International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Training of the Artificial Neural Networks using Crow Search Algorithm

Feyza Erdoğan*[1], Şaban Gülcü[2]

Accepted : 06/05/2021 **Published: 31/07/2021**

Abstract: Artificial Neural Networks are methods frequently used in problem-solving today. In the past, it has been used in many areas, such as classification, pattern recognition and image processing. The most important and demanding part of Artificial Neural Networks is the training process of the network. The main challenge in network training is the process of determining the optimum connection weights and bias values for the network. In the literature, many algorithms have been proposed for training Artificial Neural Networks. This article proposed a new hybrid algorithm called CSA-MLP, for training Artificial Neural Networks using the Crow Search Algorithm. Crow Search Algorithm is a population-based meta-heuristic optimization algorithm, inspired by the behavior of crows to store their surplus nutrients and take them back from the storage area when needed. Crow Search Algorithm has been proposed to solve different optimization problems in terms of its simplicity with two different adjustable parameters (flight length and awareness probability), obtaining an effective convergence rate in a short time and having a faster technique compared to algorithms frequently used in engineering problems with different constraints and functions. In the experiments, five classification datasets (xor, balloon, iris, breast cancer, heart) were used. The CSA-MLP algorithm was compared with the SMS-MLP in terms of the mean squared error, classification rate, the statistical metrics (sensitivity, specificity, precision, f1-score) and the convergence graph. Furthermore, the proposed CSA-MLP algorithm was compared with seven algorithms in literature in terms of best classification accuracy. The experimental results show that the Crow Search Algorithm is a reliable approach in training Multi-Layer Perceptron. CSA-MLP achieved better results than SMS-MLP and other algorithms.

Keywords: *Artificial neural networks, training of artificial neural networks, training of multi-layer perceptron, crow search algorithm, optimization, feed-forward artificial neural networks.*

1. Introduction

Artificial Neural Network (ANN) is one of the popular topics in the fields of artificial intelligence and machine learning designed, inspired by the working mechanism of the biological nervous system in the human brain in the 1940s [1]. ANN is basically a technology developed entirely by sampling the human brain [2]. It has been used in many fields as a result of studies that gained speed in the 1980s and its success has been proven [3]. ANN is successfully used in areas such as classification, system modelling, generalization, pattern recognition, prediction, control and optimization. An effective problem-solving strategy for complex and difficult-to-solve problems, which is among the most important features of ANN, is one of the methods frequently used in solving many problems, especially engineering studies [4].

ANN is formed as a result of connecting many artificial process elements (neurons) in its layers to each other with different effect levels. It obtains information from the given examples related to a problem and then makes a decision by using the information it acquired when encounter the samples of the problem that it has never seen [4].

The most important feature of ANN is its ability to infer for different situations with the experiences acquired by learning the information given to it [5]. ANN consists of two stages called training (learning) and testing. First, ANN is trained with training data, then the network is tested with test data to evaluate the classification performance of the trained ANN [6]. The most important and demanding part of ANN is the training process of the network. The ANN's training process is the process of finding values that match the weights in the network architecture and it is a very difficult optimization problem. In order to solve this problem, scientists have proposed intuitive and meta-intuitive algorithms, which are often used in the solution of optimization problems. These algorithms were also used in the training process of ANN.

Recently, there are many population-based metaheuristic optimization algorithms designed, inspired by the behavior of living things in nature. Population-based algorithms mathematically formulate the behavior of creatures living in nature, such as birds, fish, insects, and try to achieve an acceptable solution to complex optimization problems within the framework of certain constraints in an acceptable time. These algorithms are frequently used in problem-solving and have proven success.

In the literature, many different metaheuristic optimization algorithms have been proposed to ensure the training of ANN with optimum parameters. Yamany et al [7] proposed the Moth Flame Optimization for the training of Feed-Forward Multilayer ANN. The proposed algorithm has been used to generate the weighting and bias values that ensure minimum error and high classification

[1] *Computer Eng. Dept., Necmettin Erbakan University, Konya, Turkey ORCID ID: 0000-0002-9750-0495*
[2] *Computer Eng. Dept., Necmettin Erbakan University, Konya, Turkey ORCID ID: 0000-0001-7714-8861*
** Corresponding Author Email: feyzaaerdogann@gmail.com*

success. Five classification data sets were used to evaluate the performance of the proposed method. The success of the proposed method has been compared with Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Evolution Strategy (ES). Experimental results prove that the MFO algorithm solves the problem of entrapment in local optima and achieves high accuracy. Jaddi, Abdullah and Hamdan [8] proposed the Bat Optimization Algorithm to generate the weight and bias values of ANN that would ensure minimum error and high classification success. Six classifications and two time series Benchmark data sets were used to test the performance of the proposed approach in terms of classification and predictive accuracy. Statistical tests show that the proposed method produces the best results when compared with the latest methods in the literature. The proposed method has been applied to a real-world problem to predict future values of precipitation data, and the results have proven the method's success.

CSA has been proposed to solve different optimization problems in terms of its simplicity with two different adjustable parameters (flight length and awareness probability), obtaining an effective convergence rate in a short time and having a faster technique compared to algorithms frequently used in engineering problems, with different constraints and functions. Mohammadi and Abdi [9] proposed a new evolutionary optimization algorithm called the modified crow search algorithm (MCSA) to solve the economic load distribution (ELD) problem. The proposed MCSA improves the search capability of crows in the original CSA, offering a new way to select the target, to be pursued by the crow. It was tested on five different well-known test systems, to determine the success of MCSA's ELD problem. The results were compared with other methods in terms of solution quality, solution robustness and calculation time, and the proposed method has been proven to have more superior performance than other techniques. Gupta et al [10] recommend a new, improved and optimized version of the crow search algorithm (OCSA) to improve the diagnosis of Parkinson's disease. It was emphasized that the proposed OCSA can be used to predict Parkinson's disease with 100% accuracy and can help the individual to have the appropriate treatment at an early stage. The performance of OCSA has been tested on 20 data sets and the results were compared with the original chaotic crow search algorithm (CCSA). Experimental results reveal that the proposed model finds an optimal subset of features, maximizes accuracy, minimizes a selected set of features, and is more stable.

In this study, Crow Search Algorithm is recommended in the process of updating the network with the most optimum weights in order to increase the classification success of ANN. Crow Search Algorithm is a population-based metaheuristic optimization algorithm, inspired by the behavior of crows storing surplus nutrients and consuming them by retrieving them from the storage area when needed.

The following parts of this study are organized as follows: Section 2, consists of three parts. In the first part, basic information about ANN is given. The second part includes the development process of the CSA algorithm, the mathematical formulas in the algorithm and the pseudo-code of the algorithm. In the third part, information is given about the training process and modeling of ANN with CSA. In Section 3, information is given about the data sets to be used in the training and testing phase of ANN, explanations about the platform, where the program will be coded and run, and the performance determination criteria used to determine the success of the proposed method. In addition, in this section, the performance comparison of the CSA algorithm, proposed for the training of ANN, and some of the algorithms recommended for training of ANN in the literature is made. In Section 4, the results obtained as a result of the experimental studies carried out with the proposed model, are evaluated and suggestions for future studies are provided.

2. Material and Method

2.1. Artificial Neural Networks (ANN) and Multi-Layer Perceptrons (MLP)

According to the structure of the connections between neurons, ANN is examined in two classes, as Feed Forward and Feedback. Feed-Forward Networks are network structures in which data submitted to the network is forwarded from input units to output units. There is no feedback in this network structure [4].

One widely used type of feed-forward network is Multilayer Perceptrons [11]. The MLP consists of three basic layers. These layers are Input Layer, Intermediate (Hidden) Layer and Output Layer, respectively. Layers are located in parallel in the network structure. There are several neurons in the layers, known as processing elements that are involved in the processing of information. The neurons in each layer are unidirectionally connected from the input layer to the output layer. The structure that allows neurons to connect to each other is called connection weights. Connection weights are generally determined from real values between -1 and 1. The most important parameters of the MLP are connection weights and bias values. Connection weights and bias values shape the final result of the outputs produced by the MLP. Fig. 1 shows the MLP structure with only one hidden layer.

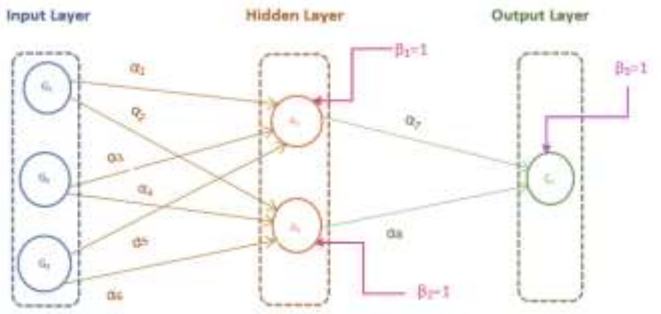


Fig. 1 MLP Structure with a single hidden layer

The MLP aims to obtain the expected output data of the input data of the problem transmitted to it over the input layer and present it from the output layer. In order to achieve this goal, the network should be trained using the input and output data of the data set of the problem and the connection weights should be updated with optimum values. The training process continues until the network termination criteria are met [1]. Using the input data of the MLP, obtaining the output data is realized by using the following equations [12]:

Step 1: The weighted sums of input data are calculated by Equation (1).

$$p_n = \sum_{m=1}^{k} (w_{mn} \times v_m) + \theta_n \tag{1}$$

where k is the number of neurons in the input layer, w_{mn} is the weight of the connection between the mth neuron in the input layer and the nth neuron in the hidden layer, θ_n is the bias value of the nth hidden neuron, v_m is the mth input value.

Step 2: The output value of each neuron in the hidden layer by Equation (2). An activation function is used here. Care should be taken that the activation function is easily derivative and not linear.

Today, the most preferred activation function for the MLP model is the Sigmoid Activation Function [12].

$$P_n = Sigmoid(p_n) = \frac{1}{(1+e^{-p_n})} \tag{2}$$

Step 3: Steps 1 and 2 are performed for the output layer. Thus, the output value produced by the network against inputs is calculated by Equations (3) and (4).

$$o_t = \sum_{n=1}^{h}(w_{nt} \times P_n) + \beta_t \tag{3}$$

$$O_t = Sigmoid(o_t) = \frac{1}{(1+e^{-o_t})} \tag{4}$$

where h is the number of neurons in the hidden layer, w_{nt} is the connection weight between the *n*th neuron in the hidden layer and the *t*th neuron in the output layer, β_t is the bias value of the *t*th output layer neuron.

2.2. Crow Search Algorithm (CSA)

Crows are bird species accepted among the smartest animals in the world that can easily memorize the faces they see, have the ability to use tools when they see it, have the ability to communicate in complex ways, and can store and use food according to need [13],[14]. Crows are in search of better food sources, and they follow each other at all the time. From an optimization point of view, crows are in search of the best food source in the searching area. Each location of the environment corresponds to an acceptable solution, and the quality of the solution is evaluated by the fitness function. When evaluated in this respect, it is concluded that, the crow populations have similar characteristics with the optimization process. As a result, Crow Search Algorithm was introduced to the literature by Alireza Askarzadeh in 2016. CSA is a population-based metaheuristic optimization algorithm [15].

Crow Search Algorithm is inspired by the behavior of crows to store their surplus nutrients and to take them back from the storage area when needed [15]. As per their philosophy of life, crows follow other crows and learn about the areas where they store their food and steal these foods when the owner goes away. Therefore, each crow takes various measures to prevent theft and tries to determine the safest area by using their memories [16]. In short, crows are creatures that live in herd and keep food and memorize these hiding places in order to survive. They tend to steal food by constantly following each other and try to prevent other crows from stealing their food with their memories [15].

While formulating the CSA, 4 basic information was taken into account [15]. Those are;

- Crows usually live-in flocks.
- They memorize the places where they keep their food.
- They follow each other to steal food.
- They protect food stores from being robbed using an awareness possibility.

It is thought that crows store their food in a *d*-dimensional search environment. In the case where there are *n* crows in the *d* dimensional search space, the position of the *i*th crow in the *t*th iteration is represented by the vector in Equation (5).

$$K^{i,t} = (i = 1,2,\dots,n; t = 1,2,\dots,t_{max})$$

$$K^{i,t} = [K_1^{i,t}, K_2^{i,t}, \dots, K_d^{i,t}] \tag{5}$$

where t_{max} is the maximum number of iterations.

All crows in the population have a memory that keeps track of where they store their food. For *i*th crow, the memory is represented as $B^{i,t}$ and holds the best storage location information obtained up to the current iteration. Thus, all crows keep memorizing the best position up to that point in their memory. Crows in the population search for better food sources by moving through the search space until the termination criteria are met.

Assuming that the *j*th crow in the population needs nourishment and moves to the place where it stores its food, two situations can occur if it is assumed that the *i*th crow in the population is following to steal the food of the *j*th crow [15],[17]:

Case 1: *j*th crow is unaware that *i*th crow is following its. This situation is mathematically expressed as $r_j \geq AP^{j,t}$ in the CSA. As a result of this situation, *i*th crow will approach the place where *j*th crow stores its nutrients and updates its position information with the equation in Equation (6).

$$K^{i,t+1} = K^{i,t} + r_i \times fl^{i,t} \times \left(B^{j,t} - K^{i,t}\right) \qquad r_j \geq AP^{j,t} \tag{6}$$

where r_i is a random number that is evenly distributed between 0 and 1, $fl^{i,t}$ is the flight length of the *i*th crow in the *t*th iteration, Fig. 2 shows the scheme of this situation and the effect of *fl* on the search ability. *fl* significantly affects the algorithm's search capability. If *fl* is determined at small values (around $K^{i,t}$), it increases the local search capability. If *fl* is determined in large values (far from $K^{i,t}$), it increases the global search capability.

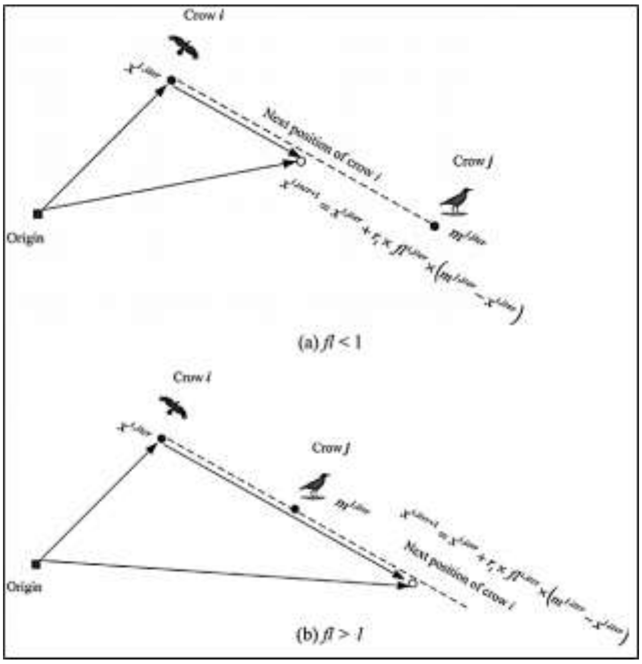

Crow i
Crow j
Origin
$x^{i,iter}$
Next position of crow i
$m^{j,iter}$
$x^{i,iter+1} = x^{i,iter} + r_i \times fl^{i,iter} \times (m^{j,iter} - x^{i,iter})$
(a) fl < 1
Crow i
Crow j
Origin
$x^{i,iter}$
$m^{j,iter}$
$x^{i,iter+1} = x^{i,iter} + r_i \times fl^{i,iter} \times (m^{j,iter} - x^{i,iter})$
Next position of crow i
(b) fl > 1

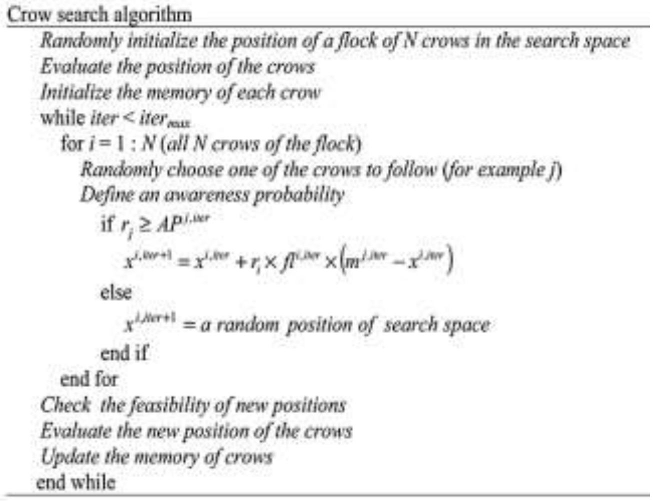
Fig. 2 Effect of *fl* Parameter in CSA [15]

Case 2: *j*th crow is aware that the *i*th crow is following it. In this case, the *j*th crow goes on the defensive to prevent the *i*th crow from discovering the food source, and misleads the *i*th crow and moves to another randomly determined location in the search space.

Considering Case 1 and Case 2, the position update mechanism of crows can be represented by the equation in Equation (7):

$$K^{i,t} = \begin{cases} K^{i,t+1} = K^{i,t} + r_i \times fl^{i,t} \times B^{j,t} - K^{i,t}, & r_j \geq AP^{j,t} \\ random\ position & ,otherwise \end{cases} \tag{7}$$

where r_i is a random number that is evenly distributed between 0 and 1, $AP^{j,t}$ is the awareness probability of *j*th crow in *t*th iteration. The pseudo-code of the Crow Search Algorithm proposed by Alireza Askarzadeh is given in Fig. 3.



```
Crow search algorithm
    Randomly initialize the position of a flock of N crows in the search space
    Evaluate the position of the crows
    Initialize the memory of each crow
    while $iter < iter_{max}$
       for $i = 1 : N$ (all N crows of the flock)
          Randomly choose one of the crows to follow (for example j)
          Define an awareness probability
             if $r_j \geq AP^{j,iter}$
                $x^{i,iter+1} = x^{i,iter} + r_i \times fl^{i,iter} \times \left(m^{j,iter} - x^{i,iter}\right)$
             else
                $x^{i,iter+1}$ = a random position of search space
             end if
       end for
    Check the feasibility of new positions
    Evaluate the new position of the crows
    Update the memory of crows
   end while
```

Fig. 3 Pseudo code of the Crow Search Algorithm [15]

2.3. Training ANN using the CSA algorithm (CSA-MLP)

Crow Search Algorithm is recommended to obtain the optimum connection weight and bias values of the MLP. In the Crow Search Algorithm, each individual (crow) in the herd represents a candidate solution for determining the optimum connection weight and bias values of the MLP. Each individual who creates the candidate solution is represented as a d-dimensional array of total connection weight and bias value size in the designed MLP model, and the candidate solution vector is created.

Fig. 4 shows the schematic structure of the MLP with a single hidden layer and the example of the representation in the candidate solution vector of this structure.

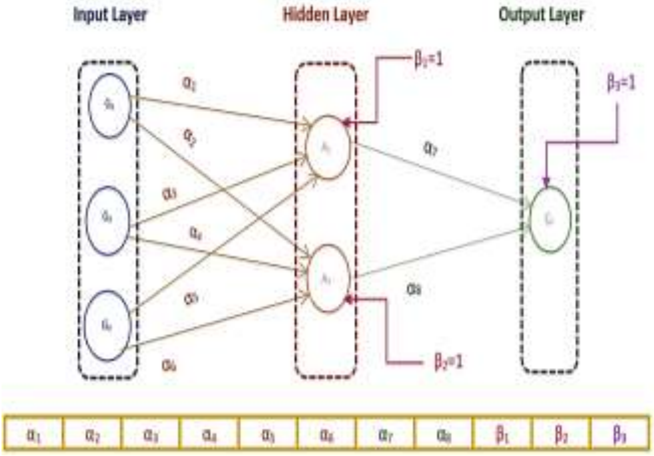


Fig. 4 Schematic structure of the MLP and illustration of the candidate solution vector

In Fig. 4, α_1, α_2, α_3, α_4, α_5 and α_6 represent the connection weight values between the input layer and the neurons in the hidden layer. α_7 and α_8 represent the connection weight values between the hidden layer and the neurons in the output layer. β_1 and β_2 are the bias values of neurons in the hidden layer. β_3 is the bias value of the neuron in the output layer.

Within the scope of this article, the structure of the MLP consists of a single hidden layer. Equation (8) gives the formula to calculate the size of the candidate solution vector of the MLP with a single hidden layer.

$$\alpha = (n \times h) + (h \times o)$$
$$\beta = h + o$$
$$Candidate\ Solution\ Vector\ Size = \alpha + \beta \tag{8}$$

where n is the number of neurons in the input layer of the network, h number of neurons in the hidden layer, o number of neurons in the output layer, α total number of connection weight values in the network, β total number of bias values in the network.

In the literature, the mean square error calculation function called MSE (Mean Squared Error) is used in the process of determining the suitability of each candidate solution in MLP training [12]. MSE function is given in Equation (9).

$$MSE = \sum_{n=1}^{m} \frac{\sum_{i=1}^{k} (g_i^n - t_i^n)^2}{m} \tag{9}$$

where m number of training samples, k number of outputs, g_i^n is the actual output value of the ith input in the nth training example, t_i^n is the predicted output value of the ith input in the nth training example.

CSA which is recommended to determine the optimum connection weight and bias values of the MLP calculates the error value with the MSE by presenting the candidate solution vectors represented by the individuals and containing the connection weight and bias values to the MLP. In order to bring the average error value of the training samples used in the training phase to the minimum value, the CSA continues to update the connection weight and bias values in the candidate solution vectors represented by the individuals until the termination criteria are met. Schematic representation of the proposed method is given in Fig. 5.

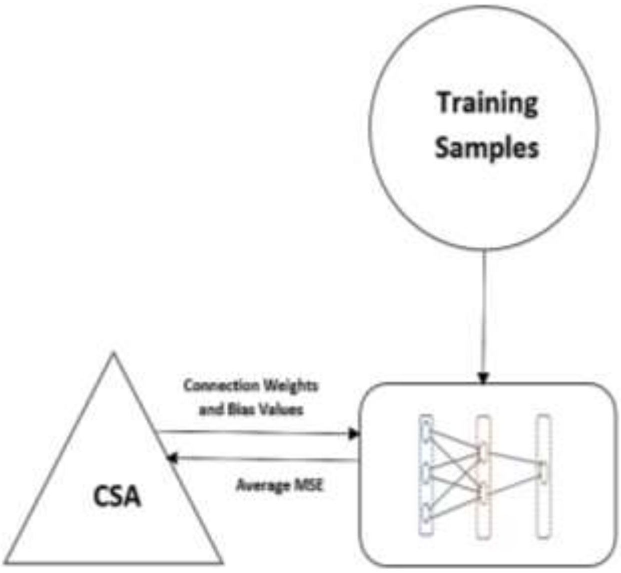


Fig. 5 Schematic display of the proposed method

3. Experimental Results

Experimental studies have been carried out to determine the success of the proposed CSA-MLP algorithm in order to provide the optimum connection weight and bias values of the MLP. The classification datasets (xor, balloon, iris, breast cancer and heart) used in the experiments were taken from the University of California Irvine Machine Learning Database. The datasets have different numbers of samples and features. The purpose of this is to verify the success of the proposed method by testing it under different conditions. In Table 1, information was given about the number of samples, the number of attributes, the number of classes, the number of samples to be used in the training, the number of samples to be used in the test, the MLP architecture and candidate solution vector size of the five data sets used in the experiments.

The CSA-MLP algorithm has been developed using MATLAB and the experiments have been carried out in the working environment on a computer with Windows 10 operating system, i5 3.0 GHz processor, 4 GB memory.

Minimum-Maximum Normalization was applied in order to achieve more successful results from the data set of the problem, to significantly reduce the calculation time and to represent the same value range by eliminating the scale difference between the attribute data [18]. This method normalizes the values of a D attribute to their minimum and maximum values. The Minimum-Maximum Normalization formulation is presented in Equation (10).

$$d' = \frac{d - min_d}{max_d - min_d} \tag{10}$$

where D is the corresponding attribute, d is the value of the attribute to be normalized, min_d is the smallest value among the attribute's values, max_d is the largest value of the attribute's values, d' is the normalized value of d.

Parameter information for SMS and CSA is presented in Table 2. The parameter values of the SMS are taken from the study [19]. The initial values of the connection weight values and bias values in the MLP structure were randomly determined in the range [-10,10]. In order to verify the success of the model, 30 runs were performed on each data set with independent initial parameters. In order to determine the success of the proposed model, the CSA-MLP algorithm was compared with the SMS-MLP algorithm in terms of the MSE value, the classification rate, the statistical metrics and the convergence graph.

The CSA-MLP and SMS-MLP were tested with the test dataset. The average and standard deviation of the MSE values and average classification rate, which are the results of the experiments, are presented in Table 3. Table 3 shows that the CSA-MLP algorithm left the SMS-MLP algorithm behind in terms of average and standard deviation of the MSE values on the five data sets. This means that the CSA-MLP algorithm is more successful than the SMS-MLP algorithm. When the average classification rate on the test dataset presented in Table 3 was analyzed, it is seen that the CSA-MLP algorithm outperforms the SMS-MLP algorithm. The CSA-MLP algorithm achieved the classification rates over 95% on all datasets without heart. The SMS-MLP algorithm achieved the classification rates over 80% on all datasets without heart. As a result, the CSA-MLP algorithm achieved better results than the SMS-MLP algorithm on five classification data sets.

The sensitivity, specificity, precision, f$_1$-score performance metrics are used to evaluate classification models that estimate class labels from certain input data. The basis of these criteria is based on the estimated class label results with the real class label, which are expressed as true positive (TP), true negative (TN), false positive (FP) and false negative (FN). The sensitivity metric shown in Equation (11), measures the proportion of positively classified positive samples on all samples belonging to the positive grade. The specificity metric shown in Equation (12), measures the proportion of negative samples, correctly classified on all samples belonging to the negative grade. The precision metric shown in Equation (13), measures the proportion of correctly classified positive samples, on all samples classified as positive. The f1-score metric shown in Equation (14) determines the balance between sensitivity and precision [19]. The results of the metrics for each dataset, are presented in Table 4. When the CSA-MLP results in Table 4 are analyzed, there is a balance between sensitivity and specificity on xor, balloon, iris and breast cancer datasets. However, the CSA-MLP algorithm has 80.62% sensitivity percentage and 66.22% specificity percentage on the heart dataset. The CSA-MLP algorithm has high precision percentage and high F$_1$-score value on all the datasets. When the SMS-MLP results in Table 4 are analyzed, there is a balance between sensitivity and specificity on xor, balloon and iris datasets. However, the SMS-MLP algorithm has a low sensitivity percentage and a high specificity percentage on the breast cancer and heart datasets. The CSA-MLP algorithm has a high precision percentage on all the datasets. The SMS-MLP algorithm has an acceptable F$_1$-score value on xor, balloon, iris and heart datasets. However, it has a low F$_1$-score value on the breast cancer dataset. According to the performance metrics, the CSA-MLP algorithm is a successful algorithm and has high values than the SMS-MLP in all the performance metrics.

$$Sensitivity = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{11}$$

$$Specificity = \frac{\mathrm{TN}}{\mathrm{TN+FP}} \tag{12}$$

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \tag{13}$$

$$F_1 - score = 2 \times \frac{\mathrm{Precision \times Sensitivity}}{\mathrm{Precision + Sensitivity}} \tag{14}$$

Table 1. Properties of the datasets

Dataset	Attribute Number	Number of Training Samples	Number of Test Samples	Class Number	MLP Architecture	Candidate Solution Vector Size
Xor	3	8	8	2	3-7-1	36
Balloon	4	20	20	2	4-9-1	55
Iris	4	150	150	3	4-9-3	75
Breast Cancer	9	599	100	2	9-19-1	210
Heart	22	80	187	2	22-45-1	1081

Table 2. Parameter information for CSA and SMS algorithm

Algorithm	Parameter	Value
CSA	*AP* (awareness possibility)	0.1
	fl (flight length)	2
	Population size	50 for xor and balloon, 200 for the other datasets
	Maximum number of iterations	250
SMS	α, β, H, ρ (gas state)	0.3, 0.9, 0.9, 0.85
	α, β, H, ρ (liquid state)	0.05, 0.5, 0.2, 0.35
	α, β, H, ρ (solid state)	0, 0.1, 0, 0.1
	Population size	50 for xor and balloon, 200 for the other datasets
	Maximum number of iterations	250

Table 3. The results of the CSA-MLP and SMS-MLP on datasets

Dataset	MSE (Avg±Std)		Average Classification Rate	
	CSA-MLP	SMS-MLP	CSA-MLP	SMS-MLP
Xor	**1.30E-02±2.24E-02**	1.33E-01±3.39E-02	**99.58**	83.75
Balloon	**2.41E-06±7.66E-06**	1.11E-02±1.31E-02	**100.00**	99.17
Iris	**2.82E-02±3.38E-03**	2.58E-01±5.12E-02	**98.51**	86.78
Breast Cancer	**1.50E-03±7.20E-05**	2.27E-02±4.33E-03	**99.30**	85.67
Heart	**9.79E-02±1.39E-02**	1.18E-01±3.13E-02	**79.47**	68.57

Table 4. The result of performance determination criteria

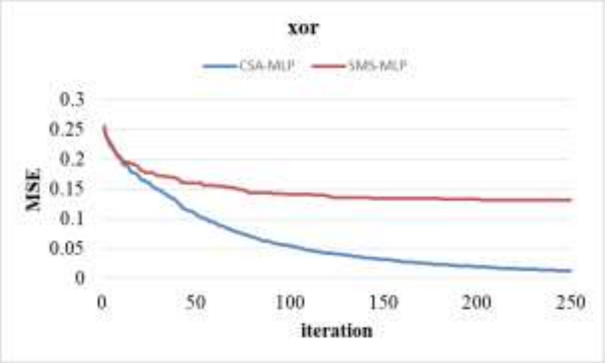
Algorithm	xor				balloon			
	sensitivity	specificity	precision	F_1-Score	sensitivity	specificity	precision	F_1-Score
CSA-MLP	100.00%	99.17%	99.33%	0.9963	100.00%	100.00%	100.00%	1.0000
SMS-MLP	81.70%	86.70%	88.90%	0.8349	99.20%	99.20%	98.80%	0.9894
	iris				breast cancer			
	sensitivity	specificity	precision	F_1-Score	sensitivity	specificity	precision	F_1-Score
CSA-MLP	98.58%	99.29%	98.58%	0.9858	100.00%	99.11%	96.83%	0.9838
SMS-MLP	82.10%	91.10%	81.00%	0.8002	50.20%	95.10%	83.70%	0.5828
	heart							
	sensitivity	specificity	precision	F_1-Score				
CSA-MLP	80.62%	66.22%	96.50%	0.8775				
SMS-MLP	68.39%	70.67%	96.41%	0.7980				

Table 5. The comparison of the algorithms

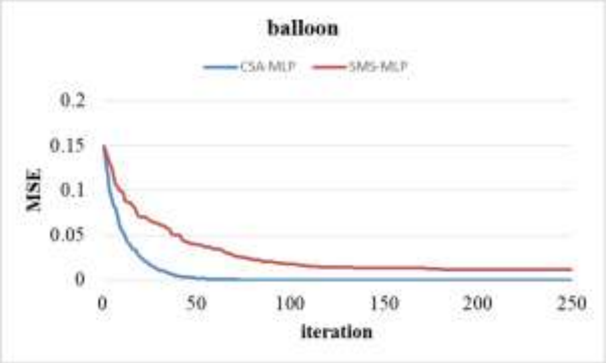
Dataset	Best Classification Accuracy (%)							
	CSA-MLP	SMS-MLP	GWO-MLP	PSO-MLP	GA-MLP	ACO-MLP	ES-MLP	PBIL-MLP
Xor	**100.00**	**100.00**	**100.00**	37.50	**100.00**	62.50	62.50	62.50
Balloon	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**
Iris	**98.67**	93.33	91.33	37.33	89.33	32.66	46.66	86.66
Breast Cancer	**100.00**	93.00	99.00	11.00	98.00	40.00	06.00	07.00
Heart	**86.63**	77.54	75.00	68.75	58.75	00.00	71.25	45.00

The proposed CSA-MLP algorithm is compared with the seven algorithms in literature: SMS-MLP based on the States of Matter Search Algorithm, GWO-MLP based on Grey Wolf Optimization, PSO-MLP based on Particle Swarm Optimization, GA-MLP based on Genetic Algorithm, ACO-MLP based on Ant Colony Optimization, ES-MLP based on Evolution Strategy and PBIL-MLP based on Population-based Incremental Learning. The results of these seven algorithms were taken from the reference [19]. The population size in the CSA-MLP, SMS-MLP and the other algorithms for xor and balloon datasets is 50. The population size for other datasets is 200. The maximum number of iterations is 250. The initial values of the connection weight values and bias values in the MLP structure were randomly determined in the range [-10, 10]. Table 5 shows the comparison results of the algorithms according to the best classification accuracy. The CSA-MLP algorithm has the best results on iris, breast cancer and heart datasets than the other algorithms. The CSA-MLP, SMS-MLP, GWO-MLP and GA-MLP algorithms have the best results on the xor dataset. All the algorithms has the best results on the balloon dataset. The experimental results show that the CSA-MLP algorithm is more successful and effective than the other seven algorithms.

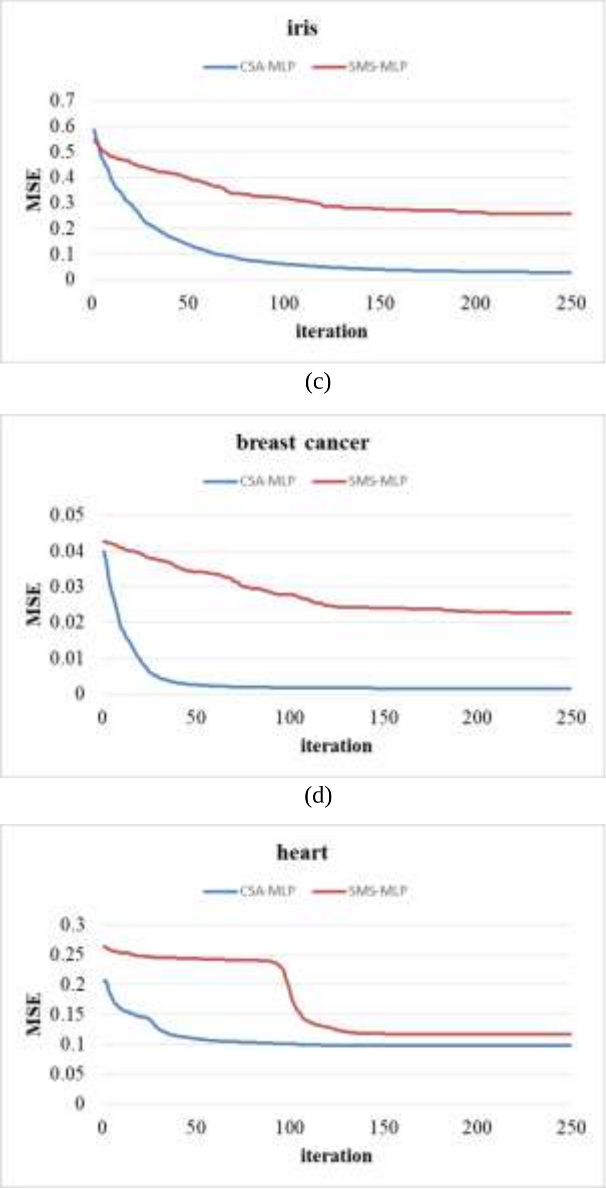
Fig. 6 shows the convergence graphs on the datasets of the algorithms and it is seen that the CSA-MLP algorithm attempts to reach the global optimum quickly and successfully than the SMS-MLP during the iterations.



(a)



(b)



(c)

(d)

(e)

Fig 6. The convergence graphs of the CSA-MLP and SMS-MLP on the datasets (a-e)

4. Conclusions

In this article, the CSA-MLP algorithm was proposed for training Multi-Layer Perceptron (MLP) using Crow Search Algorithm (CSA). The CSA algorithm is a population-based meta-heuristic optimization algorithm, inspired by the behavior of crows to store their surplus nutrients and take them back from the storage area when needed. CSA has been proposed to solve different optimization problems in terms of its simplicity with two different adjustable parameters (flight length and awareness probability), obtaining an effective convergence rate in a short time and having a faster technique compared to algorithms frequently used in engineering problems with different constraints and functions. Each individual in the population tries to bring the connection weights and bias values of the MLP to the optimum level. Each individual in the population is represented by a candidate solution vector containing connection weight and bias values. In the CSA-MLP algorithm, the candidate solution vectors are updated at each iteration using the data set reserved for training. The suitability of the solutions obtained by individuals is evaluated using MSE. At the end of the training, the values of the connection weights and bias for MLP are determined by the candidate solution vector with the minimum MSE. Then, MLP is tested with the test dataset. In the experiments, five classification datasets (xor, balloon, iris, breast cancer, heart) were used. Datasets were taken from the UCI Machine Learning Database. The CSA-MLP algorithm was compared with the SMS-MLP algorithm in terms of the MSE values, average classification rate, the performance metrics and the convergence graph. The CSA-MLP algorithm left the SMS-MLP algorithm behind in terms of average and standard deviation of the MSE values on the five data sets. This means that the CSA-MLP algorithm is more successful in terms of error than the SMS-MLP algorithm. When the average classification rate on the test dataset was analyzed, it is seen that the CSA-MLP algorithm outperforms the SMS-MLP algorithm. The CSA-MLP algorithm achieved the classification rates over 95% on all datasets without heart. The SMS-MLP algorithm achieved the classification rates over 80% on all datasets without heart. As a result, the CSA-MLP algorithm achieved better results than the SMS-MLP algorithm on five classification data sets. Furthermore, the CSA-MLP algorithm attempts to reach the global optimum quickly and successfully than the SMS-MLP algorithm during the iterations. The proposed CSA-MLP algorithm was compared with seven algorithms in literature (SMS-MLP, GWO-MLP, PSO-MLP, GA-MLP, ACO-MLP, ES-MLP and PBIL-MLP) in terms of best classification accuracy. The CSA-MLP, SMS-MLP and GWO-MLP algorithms achieved %100 success on the xor dataset, all the algorithms achieved 100% success on the balloon dataset. The CSA-MLP algorithm has the best results %98.67 on iris, %100 on breast cancer and %86.63 on heart datasets than the other algorithms. The results show that the CSA-MLP algorithm is more successful and effective than the seven algorithms.

The experimental results show that the Crow Search Algorithm is a reliable approach in Multi-Layer Perceptron Neural Network training. The proposed CSA-MLP algorithm achieved better results than the SMS-MLP and other algorithms.

Within the scope of this study, the CSA algorithm was used only to optimize the values of the connection weights and biases of the MLP. As future work, the network structure can be optimized and also the proposed model can be tested on other data sets. In addition, the proposed method can be improved by obtaining optimum values for the awareness probability and flight length parameters in CSA.

References

[1] Tumer, A.E. and S. Edebali, An artificial neural network model for wastewater treatment plant of Konya. International Journal of Intelligent Systems and Applications in Engineering, 2015. 3(4): p. 131-135.

[2] Ergezer, H., M. Dikmen, and E. Özdemir, Yapay sinir ağları ve tanıma sistemleri. PiVOLKA, 2003. 2(6): p. 14-17.

[3] Yegnanarayana, B., Artificial neural networks. 2009: PHI Learning Pvt. Ltd.

[4] Öztemel, E., Yapay sinir ağları. 2012: Papatya.

[5] Yurdakul, E.M., Türkiye'de ithalatın gelişimi ve ithalatın yapay sinir ağları yöntemi ile tahmin edilebilirliğine yönelik bir analiz. 2014.

[6] GÜNLER, M.A., Identification of Facial Expressions Using Artificial Neural Networks (Ann). a Master's Thesis, Electrical & Electronics Engineering, 2012.

[7] Yamany, W., et al. Moth-flame optimization for training multi-layer perceptrons. in 2015 11th International computer engineering Conference (ICENCO). 2015. IEEE.

[8] Jaddi, N.S., S. Abdullah, and A.R. Hamdan, Optimization of neural network model using modified bat-inspired algorithm. Applied Soft Computing, 2015. 37: p. 71-86.

[9] Mohammadi, F. and H. Abdi, A modified crow search algorithm (MCSA) for solving economic load dispatch problem. Applied Soft Computing, 2018. 71: p. 51-65.

[10] Gupta, D., et al., Improved diagnosis of Parkinson's disease using optimized crow search algorithm. Computers & Electrical Engineering, 2018. 68: p. 412-424.

[11] Kocyigit, Y., A. Alkan, and H. Erol, Classification of EEG recordings by using fast independent component analysis and artificial neural network. Journal of Medical Systems, 2008. 32(1): p. 17-20.

[12] Aljarah, I., H. Faris, and S. Mirjalili, Optimizing connection weights in neural networks using the whale optimization algorithm. Soft Computing, 2018. 22(1): p. 1-15.

[13] Rincon, P., Science/nature| crows and jays top bird IQ scale. BBC News, 2005.

[14] Prior, H., A. Schwarz, and O. Güntürkün, Mirror-induced behavior in the magpie (Pica pica): evidence of self-recognition. PLoS biology, 2008. 6(8): p. e202.

[15] Askarzadeh, A., A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Computers & Structures, 2016. 169: p. 1-12.

[16] Emery, N.J., Cognitive ornithology: the evolution of avian intelligence. Philosophical Transactions of the Royal Society B: Biological Sciences, 2006. 361(1465): p. 23-43.

[17] Díaz, P., et al., An improved crow search algorithm applied to energy problems. Energies, 2018. 11(3): p. 571.

[18] Sola, J. and J. Sevilla, Importance of input data normalization for the application of neural networks to complex industrial problems. IEEE Transactions on nuclear science, 1997. 44(3): p. 1464-1468.

[19] Gulcu, Ş., Training of the Artificial Neural Networks using States of Matter Search Algorithm. International Journal of Intelligent Systems and Applications in Engineering, 2020. 8(3): p. 131-136.