization of neuro-fuzzy system parameters for diagnosis of cardiac disease," in *Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference, GECCO-2010*, pp. 1235–1242, New York, NY, USA, 2010.
- [25] W. Zhu and H. Duan, "Chaotic predator-prey biogeography-based optimization approach for UCAV path planning," *Aerospace Science and Technology*, vol. 32, no. 1, pp. 153–161, 2014.
- [26] S. H. Kellert, "Books-received - in the wake of chaos - unpredictable order in dynamical systems," vol. 267, *Science*, 95 edition, 1995.
- [27] L. Zhang, Y. Xue, B. Jiang et al., "Multiscale agent-based modelling of ovarian cancer progression under the stimulation of the STAT 3 pathway," *International Journal of Data Mining and Bioinformatics*, vol. 9, no. 3, pp. 235–253, 2014.
- [28] S. Mirjalili, S. Z. Mohd Hashim, and H. Moradian Sardroudi, "Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm," *Applied Mathematics and Computation*, vol. 218, no. 22, pp. 11125–11137, 2012.
- [29] L. Zhang and S. Zhang, "Using game theory to investigate the epigenetic control mechanisms of embryo development: comment on: epigenetic game theory: how to compute the epigenetic control of maternal-to-zygotic transition "by Qian Wang et al" ," *Physics of Life Reviews*, vol. 20, pp. 140–142, 2017.
- [30] C. J. M. C. Blake, *Repository of Machine Learning Databases*, <http://archive.ics.uci.edu/ml/datasets.html>.
- [31] B. Jiang, W. Dai, A. Khaliq, M. Carey, X. Zhou, and L. Zhang, "Novel 3D GPU based numerical parallel diffusion algorithms in cylindrical coordinates for health care simulation," *Mathematics and Computers in Simulation*, vol. 109, pp. 1–19, 2015.
- [32] H. Peng, T. Peng, J. Wen et al., "Characterization of p38 MAPK isoforms for drug resistance study using systems biology approach," *Bioinformatics*, vol. 30, no. 13, pp. 1899–1907, 2014.
- [33] Y. Xia, C. Yang, N. Hu et al., "Exploring the key genes and signaling transduction pathways related to the survival time of glioblastoma multiforme patients by a novel survival analysis model," *BMC Genomics*, vol. 18, no. Suppl 1, 2017.
- [34] L. Zhang, M. Qiao, H. Gao et al., "Investigation of mechanism of bone regeneration in a porous biodegradable calcium phosphate (CaP) scaffold by a combination of a multi-scale agent-based model and experimental optimization/validation," *Nanoscale*, vol. 8, no. 31, pp. 14877–14887, 2016.
- [35] L. Zhang, Y. Liu, M. Wang et al., "EZH2-, CHD4-, and IDH-linked epigenetic perturbation and its association with survival in glioma patients," *Journal of Molecular Cell Biology*, 2017.



Hindawi

Submit your manuscripts at
www.hindawi.com

