Engineering Science and Technology, an International Journal 23 (2020) 1342–1350

HOSTED BY

Contents lists available at ScienceDirect

Engineering Science and Technology, an International Journal

journal homepage: www.elsevier.com/locate/jestch

Full Length Article

Training multi-layer perceptron with artificial algae algorithm

Bahaeddin Turkoglu, Ersin Kaya *

Dept. of Computer Engineering, Konya Technical University, Konya, Turkey

ARTICLE INFO

Article history:
Received 6 January 2020
Revised 30 June 2020
Accepted 4 July 2020
Available online 19 July 2020

Keywords:
Artificial algae algorithm
Training multi-layer perceptron
Optimization

ABSTRACT

Artificial Neural Networks are commonly used to solve problems in many areas, such as classification, pattern recognition, and image processing. The most challenging and critical phase of an Artificial Neural Networks is related with its training process. The main challenge in the training process is finding optimal network parameters (i.e. weight and biase). For this purpose, numerous heuristic algorithms have been used. One of them is Artificial Algae Algorithm, which has a nature-inspired metaheuristic optimization algorithm. This algorithm is capable of successfully solving a wide variety of numerical optimization problems. In this study, Artificial Algae Algorithm is proposed for training Artificial Neural Network. Ten classification datasets with different degrees of difficulty from the UCI database repository were used to compare the proposed method performance with six well known swarm-based optimization and backpropagation algorithms. The results of the study show that Artificial Algae Algorithm is a reliable approach for training Artificial Neural Networks.

1. Introduction

Artificial Neural Network (ANN) is one of the most popular study topics in machine learning and artificial intelligence. Over the past two decades, ANN has been applied to a wide range of fields such as classification, regression, prediction problems, pattern recognition, robotics, and signal processing [1–9].

ANN must be trained to produce accurate output values for the input data presented to it [10]. Training is the process of calculating the best weight and biase from the data presented to a network system. The purpose of learning in ANN is minimizing the error between the desired and calculated network outputs. Training is done by fine-tuning the network weight and bias values. The training process for artificial neural networks directly affects the performance of the network. In the literature, many deterministic methods for the training of feedforward ANN have been proposed. The deterministic methods do not involve randomness, and produce the same result when running with the same initial parameters. Deterministic methods are mostly based on gradient. Gradient-based methods benefit from the derivative of the objective function. These methods do not guarantee to find a global optimum solution for problems where the objective function has local optimum. The backpropagation algorithm and its variants are considered conventional examples of gradient-based methods [11]. In general, the advantages of gradient-based methods are speed and simplicity; and the disadvantages are the tendency to stick to local optimum, high dependence on initial parameters, and early and slow convergence [12–15]. As an alternative to gradient-based approaches in the literature, metaheuristic algorithms have been proposed to train ANNs. Metaheuristic algorithms begin with random solutions to the training process and improve the solution throughout the iterations to reduce the error. Metaheuristic algorithms are more suitable for global optimization [16]. The advantage of such methods is their high success in avoiding local optimizations. However, it is usually much slower than deterministic approaches [17]. Metaheuristic methods have been used in the training process of ANN, and it was suggested that these algorithms were better than the gradient-based algorithm when the problem was more complex and multidimensional [18]. Swarm intelligence optimization algorithms are an essential part of metaheuristic methods, and have been used successfully in many real-world problems and in the engineering field [19,20].

Swarm intelligence is the modelling of species that behave intelligently in the form of swarms such as birds [21,22], wolves [23], whales [24], fireflies [25], and bats [26]. Every agent (individual) in the swarm represents a possible solution. Each self-organized agent sharing information between agents in the swarm will try to move to the appropriate solution [16]. There are a lot of studies in the literature using swarm intelligence for the training of ANN such as Particle Swarm Optimization [27,28], Cuckoo Search Algorithm [29], Bat Optimization Algorithm [30], Grey Wolf Optimization [31], Whale Optimization Algorithm [15], Firefly

* Corresponding author.
E-mail address: ekaya@ktun.edu.tr (E. Kaya).
Peer review under responsibility of Karabuk University.

https://doi.org/10.1016/j.jestch.2020.07.001
2215-0986/

algorithm [32], Grasshopper Optimization [33], and Dragonfly Algorithm [34].

Although many algorithms have been proposed in the literature for ANN training, new algorithms need to be developed to overcome problems such as early convergence and being stuck to local minimums. According to the No-Free-Lunch Theorem (NFL), an optimization algorithm cannot achieve success in all of the optimization problems [35–37]. Motivated by these reasons, in this study, the Artificial Algae Algorithm, which is inspired by the features and life behaviours of microalgae, was proposed for the Multi-Layer Perceptron training process. MLP is a specific form of ANN, and is often used to solve complex problems in the real-world. AAA has three main sections, which are evolutionary process, helical movement, and adaptation. The Artificial Algae Algorithm (AAA) method can effectively achieve the convergence and discovery of the optimal solution through its adaptation and evolution processes. Besides, the AAA helical motion mechanism performs an effective search in the solution space. These processes work together in harmony and provide a balanced exploration and exploitation ability to the AAA algorithm [38]. With such effective processes, AAA has shown superior performance in optimization problems of different dimensions and characteristics, and has inspired it to be used for implementation to new optimization problems [39–42].

This paper is organized as follows. Section 2 describes the methodology by summarizing the MLP and mathematical models of AAA. The AAA-based MLP training model is proposed in Section 3. The general performance tests of the algorithms and the analysis of the test results are carried out in Section 4. In Section 5, the results obtained in this study are evaluated, and suggestions are given for future studies.

2. Methodology

Various ANN models have been proposed in the literature. Convolutional Neural Network (CNN), Feedforward Neural Network (FNN), Kohonen Self-Organizing Network (SOM), Radial Basis Function Network (RBFN), and Recurrent Neural Networks (RNN) are popular neural network models [18,43–46]. Among these, the most commonly used model is FNN. In this model, Neurons, which are processing elements, are designed as layers. The information is transmitted in one direction towards the output. Although there are different neural network models, the learning process is similar. In this study, a multi-layer perceptron has been preferred for the solution of classification problem.

2.1. Multi-layer perceptron

The MLP is a specialized form of supervised ANNs, and consists of the connection of the process elements called neurons with each other in a certain order. The most common connection method is that all neurons in one layer are fully connected to all neurons in the next layer. It has an input layer, at least one hidden layer, and one output layer. It is useful in solving nonlinear problems. The first layer, called the input layer, maps the problem's input to the neural network. The last layer, the output layer, shows the output of the problem. The interior layers other than the input layer and the output layer are called hidden layers. In the literature, the term "node" is used instead of the term "neuron" [10,47].

The critical parameters of MLP are values of weights and biases. Weight and bias values determine the outputs of the network. MLP training is the process of finding the optimum weight and bias values to obtain the desired output from system inputs. In other words, the MLP training process is the creation of a network of relationships of outputs corresponding to inputs.

2.2. Artificial algae algorithm

Artificial Algae Algorithm (AAA) is a meta-heuristic optimization algorithm inspired by microalgae's properties and living behaviour [38]. The coexisting algal cell group is called an algal colony. Every colony represents a possible solution. The population consists of algal colonies, and is expressed as follows:

$$\text{Population} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \cdots & \vdots \\ x_{PN,1} & x_{PN,2} & \cdots & x_{PN,D} \end{bmatrix} \quad (1)$$

$$i^{th}\text{algal colony}\,(X_i) = \left[x_{i,1}, x_{i,2}, \ldots x_{i,D}\right] \quad (2)$$

where x_{ij} is algal cell in the j^{th} dimension of the i^{th} algal colony, D indicates the dimension of algal colonies, and PN indicates the number of algal colony in population.

Each algal colony is formed of a series of algal cells that are considered dimensions of a solution. Algal colony moves together towards the appropriate environment with a nutrient source. The algal colony tries to reach a better position by moving, adapting itself and developing. Optimum solution is obtained when the colony is positioned in the ideal position.

Each algal colony grows according to the nutrients and light it receives during the search process. Initially, the size (greatness-G) of all algal colonies is one. The growth kinetics (μ) of the algal colony is calculated by Eqs. (3) and (4).

$$\mu_i^t = \frac{\mu_{max}^t \times S^t}{K_s^t + S^t} \quad (3)$$

$$G_i^{t+1} = G_i^t + \mu_i^t G_i^t \quad (4)$$

where μ_{max}^t is the maximum specific growth rate at time t, K_s^t is the substrate saturation constant at time t (it is accepted as half of G at time t), and G_i^t is the size of i^{th} algal colony at time t.

In AAA, there are three main sections, which are *evolutionary process*, *helical movement*, and *adaptation*.

2.2.1. Evolutionary process

The algal colony that finds a good solution grows and develops. If the algal colonies do not find a suitable solution, they cannot develop, and die eventually. In the evolutionary process, the algal colonies are sorted by their greatness (G). A randomly selected one dimension of the smallest algal colony is killed, and the same dimension of the largest algal colony is copied instead (Eqs. (5)–(7)).

$$\text{biggest}^t = \max(G_i^t) i = 1, 2, \cdots PN \quad (5)$$

$$\text{smallest}^t = \min(G_i^t) i = 1, 2, \cdots PN \quad (6)$$

$$\text{smallest}_m^t = \text{biggest}_m^t m = 1, 2, \cdots D \quad (7)$$

where biggest^{t}is the biggest algal colony in the population at time t, and smallest^t is the smallest one in the population at time t [38].

2.2.2. Helical movement

At the beginning of each cycle, the energy is calculated in proportion to the greatness of the algal colonies (i.e. normalizing their greatness). In each cycle, the number of times each algal colony moves helically is determined by its energy. The energy of the colony is directly proportional to the nutrient concentration it receives from the environment. During the helical movement, the algal colony which finds a better solution suffers from energy loss

by half ($e/2$) of the initial energy loss parameter. The algal colony, which does not find a better solution, suffers from energy loss as much as the energy loss parameter (e). The frictional force is proportional to the greatness of the algal colony. As the friction surface of the algal colony grows, its movements are slower in the liquid. Therefore, the local search capability of the method increases (i.e. exploitation). The small algal colony has more global search capabilities (i.e. exploration) due to its velocity in the liquid. The friction surface of the colony is calculated by Eq. (8).

$$\tau(X_i) = 2\pi\left(\sqrt[3]{\frac{3G_i}{4\pi}}\right)^2 \tag{8}$$

where $\tau(X_i)$ is the friction surface of the i^{th} algal colony.

The three dimensions (p,r and v) of an algal colony selected by the tournament method for the helical movement is determined. The step size of the movement is calculated by Eqs. (9)–(11).

$$x_{ip}^{t+1} = x_{ip}^{t} + \left(x_{jp}^{t} - x_{ip}^{t}\right)\left(\Delta - \tau^{t}(X_i)\right)p \tag{9}$$

$$x_{ir}^{t+1} = x_{ir}^{t} + \left(x_{jr}^{t} - x_{ir}^{t}\right)\left(\Delta - \tau^{t}(X_i)\right)\cos\alpha \tag{10}$$

$$x_{iv}^{t+1} = x_{iv}^{t} + \left(x_{jv}^{t} - x_{iv}^{t}\right)\left(\Delta - \tau^{t}(X_i)\right)\sin\theta \tag{11}$$

where x_{ip}^{t}, x_{ir}^{t} and x_{iv}^{t} are x, y and z coordinates of the i^{th} algal colony at time t, $\alpha, \theta \in [0, 2\pi]$, $p \in [-1, 1]$, Δ is shear force, and $\tau(X_i)$ is the friction surface area of the i^{th} algal colony.

AAA has three parameters: shear force Δ, energy loss e, and adaptation parameter Ap. In the present study, the shear force value is 2, the energy loss value is 0.3, and the adaptation parameter is 0.2 [38].

2.2.3. Adaption

Adaptation is the step in which the colony, which survives but cannot grow sufficiently, likes itself to be the best colony. Initially, the starvation level of each algal colony is zero. In each helical movement, the starvation level of the algal colony, which cannot find a better solution, increases. The algal colony with the highest starvation level is subjected to adaptation procedure after each helical movement cycle (Eqs. (12) and (13)).

$$\text{starving}^{t} = \max A_i^{t} \tag{12}$$

$$\text{starving}^{t+1} = \text{starving}^{t} + \left(\text{biggest}^{t} - \text{starving}^{t}\right) \times \text{rand} \tag{13}$$

Where A_i^{t} indicates the starvation value of i^{th} algal colony at time t, and starving^{t} indicates the algal colony which has the highest starvation value at time t. The adaptation parameter (Ap) determines whether the adaptation process would be applied in time t or not. Ap is a method-specific parameter that takes values in 0–1 range .

3. Artificial algae algorithm for training Multi-Layer Perceptron

In the literature, heuristic algorithms in MLP training can be used in three different methods. The first method is to use heuristic algorithms to find the weight and bias values of MLP. The second method is using heuristic algorithms to design an architecture suitable for MLP in a particular problem. The third method is the use of a heuristic algorithm to adjust parameters such as learning speed and momentum [48]. The first method aims is to find the appropriate parameters (weights and biases) in a fixed MLP architecture with a heuristic optimization algorithm. Many studies were conducted with this method [12,15,31]. In the second method, the heuristic optimization algorithm aims to obtain the most appropriate MLP architecture for the problem to be solved. As an example of this method, the PSO algorithm was used to design the appropriate network structure to solve two real-world problems [28].

This study aims to train MLP architecture with a single hidden layer by using AAA according to the first method. There are two critical points in the proposed approach.

1. Representation Strategy: Weight and bias values should be represented in appropriate form (algae colonies) into search agents in AAA.
2. Fitness Function: A fitness function that uses the error in the MLP, represented by algae colonies, must be defined.

In the literature, there are three methods for coding the weight and bias values of MLP: *vector coding*, *matrix coding*, and *binary coding* [48,49]. In vector coding, each individual is encoded as a vector. Each vector represents all the weights and biases of an MLP. In matrix coding, each individual is represented as a matrix; and in binary encoding, each individual is represented as binary bit strings. Each of these strategies has advantages and disadvantages over the selected problem [50]. In this study, the most preferred vector coding strategy in the literature was employed. After representing algae colonies in vector form, a fitness function is required to evaluate each of these colonies. The purpose of learning is to train an MLP that can classify validation and test data accurately. The training set has a crucial role in the learning process. This is because each training sample is used to calculate the suitability of the vector represented as the solution. The Mean Square Error (*MSE*) was used to evaluate the accuracy of the network model (Eq. (14)).

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(y - \hat{y}\right)^2 \tag{14}$$

where y indicates actual output, $\hat{y}$ indicates the predicted output, and m indicates the number of samples in the training dataset.

In the proposed model based on AAA, each algae colony represents a candidate solution (i.e. MLP), which is represented as a vector. The candidate solution vector consists of three parts. The first part shows the bias values of the nodes, the second part shows the weight values from the input layer to the hidden layer, and the third part shows the weight values from the hidden layer to the output layer. The length of the vectors is the total weight and bias number in the network.

The vector representation of the MLP with $\boldsymbol{n}$ input nodes, $\boldsymbol{h}$ hidden nodes, and $\boldsymbol{m}$ output nodes is shown in Fig. 1. The blue arrows

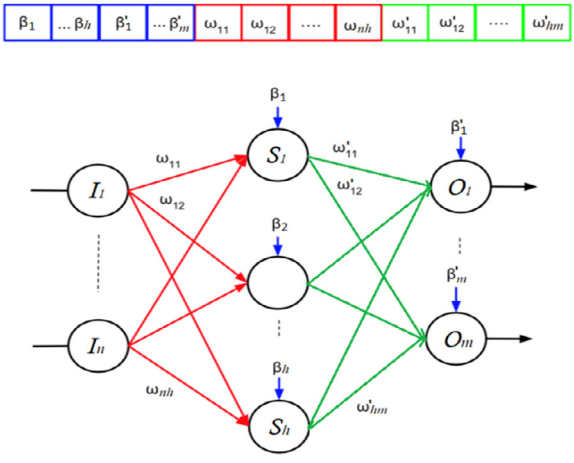

Fig. 1. Assigning a AAA search agent vector to an MLP.

indicate the bias values, the red arrows indicate the weights from the input to the hidden layer, and the green arrows indicate the weights from the hidden layer to the output layer.

The calculation of vector length equation is shown in Eq. (15).

$$VectorLength = (n \times h) + (h \times m) + h + m \quad (15)$$

where n indicates the number of input nodes, h indicates the number of hidden nodes, and m indicates the number of output nodes.

The Mean Square Error (MSE) function was used to determine the fitness values of the generated vectors. The MSE is based on the square of the different calculations between actual and predicted values. The steps of the proposed method are as follows:

1. Initialization: Initially, the algae colonies are randomly created. Each algae colony represents the weight and bias of a network.
2. Assessment of fitness: The performance of the MLP produced is used as a fitness function (i.e. *MSE*). In this step, the parameters of individuals are transferred to MLP, and the performance of MLP is evaluated on the training data. The resulting MSE value is considered as the fitness value of the individual.
3. Update: The positions of algae colonies are updated according to the equations in AAA.
4. Cycle: The steps are repeated until the maximum iteration count reached or specific error rate is obtained. Eventually, the MLP generated from the parameters of the individual with the Minimum MSE value is tested on the test data.

As a result, AAA offers weight and bias values to the MLP and takes the error that occurs in the network as a return. AAA updates weights and biases for iterations to minimize the average error (i.e. MSE) of the network. The flow chart of using AAA for MLP training is given in Fig. 2.

4. Result and discussion

In this section, the performance of the proposed method is evaluated using ten classification datasets from UCI. These datasets are Australian, Blood, Breast cancer, Chess, Diabetes, Ionosphere, Liver, Parkinson's, Tic-tac-toe, and Vertebral [51]. In Table 1, details of datasets are presented with number of classes, number of attributes, training and test sample counts.

For the verification, the results are compared to six swarm-based algorithms and one gradient-based algorithm. These methods are Particle Swarm Optimization (PSO) [21], Grey Wolf Optimizer (GWO) [23], Cuckoo Search Algorithm (CS) [22], Whale Optimization Algorithm (WOA) [24], Bat Optimization Algorithm (BAT) [26], Firefly Algorithm (FFA) [25], and Backpropagation Algorithm (BP) [11]. The results were compared according to statistical tests, the capability of escaping local minimums, the best and average classification accuracy standard deviation values, convergence speeds, and training times.

4.1. Experimental setup

In this study, all datasets were divided for training and testing by 66% and 33%, respectively. All datasets were normalized using [0, 1] min-max normalization.

For all experiments and algorithms, the EvoloPy was used, which is an open-source nature-inspired optimization algorithm toolbox in Python [52]. The default parameters of EvoloPy were used as the initial parameters of the algorithms. All experiments

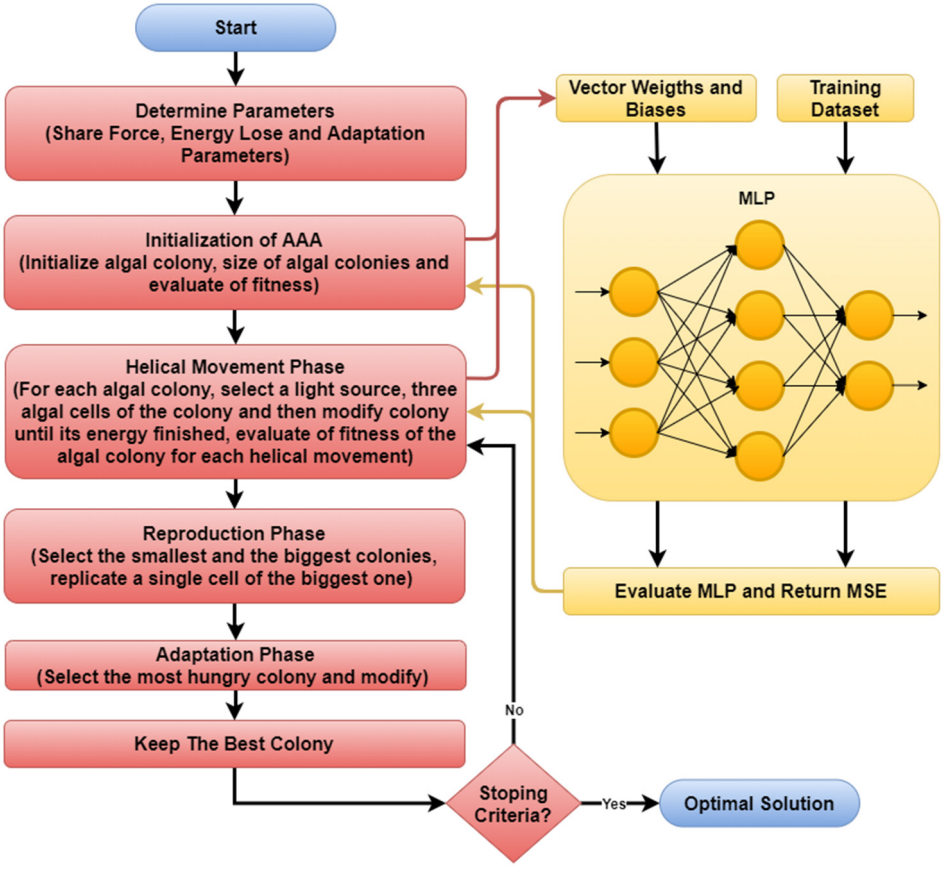

Fig. 2. The flow chart of using AAA for MLP.

Table 1
Classification Datasets.

Datasets	Class	Attributes	Training Samples	Test Samples
Australian	2	14	455	235
Blood	2	4	493	255
Breast cancer	2	8	461	238
Chess	2	36	2109	1087
Diabetes	2	8	506	262
Ionosphere	2	33	231	120
Liver	2	6	79	41
Parkinson's	2	22	128	67
Tic-tac-toe	2	9	632	326
Vertebral	2	6	204	106

were executed for 30 individual runs with random initial parameters. The number of individuals in the population was determined as 40. The algorithms were run for 10,000 fitness calculations. The weights of the network were real numbers in −1, 1 range.

There are different suggestions to set the number of nodes in the hidden layer [15,48,53]. However, there is no certainty about which of the proposed approaches is superior. In this study, the number of nodes in the hidden layer was chosen as $2 \times n + 1$, n representing the number of nodes in the input layer. With this method, the MLP structure for each dataset was built as shown in Table 2.

4.2. Experimental result

The results were interpreted by means of average classification success, best classification success, standard deviation value, convergence curve, and Wilcoxon signed-rank test of 30 independent runs. Table 3 shows the Average Classification Success (ACS), Best Classification Success (BCS), and Standard Deviation Values (SDV). The values in the table are the average result of 30 independent runs of the methods on datasets allocated for testing.

According to the results given in Table 3, the AAA was successful in terms of average classification success in seven of ten datasets (Blood, Breast Cancer, Diabetes, Liver, Parkinson's, Tic-tac-toe, and Vertebral). GWO and BP achieved high success in the classification of the two datasets and one dataset, respectively. GWO, BAT, and BP achieved the highest average classification success in two datasets (Chess and Ionosphere), one dataset (Parkinson's) and one dataset (Australian), respectively. PSO, WOA, BAT, CS, and FFA did not achieve the highest average classification success in any dataset. When evaluated in terms of the best classification success results, AAA achieved the best classification success results in five datasets (Breast Cancer, Diabetes, Ionosphere, Liver, and Parkinson's). PSO, CS, GWO, and BP yielded the best classification success results in four datasets (Breast Cancer, Diabetes, Liver, and Tic-tac-toe), two datasets (Breast Cancer and Vertebral), one dataset (Chess) and two datasets (Australian and Blood), respectively. WOA and BAT were never successful in the best classification success results of any dataset. The standard deviation of the classification success of 30 runs of the AAA seems stable compared to the results of other algorithms. In particular, although the PSO achieved the best classification success in four datasets, it did not achieve the average classification success. AAA even achieved the best average classification success in Vertebral and Blood datasets, although it did not yield the best classification success. These results show that the AAA achieved the best average classification success in most datasets, and is therefore a stable algorithm.

Table 2
The Parameters of MLP for datasets.

Datasets	Network Structure	Vector Length
Australian	14-29-1	465
Blood	4-9-1	55
Breast Cancer	8-17-1	171
Chess	36-73-1	2775
Diabetes	8-17-1	171
Ionosphere	33-67-1	2346
Liver	6-13-1	105
Parkinson's	22-45-1	1081
Tic-tac-toe	9-19-1	210
Vertebral	6-13-1	105

According to the results in Table 3, the average classification success of algorithms is very close to each other. Whether the differences in the classification success are significant can be decided by non-parametric statistical tests. For this purpose, the results in Table 3 were calculated by Wilcoxon signed-rank test. The purpose of this test is to determine whether there is a significant difference between the results obtained from different classification algorithms in the same datasets. The test is carried out between 30 independent results of two algorithms. In this test, H0 and H1 hypotheses are used. The H0 (Null hypothesis) hypothesis is "There is no significant difference between the results of the first algorithm and the second algorithm" and the H1 (Alternative hypothesis) hypothesis is the opposite of H0. At the end of the test, the H0 hypothesis is accepted or rejected. The degree to which the hypothesis is rejected is determined by the level of significance. In this study, the level of significance was set at 0.05.

The calculated p-Value represents the probability that the Null Hypothesis is rejected. A $\boldsymbol{p}$ value of less than 0.05 indicates a significant difference between the two algorithms; and $\boldsymbol{p}$ value higher than 0.05 indicates that the difference between the two algorithms is insignificant. The critical point in the results tables of the Wilcoxon signed-rank test is W value. If W value is 1, the AAA algorithm is successful, and if it is 2, the other algorithm is successful. In cases where the W value is 0, there is no statistical difference between the algorithms. R + is the sum of the ranks corresponding to the positive differences between the first algorithm and the second algorithm, and R − is the sum of the ranks corresponding to the negative differences between the first algorithm and the second algorithm. T is the smaller one of the sums (T = min (R+, R −)). Table 4 shows the result of the Wilcoxon signed-ranks test between AAA and other algorithms. In the last row of each statistical test results in Table 4, the total number of W values are given in the structure (1/0/2). In this line, it can be seen how many datasets the AAA (marked with '1') and the compared algorithm (marked with '2') are more successful in total. If there is no statistically significant difference between the two compared algorithms, it is marked with '0'.

When Table 4 is analysed, AAA is statistically more successful than other algorithms in six datasets (Blood, Breast Cancer, Diabetes, Liver, Tic-tac-toe, and Vertebral). BP is more successful than

Table 3
Average experimental results of 30 runs.

Data		AAA	PSO	WOA	BAT	CS	GWO	FFA	BP
Australian	ACS	0.848	0.817	0.794	0.837	0.839	0.838	0.827	**0.857**
	SDV	0.006	0.021	0.049	0.005	0.013	0.005	0.010	0.016
	BCS	0.860	0.851	0.834	0.843	0.864	0.843	0.838	**0.877**
Blood	ACS	**0.748**	0.743	0.741	0.741	0.745	0.740	0.741	0.663
	SDV	0.003	0.004	0.004	0.003	0.004	0.003	0.004	0.205
	BCS	0.757	0.749	0.745	0.745	0.753	0.741	0.745	**0.784**
Breast Cancer	ACS	**0.981**	0.968	0.965	0.971	0.970	0.973	0.966	0.892
	SDV	0.003	0.008	0.006	0.004	0.008	0.003	0.005	0.101
	BCS	**0.987**	**0.987**	0.971	0.975	**0.987**	0.979	0.971	0.970
Chess	ACS	0.710	0.678	0.662	0.736	0.715	**0.939**	0.697	0.690
	SDV	0.018	0.037	0.089	0.094	0.024	0.003	0.020	0.047
	BCS	0.767	0.744	0.822	0.828	0.752	**0.944**	0.723	0.725
Diabetes	ACS	**0.753**	0.747	0.692	0.746	0.739	0.747	0.742	0.596
	SDV	0.007	0.016	0.026	0.006	0.014	0.003	0.006	0.140
	BCS	**0.771**	**0.771**	0.718	0.752	0.767	0.752	0.752	0.690
Ionosphere	ACS	0.867	0.791	0.599	0.841	0.809	**0.897**	0.819	0.751
	SDV	0.022	0.046	0.046	0.054	0.040	0.012	0.029	0.038
	BCS	**0.925**	0.875	0.658	0.892	0.892	0.908	0.850	0.783
Liver	ACS	**0.760**	0.750	0.616	0.736	0.714	0.742	0.726	0.552
	SDV	0.011	0.015	0.027	0.021	0.026	0.017	0.011	0.071
	BCS	**0.780**	**0.780**	0.653	0.754	0.763	0.754		0.737
								0.746	
Parkinson's	ACS	**0.860**	0.790	0.721	**0.860**	0.812	0.858	0.804	0.778
	SDV	0.017	0.045	0.059	0.009	0.036	0.009	0.026	0.148
	BCS	**0.896**	0.881	0.791	0.866	0.866	0.866	0.836	0.865
Tic-tac-toe	ACS	**0.713**	0.688	0.646	0.692	0.675	0.697	0.693	0.583
	SDV	0.013	0.027	0.018	0.009	0.022	0.014	0.018	0.046
	BCS	0.739	**0.755**	0.663	0.702	0.718	0.712	0.715	0.625
Vertebral	ACS	**0.881**	0.869	0.740	0.854	0.835	0.869	0.855	0.717
	SDV	0.006	0.016	0.026	0.019	0.035	0.010	0.020	0.146
	BCS	0.896	0.906	0.783	0.877	**0.906**	0.877	0.868	0.811

Table 4
Wilcoxon Signed Rank test results of AAA and the other algorithm.

	AAA – PSO			AAA – WOA			AAA – BAT			AAA – CS		
	p-Value	T	W	p-Value	T	W	p-Value	T	W	p-Value	T	W
Australian	3.1779E-06	459	1	1.7235E-06	465	1	1.7718E-05	344	1	1.3772E-03	388	1
Blood	1.0671E-05	446	1	8.9272E-06	325	1	9.8221E-06	325	1	1.6391E-02	349	1
Breast Cancer	4.6640E-06	455	1	1.3475E-06	465	1	1.2293E-06	465	1	1.0027E-05	447	1
Chess	8.9415E-04	394	1	2.1816E-02	344	1	7.3520E-02	145.5	0	2.1336E-01	172	0
Diabetes	4.1669E-04	270	1	1.7170E-06	465	1	1.1426E-03	263.5	1	3.3088E-04	407	1
Ionosphere	3.1485E-06	433	1	1.7019E-06	465	1	1.8348E-02	306.5	1	1.6352E-05	442	1
Liver	1.5609E-02	350	1	1.7019E-06	465	1	1.7148E-05	300	1	1.9018E-06	464	1
Parkinson's	2.8526E-06	460	1	1.6437E-06	465	1	3.4479E-01	168.5	0	1.6126E-05	442	1
Tic-tac-toe	1.8902E-04	414	1	1.7181E-06	465	1	2.5339E-06	435	1	3.8799E-06	457	1
Vertebral	7.5079E-04	396	1	1.6699E-06	465	1	1.4818E-06	465	1	4.5941E-06	455	1
1/0/2	**10/0/0**			**10/0/0**			**8/2/0**			**9/1/0**		
	AAA – GWO			AAA – FFA			AAA – BP					
	p-Value	T	W	p-Value	T	W	p-Value	T	W			
Australian	8.2918E-06	398	1	2.4813E-06	435	1	1.9608E-03	82	2			
Blood	1.1897E-06	465	1	2.9261E-06	406	1	1.0897E-06	465	1			
Breast Cancer	1.9182E-06	435	1	1.5224E-06	465	1	1.7279E-06	465	1			
Chess	1.7300E-06	0	2	8.9682E-02	277.5	0	1.0639E-01	311	0			
Diabetes	4.1669E-04	293	1	1.8040E-05	344	1	1.7257E-06	465	1			
Ionosphere	1.3820E-04	36	2	2.7441E-06	460	1	1.7289E-06	465	1			
Liver	1.6262E-05	300	1	1.5760E-06	465	1	1.4656E-06	465	1			
Parkinson's	2.1344E-01	178	0	3.6072E-06	406	1	2.0477E-02	345	1			
Tic-tac-toe	8.0817E-05	424	1	1.6228E-04	368.5	1	1.7344E-06	465	1			
Vertebral	1.4101E-05	276	1	1.4346E-06	465	1	1.7344E-06	465	1			
1/0/2	**7/1/2**			**9/1/0**			**8/1/1**					

AAA in the Australian dataset, and the most successful algorithm in Chess and Ionosphere datasets is GWO. In the Parkinson's dataset, while AAA and BAT have the best average classification success, there is no statistically significant difference with the results of GWO. In the Chess dataset, the GWO has obvious classification success compared to other algorithms. On the other hand, when the results of AAA in the Chess dataset are compared with BAT, CS, FFA, and BP, there is no statistically significant difference between

Table 5
The results of the Average Ranking and the Friedman test.

Data	AAA	PSO	WOA	BAT	CS	GWO	FFA	BP
Australian	2	7	8	5	3	4	6	1
Blood	1	3	5	5	2	7	5	8
Breast Cancer	1	5	7	3	4	2	6	8
Chess	4	7	8	2	3	1	5	6
Diabetes	1	2.5	7	4	6	2.5	5	8
Ionosphere	2	6	8	3	5	1	4	7
Liver	1	2	7	4	6	3	5	8
Parkinson's	1.5	6	8	1.5	4	3	5	7
Tic-tac-toe	1	5	7	4	6	2	3	8
Vertebral	1	2.5	7	5	6	2.5	4	8
Average Ranking Results								
Mean Rank	1.55	4.60	7.20	3.65	4.50	2.80	4.80	6.90
Final Rank	1	4	8	3	5	2	6	7
Friedman Test Results								
Mean Rank	1.50	4.50	7.20	3.60	4.50	2.90	4.90	6.90
Final Rank	1	4.5	8	3	4.5	2	6	7
p-Value	3.9259e−07							

the results. As a result, the best values of AAA in six datasets are statistically significant. In two datasets, the best results are shared with different algorithms.

The Friedman test is a nonparametric statistical test used in the analysis of experimental results. The Friedman test evaluates the results of more than two classifiers [54]. Another method used to compare more than two algorithms is the Average Ranking method. Friedman test and Average Ranking are also applied for evaluating the algorithms with a significance level of 0.05. The results of the Friedman test and the Average ranking of the algorithms are given in Table 5.

As shown in Table 5, the p-Value obtained from the Friedman test is less than 0.05. Therefore, there is a significant difference between the results obtained. In addition, considering both the Average ranking and Friedman test ranking values, it is clear that AAA is more successful than other algorithms. In addition, the Average Ranking and the Friedman test final rank results are largely similar.

The average convergence curve of the 30 independent runs of six datasets is shown in Fig. 3. In the six datasets selected for the average convergence curve, AAA, GWO, and BP are more successful than the other algorithm in the three (Breast Cancer, Liver, and Tic-tac-toe), two (Chess and Ionosphere), and one (Australian) dataset (s), respectively. In four datasets (Blood, Diabetes, Parkinson's, and Vertebral), where the average convergence curves are not given in Fig. 3, AAA achieved successful results. These convergence curves are similar to the convergence curves of the other datasets where AAA was successful.

Convergence curves are an essential indicator in terms of examining the behaviour of the algorithm against the problems. When the convergence curves are analysed, the WOA and BP usually reach a local minimum due to early convergence. For this reason, these two algorithms cannot show any improvement in terms of reducing errors in later parts of the iterations. Although BP achieved the best classification success in the Australian dataset, its development towards the end of iteration decreased considerably. The GWO continued its development until the end of the iterations better than other algorithms, without reaching any local minimum in Chess and Ionosphere datasets. PSO, BAT, CS, FFA, and AAA algorithms improved by reducing the error to the end of iteration without reaching any local minimum. Therefore, it can be foreseen that they can show more improvements when the number of iterations is increased. When convergence curves are generally evaluated in terms of AAA, it can search for solutions without reaching local minimums thanks to its exploration capability. Additionally, with its exploitation ability, it can develop the solution in the later iterations.

4.3. Discussion and analysis of the result

In this experimental study, the MLP training method was compared with AAA with seven successful methods mentioned in the literature. The results obtained were analysed in detail from different perspectives. Firstly, the average classification success, the best classification success, and the standard deviation value of the algorithms achieved in 30 independent runs were examined and interpreted. AAA achieved the highest average classification success in six of ten datasets, and the highest average classification success in six of ten datasets. Also, the low standard deviation indicated that AAA is a stable algorithm for training MLP. The low standard deviation of the AAA in 30 independent runs indicates that it runs independently of the initial parameters and avoids stuck in minimum values.

Secondly, the average classification success results obtained by the compared algorithms were quite close to each other. The Wilcoxon signed-rank statistical test with a 95% level of significance determined whether there was a significant difference between the results. AAA was statistically more successful than all other algorithms in six of the seven datasets, where it achieved the highest average classification success. Average ranking and the Friedman test were used to analyse the average classification success results obtained by all algorithms on all datasets. The results of both ranking methods showed that AAA is more successful than other algorithms. On the other hand, when the results were evaluated in terms of the vector length formed depending on the number of attributes of the datasets, it did not achieve the best results in datasets with a high number of attributes such as Chess and Ionosphere. The main reason for this is that AAA changes only three dimensions of the individual due to the helical movement in each iteration. For this reason, it causes a lack of exploration in high dimensional datasets.

Finally, convergence curves show how algorithms reduce MSE values in datasets throughout iterations. In a small number of iterations, algorithms reducing the MSE value is efficient. Algorithms with exploitation capability, converges faster than others. However, faster convergence characteristics increases the probability of stuck into local minima. Therefore, early convergence algorithms may not reduce the error to the desired value.

In the case of algorithms with high exploration capabilities, convergence is partially slower. However, it is less likely to find local minimums and more global search capability. The convergence curves show that AAA improves the best solution throughout the iteration. In this study, the maximum iteration number was 250. Considering that AAA continuously reduces its error during iterations, the increase in the number of iterations will increase the success of AAA.

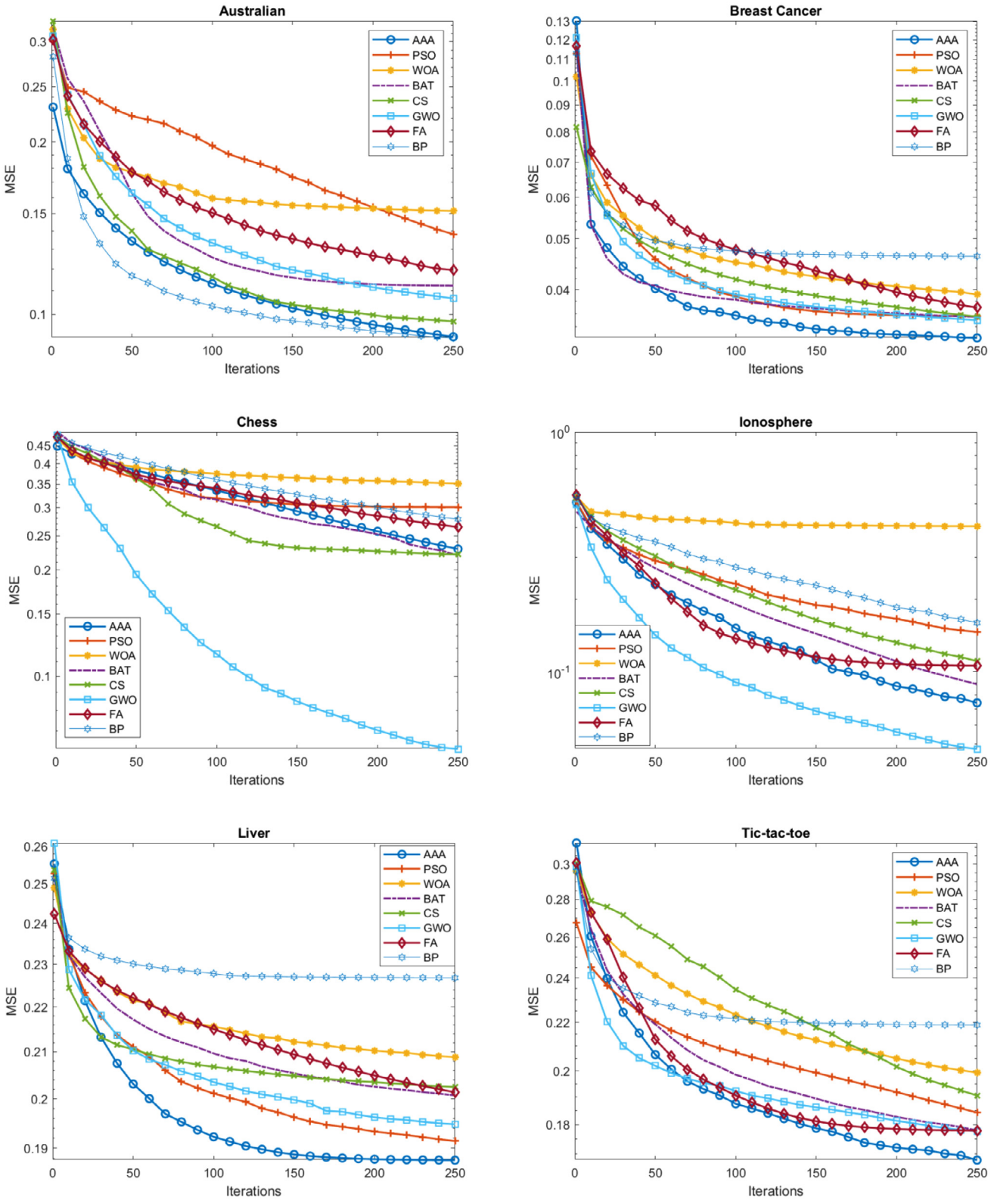

Fig. 3. MSE convergence curves of the algorithms (Australian, Breast Cancer, Chess, Ionosphere, Liver, Tic-tac-toe).

5. Conclusion

In the literature, many mathematical and heuristic strategies have been proposed for training ANN. However, existing algorithms are still not able to eliminate the tendency of the local minimum. In this study, the recently proposed artificial algae algorithm was proposed for training ANN. The success of the proposed method was confirmed using ten public datasets. The results were compared with six different population-based metaheuristic optimization algorithms and backpropagation algorithm.

The standard deviation value of AAA in 30 independent runs was low, and the average classification success was high, indicating that it works independently from the initial parameters and avoids being stuck in local minimums. The consistency of the convergence curve of AAA in the datasets increases the confidence in the proposed method. The results obtained from the experimental study were analysed by Wilcoxon signed-rank, Friedman test, and Average Ranking Methods. As a result of the analysis, AAA was found to be more successful than other algorithms. In this study, AAA was proven to be a reliable alternative to ANN training.

In future studies, the AAA algorithm can be used to determine the number of hidden layers, and the number of nodes in the hidden layer. AAA can be applied in the optimization of convolutional neural network parameters, and can be used to find the number of layers of the convolutional neural networks, and to optimize the kernel values. Besides, AAA can be strengthened by modifications that can be effective in high-dimensional optimization problems, and its contribution to the analysis of high-dimensional datasets can be increased.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] R. Linggard, D. Myers, C. Nightingale, Neural networks for vision, speech and natural language, Vol. 1, Springer Science & Business Media, 2012.
[2] M. Rezaeianzadeh et al., Flood flow forecasting using ANN, ANFIS and regression models, Neural Comput. Appl. 25 (1) (2014) 25–37.
[3] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Networks 61 (2015) 85–117.
[4] J. Khan et al., Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, Nat. Med. 7 (6) (2001) 673.
[5] M. Farzaneh-Gord et al., Precise prediction of biogas thermodynamic properties by using ANN algorithm, Renew. Energy 147 (2020) 179–191.
[6] B. Maleki et al., Moisture estimation in cabinet dryers with thin-layer relationships using a genetic algorithm and neural network, Mathematics 7 (11) (2019) 1042.
[7] S. Vahora, N. Chauhan, Deep neural network model for group activity recognition using contextual relationship, Eng. Sci. Technol. Int. J. 22 (1) (2019) 47–54.
[8] S. Uslu, M.B. Celik, Prediction of engine emissions and performance with artificial neural networks in a single cylinder diesel engine using diethyl ether, Eng. Sci. Technol. Int. J. 21 (6) (2018) 1194–1201.
[9] M. Yilmaz, E. Kayabasi, M. Akbaba, Determination of the effects of operating conditions on the output power of the inverter and the power quality using an artificial neural network, Eng. Sci. Technol. Int. J. 22 (4) (2019) 1068–1076.
[10] I. Basheer, M. Hajmeer, Artificial neural networks: fundamentals, computing, design, and application, J. Microbiol. Methods 43 (1) (2000) 3–31.
[11] Li, J., et al., Brief introduction of back propagation (BP) neural network algorithm and its improvement, in Advances in computer science and information engineering. 2012, Springer. p. 553-558.
[12] D. Karaboga, B. Akay, C. Ozturk, Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks, MDAI 7 (2007) 318–319.
[13] A.S. Rakitianskaia, A.P. Engelbrecht, Training feedforward neural networks with dynamic particle swarm optimisation, Swarm Intell. 6 (3) (2012) 233–270.
[14] L. Wang, Y. Zeng, T. Chen, Back propagation neural network with adaptive differential evolution algorithm for time series forecasting, Expert Syst. Appl. 42 (2) (2015) 855–863.
[15] I. Aljarah, H. Faris, S. Mirjalili, Optimizing connection weights in neural networks using the whale optimization algorithm, Soft. Comput. 22 (1) (2018) 1–15.
[16] X.-S. Yang, Nature-inspired Metaheuristic Algorithms, Luniver press, 2010.
[17] G.-G. Wang et al., Hybrid krill herd algorithm with differential evolution for global numerical optimization, Neural Comput. Appl. 25 (2) (2014) 297–308.
[18] R. Reed, R.J. Marks, Neural smithing: supervised learning in feedforward artificial neural networks. 1999: Mit Press.
[19] N. Dey, A. Ashour, and S. Bhattacharyya, Applied Nature-Inspired Computing: Algorithms and Case Studies. 2020: Springer.
[20] N. Dey, Advancements in Applied Metaheuristic Computing, IGI Global, 2017.
[21] J. Kennedy, Particle swarm optimization, in Encyclopedia of machine learning. 2011, Springer. p. 760-766.
[22] A.H. Gandomi, X.-S. Yang, A.H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (1) (2013) 17–35.
[23] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[24] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[25] X.-S. Yang, Firefly algorithms for multimodal optimization. in International symposium on stochastic algorithms. 2009. Springer.
[26] X.S. Yang, A.H. Gandomi, Bat,, algorithm: a novel approach for global engineering optimization, Eng. Comput. (2012).
[27] M. Meissner, M. Schmuker, G. Schneider, Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training, BMC Bioinf. 7 (1) (2006) 125.
[28] J. Yu, S. Wang, L. Xi, Evolving artificial neural networks using an improved PSO and DPSO, Neurocomputing 71 (4–6) (2008) 1054–1060.
[29] E. Valian, S. Mohanna, S. Tavakoli, Improved cuckoo search algorithm for feedforward neural network training, Int. J. Artif. Intell. Appl. 2 (3) (2011) 36–43.
[30] N.S. Jaddi, S. Abdullah, A.R. Hamdan, Multi-population cooperative bat algorithm-based optimization of artificial neural network model, Inf. Sci. 294 (2015) 628–644.
[31] S. Mirjalili, How effective is the Grey Wolf optimizer in training multi-layer perceptrons, Appl. Intell. 43 (1) (2015) 150–161.
[32] S. Nandy, P.P. Sarkar, and A. Das, Analysis of a nature inspired firefly algorithm based back-propagation neural network training. arXiv preprint arXiv:1206.5360, 2012.
[33] A.A. Heidari et al., An efficient hybrid multilayer perceptron neural network with grasshopper optimization, Soft. Comput. 23 (17) (2019) 7941–7958.
[34] J. Xu, F. Yan, Hybrid Nelder-Mead algorithm and dragonfly algorithm for function optimization and the training of a multilayer perceptron, Arab. J. Sci. Eng. 44 (4) (2019) 3473–3487.
[35] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[36] Y.-C. Ho, D.L. Pepyne, Simple explanation of the no-free-lunch theorem and its implications, J. Optim. Theory Appl. 115 (3) (2002) 549–570.
[37] R. Tang, S. Fong, N. Dey, Metaheuristics and chaos theory, Chaos Theory (2018) 182–196.
[38] S.A. Uymaz, G. Tezel, E. Yel, Artificial algae algorithm (AAA) for nonlinear global optimization, Appl. Soft Comput. 31 (2015) 153–171.
[39] S. Korkmaz, A. Babalik, M.S. Kiran, An artificial algae algorithm for solving binary optimization problems, Int. J. Mach. Learn. Cybern. 9 (7) (2018) 1233–1247.
[40] M.A. Tawhid, V. Savsani, A novel multi-objective optimization algorithm based on artificial algae for multi-objective engineering design problems, Appl. Intel. (2018) 1–20.
[41] X. Zhang et al., Binary artificial algae algorithm for multidimensional knapsack problems, Appl. Soft Comput. 43 (2016) 583–595.
[42] S.A. Uymaz, G. Tezel, E. Yel, Artificial algae algorithm with multi-light source for numerical optimization and applications, Biosystems 138 (2015) 25–38.
[43] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inform. Process. Syst. (2012).
[44] T. Kohonen, The self-organizing map, Proc. IEEE 78 (9) (1990) 1464–1480.
[45] J. Park, I.W. Sandberg, Approximation and radial-basis-function networks, Neural Comput. 5 (2) (1993) 305–316.
[46] Dorffner, G. Neural networks for time series processing. in Neural network world. 1996. Citeseer.
[47] G. Panchal et al., Behaviour analysis of multilayer perceptronswith multiple hidden neurons and hidden layers, Int. J. Comput. Theory Eng. 3 (2) (2011) 332.
[48] S. Mirjalili, S.M. Mirjalili, A. Lewis, Let a biogeography-based optimizer train your multi-layer perceptron, Inf. Sci. 269 (2014) 188–209.
[49] J.-R. Zhang et al., A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training, Appl. Math. Comput. 185 (2) (2007) 1026–1037.
[50] S. Mirjalili, S.Z.M. Hashim, H.M. Sardroudi, Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm, Appl. Math. Comput. 218 (22) (2012) 11125–11137.
[51] A. Asuncion, D. Newman, UCI machine learning repository. 2007.
[52] Faris, H., et al. EvoloPy: An Open-source Nature-inspired Optimization Framework in Python. in IJCCI (ECTA). 2016.
[53] A.S.I. Wdaa, A. Sttar, Differential Evolution for Neural Networks Learning Enhancement, Universiti Teknologi Malaysia, 2008.
[54] J. Derrac et al., A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.