Int. J. Nonlinear Anal. Appl. 14 (2023) 1, 843–852
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.6904

Adaptive trainer for multi-layer perceptron using artificial gorilla troops optimizer algorithm

Tuqa Ali Mohamed*, Muntadher Khamees Mustafa

Department of Computer Science, College of Science, University of Diyala, Iraq

(Communicated by Javad Vahidi)

Abstract

This paper utilized a newly proposed algorithm based on meta-heuristics for the training of multilayer perceptrons (MLP) that was developed using the idea of artificial gorilla troops optimizers. The precision and consistency of the proposed method's convergence as performance metrics. The Artificial Gorilla Troops Optimizer (GTO) was recently proposed for use in training MLP, and it employs the five most common classification data sets currently available(XOR, balloon, heart Iris, breast cancer) in the California University at Irvine UCI Repository. The newly Optimizers (GTO) are used for the first time as a Multi-Layer Perceptron (MLP) trainer, and its results are compared to those obtained using the more established grey wolf optimization (GWO), the whale optimization algorithm (WOA), and the sine cosine algorithm (SCA). Previously, GTO was used to determine the best weights and biases for the optimal solutions.

Keywords: training MLP, ANN, WOA, SCA, GWO, GTO
2020 MSC: 68T20, 90C31, 68Q87

1 Introduction

Neural networks (NN) are one of the most ground-breaking developments in the field of AI. By modeling neuronal activity in the human brain, [17] in the research literature [16], they typically attack classification problems. Many different kinds of NNs have been proposed [4, 14] Self-referential the networks feed [20] the RBF network stands for "radial basis function." [9] a recurrent neural network, in addition to the convolutional neural network. For instance, neural networks that "spike" are a type of self-organizing network. In feed forward NNs (FNN), data is sent in a single direction across the network. Recurrent NNs, on the other hand, as their name implies, allow for bi-directional neuronal communication. The final process involves spiking. NNs cause neuronal stimulation with spikes. The use of NNs in education is widespread. When a NN has the ability to learn, it means that it can improve itself through exposure to new information. Like real neurons, artificial neural networks (ANN) can learn from experience and improve with new information [12]. Both supervised and unsupervised learning methods are applied commonly in this setting. The first instance is when the NN is responding to feedback from the outside world (supervisor). On the other hand, learning unsupervised, a NN makes changes to inputs (learns) without any additional external feedback [18].

The two most important reasons for doing this work are as follows:

*Corresponding author
Email addresses: scicompms2108@uodiyala.edu.iq (Tuqa Ali Mohamed), alkarawis@gmail.com (Muntadher Khamees Mustafa)

Received: July 2022 *Accepted:* September 2022

- Exploration and exploitation are two strong points in GTO's repertoire, which could propel it past the competition.
- There is still a problem with multi-solution stochastic trainers getting stuck at the local optimum.

Hamidzadeh et al. [10] in proposed system DDC uses a distance-based decision surface with nearest neighbor projection. DDC's kernel type has been enhanced to include nonlinear data structures. DDC doesn't require traditional learning (like k-NN), searching time to find k-nearest neighbors, or optimization (unlike SVM) (SVM). DDC computes the weighted average of training sample distances. Unclassified sample goes to class with least distance. Such a rule can be utilized to derive a decision-making formula. DDC performed between k-NN and SVM in most cases. DDC doesn't include training. To increase DDC's performance by using alternate or novel distances based on data distribution. Sample data may be spherical for an effective projection line. Kernel and may be reducing data helps get decision surfaces.

Mirjalili et al. [19] in proposed system utilization of These problems will be alleviated somewhat with the help of the recently established Biogeography-Based Optimization (BBO) algorithm for training MLPs. In order to explore the efficacy of BBO in training MLPs, we use five classification datasets in addition to six function approximation datasets. The limitation of not being able to train BBO in other types of NNs, such as recurrence, Kohonen, or Radial basis'(RBF) networks, as well as the method Biogeography-Based Optimization (BBO) and data set to balloon, iris, breast cancer, heart, sigmoid, cosine with one peak, sine with four peaks, sphere, Griewank, and Rosenbrock.

Lee and Choeh [15] in the proposed system Create mathematical models that can accurately forecast the usefulness of reviews and make available a tool that can locate the reviews that are the most valuable for a particular product. The purpose of this research is to propose HPNN, which is a neural network-based helpfulness prediction model. HPNN employs a back-propagation multilayer perceptron neural network (BPN) model to predict the level of review helpfulness using the determinants of product data, the review characteristics, and the textual characteristics of reviews. Specifically, this research will focus on how these factors interact with one another. Utilizing Artificial Neural Networks, the Helpfulness Prediction Model (HPNN) Keep your attention solely on these results while you investigate the factors that determine whether online consumer evaluations are positive or negative. In turn, individuals' attitudes toward internet buying can be changed by the provision of constructive feedback.

Mirjalili [17] in proposed system First-time MLP training uses Grey Wolf Optimizer (GWO). Five classification datasets and three function-approximation datasets are used to evaluate the proposed technique. The results are compared to popular evolutionary trainers. PIL, PSO, GA, ACO, and ES are incremental learning methods (PBIL) Five classification and three function-approximation datasets (XOR, balloon, Iris, breast cancer, and heart) were trained using a GWO-based trainer (sigmoid, cosine, and sine). GWO hasn't been used to identify the appropriate number of hidden nodes, layers, and other MLP characteristics. This algorithm needs more research to be fine-tuned.

Ramchoun et al. [21] proposed The Artificial Neural Network design is being optimized through the development of a system. When it comes to finding the best solution to a nonlinear problem, the Genetic Algorithm is an approach that works particularly well. Following training, this approach is analyzed to find the ideal weights matrix, as well as the optimal number of hidden layers and connection weights in the Multilayer Perceptron. The problem of optimizing the multilayer perceptron architecture has been recast by us as a constrained mixed-integer problem, which is the subject of our latest model. The results that were obtained provide evidence of the satisfactory generalization of neural network designs that are founded on iris data. The EBP method and approach Other databases provide only a limited amount of instruction on real-world issues such as diabetes, thyroid disease, and cancer.

Aljarah et al. [3] proposed system a brand new training algorithm that was derived from the whale optimization technique that was only recently proposed (WOA). It has been proved that this method is superior to other algorithms in terms of performance and can solve a much wider variety of optimization issues. Because of this, we were compelled to investigate its usefulness in the training of feedforward neural networks. WOA in the training for the MLP. The high local optima avoidance and rapid convergence speed were the primary drivers behind applying the WOA to the issue of training MLPs as a solution to the problem. At first, training MLPs was viewed as a minimization problem rather than a problem that needed to be solved. It has been suggested that further varieties of ANNs should be taught utilizing WOA. It is worthwhile to contemplate the manner in which WOA-trained MLPs can be applied to the resolution of engineering classification issues. In addition, a significant contribution can be made by applying the WOA-trained MLP to the resolution of datasets containing function approximation issues.

Hesami et al. [13] proposed a system the Multilayer Perceptron- Nondominated Sorting Genetic Algorithm-II was used to examine in vitro chrysanthemum sterilization. Modeling and optimizing the process were study goals (MLP-NSGAII). MLP predicted contamination frequency and explant vitality. HgCl2, Ca(ClO)2, Nano-silver, H2O2,

NaOCl, AgNO3, and immersion times were modeled (EV). Then, models were connected to NSGAII to optimize the process, and sensitivity analysis was used to discover each input's relevance. Both stages were taken for the best outcomes. All training and testing R2 values were above 94%. According to MLP-NSGAII, 1.62% NaOCl for 13.96 minutes using procedure GA yields optimal CF (0.0%) and EV (99.98%) (MLP-NSGAII). FRP (MLP-NSGAII). MLP does not evaluate multi-objective optimization methods in plant science, especially plant tissue culture.

Hesami et al. [11] proposed a system a newly developed hybrid stochastic training approach for multilayer perceptron's (MLPs) neural networks that make use of the grasshopper optimization algorithm that was just proposed (GOA). The GOA algorithm is a relatively new method that has the ability to solve a wide variety of optimization-related issues as a result of its adaptable and versatile search mechanisms. It is feasible to obtain a satisfying performance if one avoids the trap of local optima and strikes a balance between the tendencies of exploration and exploitation. After that, the GOAMLP model that was proposed is used to analyze data from five important datasets, including those on individuals with breast cancer, Parkinson's disease, diabetes, coronary heart disease, and orthopaedic conditions. The GOA, GOA approach can be utilized for analyzing other varieties of NNs in addition to big data sets.

Samadianfard et al. [22] proposed system AI algorithms select the finest weights in neural network layers to extract relevant input data attributes to generate an accurate model. Input data is essential for building the most accurate predictive model and evaluating wind energy potential. In this study, a viable and robust method for predicting wind speed for ten locations is proven by utilizing input data from neighboring reference locations. In the current study, daily wind speed values are predicted using the MLP, MLP-WOA, and MLP-GA models for each of the ten target stations. This approach forecasts wind speed without using climatic or atmospheric data. IRIMO employed several statistical indicators to evaluate MLP-WOA. Unspecified data utilized.

Al-Badarneh et al. [2] proposed system A technique for training the MLP that makes use of three different evolutionary classification algorithms. It is proposed to use the models GWO-MLP, PSO-MLP, and SSA-MLP. Accuracy, f1-score, and g-mean are the three fitness functions that are adopted by the suggested approaches. Ten datasets that are imbalanced are used to test these fitness functions. Calculations were made to determine the average results of 30 separate runs, the best values, and the standard deviations for each parameter. According to the findings, neither approach is demonstrably superior to the other. Experiments, however, showed that when the dataset is imbalanced, g-mean and f-score fitness functions have a clear advantage over the classification accuracy rate. This was the case even though the classification accuracy rate was the fitness function that was originally chosen. When an increase in the recall of both classes is required, it is recommended to use neuro-evolutionary models that have a g-mean fitness function (e.g. major and minor). When the minor class is more significant than the major class, it is preferable to use the f-measure of the minor class as the fitness function. It was decided not to investigate how the proposed meta-heuristic approach for optimizing Convolutional Neural Networks (CNN) may be applied to complicated tasks such as the classification of images and text. Declaration.

Ecer et al. [8] proposed results, Single models have a faster processing time, but they're less accurate than hybrid models. Numerous studies agree. Hybrid models make MLP–PSO faster and more precise than MLP–GA. Discussions In this study, MLP–GA and MLP–PSO were used in two scenarios, with $\tanh(x)$ and the Gaussian function as default output functions for 13 categories. RMSE and correlation coefficients were utilized to compare training and testing model correctness and performance. Using $\tanh(x)$ as the output function enhanced model accuracy. MLP–PSO with a population size of 125, followed by MLP–GA with a population size of 50, gave superior testing accuracy, as indicated by RMSE values of 0.732583 and 0.7333063, MAPE values of 28.16% and 29.09%, and correlation coefficient values of 0.694 and 0.695, respectively. Single MLPs are faster to process, but they are less accurate than hybrid models. That stock market swings won't be dealt with adequately, despite being treated satisfactorily. The study's return difference is a problem.

2 The feed-forward neural network with the multilayer perceptron

As was discussed FNNs are NNs with unidirectional neuronal connections [1]. This NN stacks neurons in numerous layers. First is input, then output. Hidden layers are between input and output. One-hidden-layer FNN or MLP [4]. MLPs output inputs, weights, and biases [1, 17]. First, we compute the input weight summing by:

$$S_j = \sum_{i=1}^{n} (W_{ij} \times X_i) - \theta_j, \quad j = 1, 2, ..., h \tag{2.1}$$

where n is the total number of input nodes, W_{ij} represents the weight of the link from the ith input node to the

jth hidden node, j is the bias (threshold) of the jth hidden node, and X_i is the Ith input. n is the total number of input nodes. The following calculation is used to determine the output of each concealed node:

$$S_j = sigmoid\ S_j = \frac{1}{(1 + \exp -(-S_j))}, \quad j = 1, 2, ..., h \tag{2.2}$$

Second the calculated results from the hidden nodes (hidden layar) are used to define the final results in the following ways:

$$O_k = h \sum_{j=1}^{h} (W_{jk}.S_j) - \theta_k, \quad k = 1, 2, ..., m \tag{2.3}$$

$$O_k = sigmoid(O_k) = \frac{1}{(1 + \exp(-O_k))}, \quad k = 1, 2, ..., m \tag{2.4}$$

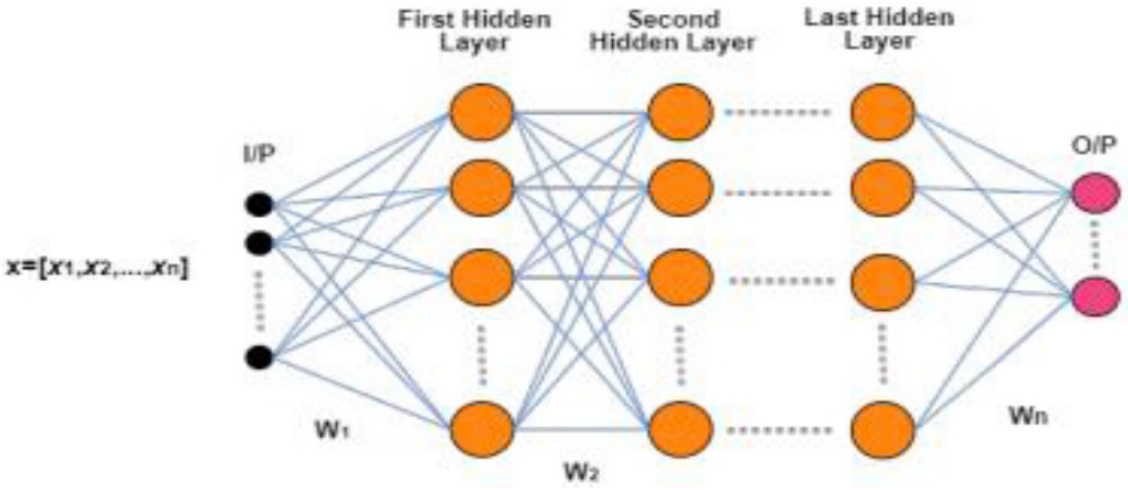

Figure 1: General MLP Architecture [7].

3 Artificial gorilla troops optimizer (GTO)

To optimize complex problems, researchers have developed a novel meta-heuristics algorithm called the "Artificial Gorilla Troop's Optimizer" (GTO), which takes cues from the social and cooperative nature of real-life gorilla groups. As with other metaheuristics, GTO's convergence accuracy and stability suffer as the optimization problems to be solved grow in complexity and variety [23]. These flaws necessitate the development of new mechanisms to carry out exploration and exploitation and to help achieve better performance. A troop consists of a dominant adult male gorilla (also called a silverback), several dominant adult females, and their offspring. Silverback gorillas are over 12 years old and get their name from the distinctive hair that grows on their backs when they reach puberty. In addition, the silverback is the leader of the entire troop and is responsible for making all decisions, mediating any conflicts that arise, allocating food and other resources, planning and executing group travel, and ensuring everyone's safety. Male gorillas between the ages of 8 and 12 are considered "black" because their silver fur is not fully developed. It is common for Gorillas to leave their birth group to join a third. But occasionally a few of the male gorillas will decide to stick around and keep following the silverback. These males may fight viciously for control of the group and access to adult females if the silverback is killed. The concept of group behavior in wild gorillas serves as the inspiration for the GTO algorithm's unique mathematical model. Initialization, global exploration, and local exploitation are the three separate steps that makeup GTO, much like they are in other intelligent algorithms.

3.1 Initialization phase

Let's pretend the D-dimensional space contains N gorillas. In order to specify where the $i-th$ gorilla is in the universe, we can write $X_i = (x_{i,1}, x_{i,2}, ..., x_{i,D})$, where $I = 1, 2, ..., N$. The establishment of a gorilla population can thus be defined as:

$$X_{N\times D} = rand(N, D) \times (ub - lb) + lb \tag{3.1}$$

The search range is defined by its upper and lower bounds, ub and lb, and the matrix X has A random number between 0 and 1 is assigned to each element of the N rows and D columns in the matrix. denoted by $rand(N, D)$.

3.2 Exploration phase

$$\begin{aligned} &GX(t+1) = (ub - lb) \times r2 + lb, \qquad r1 < p \\ &(r3 - C) \times XA(t) + L \times Z \times X(t), \quad r1 \geq 0.5 \\ &X(t) - L \times (L \times (X(t) - XB(t)) + r4 \times (X(t) - XB(t))) \quad r1 < 0.5 \end{aligned} \tag{3.2}$$

Here, t represents iteration times, $X(t)$ is the gorilla's current position vector, and $GX(t+1)$ is the potential search agent location for the next iteration. On top of that, each of the random numbers $r1, r2, r3$, and $r4$ is a number between 0 and 1. Two positions among the current population of gorillas, $XA(t)$ and $XB(t)$, were chosen at random. p is a fixed value. Using the problem's dimension as an index, Z is a row vector whose elements' values are drawn at random from $[-C, C]$. In addition, C is determined by solving for it in Eq (3.3).

$$C = (\cos(2 \times r5) + 1) \times (1 - t/Maxiter) \tag{3.3}$$

where $\cos(\bullet)$ is the cosine function, r_5 is a positive real number between 0 and 1, and Maxiter is the maximum number of iterations. It is possible to calculate L, the value of the parameter in Eq. (3.4):

$$L = C \times l \tag{3.4}$$

where l is an arbitrary number between -1 and 1. After all possible $GX(t+1)$ solutions have been generated as a result of the exploration, their fitness values are compared. In the event that GX outperforms X, it will be kept and used in place of X. This is indicated by the condition $F(GX) < F(X)$, where $F(\bullet)$ is the fitness function for the problem in question $x(t)$. Furthermore, the best option available at the time is now deemed to be the silverback. An X of the Silverback.

3.3 Exploitation phase

When a new troop of gorillas is formed, the silverback is the dominant male and is at the peak of his strength and health. They follow the silverback gorilla's every directive as they forage for food and provide for him. Unavoidably, the silverback will age and die, and in his place, younger blackbacks in the troop may engage in Fighting over mating and leadership with other males. GTO's exploitation phase models following the silverback and competing for adult female gorillas. W is introduced to control this transition. If C in Eq. (3.4) is greater than W, follow the silverback's first mechanism. Math expression:

$$GX(t+1) = L \times M \times (X(t) - X\text{silverback}) + X(t) \tag{3.5}$$

In this case, the best solution found so far is denoted by Xsilverback, and the current position vector is denoted by $X(t)$, and L is also evaluated using Eq. (3.5). The value of M could also be determined by utilizing Eq. (3.6):

$$M = (\sum_{i=1}^{n} xi(t)/N|_{xi(t)/N}|^2 l)\frac{1}{2l} \tag{3.6}$$

where N is the total number of individuals and $Xi(t)$ is a vector representing the location of a gorilla.

$$GX(t+1) = X\text{silverback} - (X\text{silverback} \times Q - X(t) \times Q) \times A \tag{3.7}$$

$$Q = 2 \times r6 - 1 \tag{3.8}$$

$$A = \phi \times E, \quad E = N1, r7 \geq 0.5 \tag{3.9}$$

$$N2, r7 < 0.5 \tag{3.10}$$

It is the current position, denoted by $X(t)$, and the impact force, Q, that is calculated with Eqs. (3.7),(3.8). A random number between 0 and 1 is used for $r6$ in Eq. (3.4). As an added bonus, Eq. (3.9) can be used to assess the effectiveness of the coefficient A used to simulate the level of violence in the game. With Eq. denoting a constant, we can determine what numbers represent E. (3.10). Equation (3.6) includes $r7$, which is also a number chosen at random from the range $[0, 1]$. Normal distribution $E(1, D)$ if $r7$ 0.5 coincidental events, and D is the number of spatial dimensions. But if $r7$ is less than half, E will be equal to a random quantity that fits neatly into the normal distribution. After the exploitation phase is complete, the values of the candidate's fitness for the newly produced $GX(t+1)$ problem are computed. GX is preserved if $F(GX)F.(X)$ and optimize, while the optimal solution resides within each and every one of us.is shorthand for the hybrid Xsilverback, and it's a silverback. An example of GTO's pseudo code can be found in Algorithm.

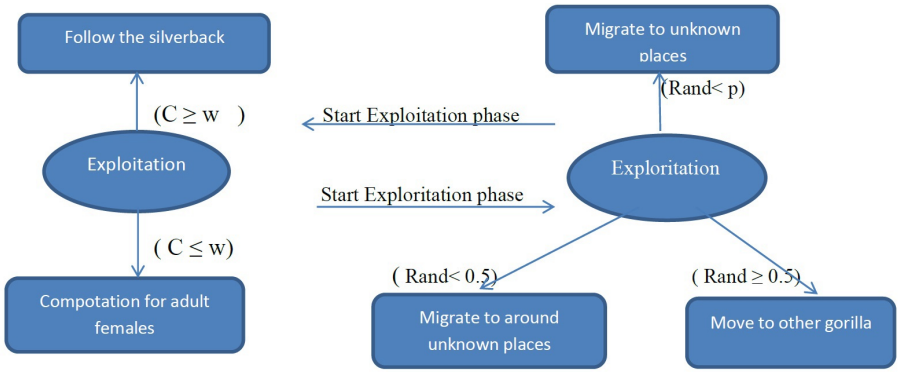

Figure 2: Gorilla Troops Optimizer Has Different Stages [1].

4 Artificial gorilla troops optimizer -based MLP trainer

While training an MLP with meta-heuristics, the representation of the problem is the first and most crucial step [5]. Therefore, it is important to frame the MLP training problem in a way that whether or not metaheuristics are acceptable. As introduced, the weights and biases are the most crucial settings when training an MLP. A trainer's task is to find the optimal combination of the weights and the biases that leads to the best classification, close approximation, and classification rate forecasting are all possibilities. Therefore, the weights and biases are the independent variables. Since the GTO algorithm expects the variables to be presented as a vector, the MLP's variables are presented in this format:

$$\overline{V} = \{\overline{W}, \overline{\theta}\} = \{W_{1,1}, W_{1,2}, ..., W_{n,h}, \theta_1, \theta_2, ..., \theta_h\} \tag{4.1}$$

W_{ij} j is the bias (threshold) of the $I - th$ hidden node, and n is the number of input nodes. Next, specify the GTO objective function. Training an MLP aims to increase its classification, approximation, or prediction accuracy. Sample training and testing. MLPs are evaluated using MSE. In this metric, the MLP is fed a predefined set of training samples, and the difference between the desired and actual output is measured using the following formula.

$$MSE = \sum_{i=1}^{m} (O_i^k - d_i^k)^2 \tag{4.2}$$

Algorithm 1 GTO

Populate Maximum iterations are N and Maxiter.
Randomly generate X_i $(I = 1, 2, ..., N)$ gorillas.
Determine each gorilla's fitness.
While $t <$ Maxiter do.
Equation. (2.3) Update C.
Calculate. Change L Equation. (2.4).
x_i Per gorilla/"Exploration gate"
Update gorilla's location using Equation (2.2)
End for
Assess gorilla fitness.
Save the best solution, silverback. Xsilverback
Do X_i for each gorilla/"Exploitation gate"
If $C > W$ then
Update gorilla's location using Equation (3.1)
Else
Update gorilla's location using Equation (3.3)
end "If"
End For
Update gorilla fitness values.
Global best solution update Xsilverback
t=t+1
End
Create the world's best product, Xsilverback, and improve its fitness.

where *dk I* is the output that should be produced by the *ith* input unit using the *kth* training sample, while O_i^k is the output that actually gets produced. where the value of m represents the number of outputs. The capacity of an MLP to generalize from examples provided during training is directly related to how well it performs. As a consequence of this, the performance of the MLP is evaluated by calculating the typical mean square error across all training samples:

$$\overrightarrow{MSE} = \sum_{k=1}^{s} \frac{\sum_{i=1}^{m} (O_i^k - d_i^k)^2}{S} \tag{4.3}$$

where s = training samples, m = outputs, dk I is the desired output of the ith input unit with the kth training sample, and ok I is the actual output. After all, the variable and average MSE for the GTO algorithm can be used to formulate the MLP training problem as follows: To reduce:

$$F(\overline{V}) = \text{Mean Squared Error} \tag{4.4}$$

Impact of biases and weights Assumed Training Samples Mean Squared Error Weights and biases are provided by GTO to MLP, and in return, GTO is given the overall average MSE for all training sample. Given that the weights and the biases tend to converge on the best MLPs that have been obtained to this point, there is a good chance that the MLP would get better with each iteration. The stochastic nature of GTO means that it cannot be relied upon to always return the best MLP for a given dataset. However, as the MLPs are evolved using the best MLPs obtained so far, the population-wide average MSE decreases over time. Basically, if you run the GTO algorithm enough times, it will converge on a solution that is superior to the initial solutions that were generated at random. In the following, we look into why the GTO algorithm is so effective when training on the MLP in real world..

5 Discussion and Results

In this subsection, five standard classification datasets from the (UCI) Machine Learning Repository [6] are used to train the proposed GTO-based MLP trainer (breast cancer, balloon iris, heart, XOR).

For this dataset, the MLP's output must be the Boolean XOR of the input. Table results show that GTO-MLP, GWO-MLP, and WOA-MLP all achieve a perfect classification rate 100%.

Table 1: The standard classification datasets

datasets	M.L.P. structure	Number of attributes
3-bitsXOR	3.7.1	3
Iris	4.9.3	4
Heart	22.45.1	22
Balloon	4.4.9	4
Breast cancer	9.19.1	9

Table 2: Results from experiments on the XOR dataset

Algorithm	Classification rate	MSE (AVE± STD)
GTO–MLP	100%	0.009410 ± 0.029500
GWO-MLP	100%	0.009410 ± 0.029500
WOA-MLP	100%	0.0006524 ± 0.00049
SCA–MLP	62.5%	0.118739 ± 0.011574

The Balloon dataset consists of two classes, eighteen training/test samples, and four attributes. This dataset's trainers have 55 dimensions. The findings are tabulated below. The table results demonstrate that across all algorithms, classification accuracy is 100%.

Table 3: Experimental research results from the balloon dataset

Algorithm	Classification rate	MSE (AVE± STD)
GTO–MLP	100%	0.009410 ± 0.029500
GWO-MLP	100%	$9.38e-15 \pm 2.81e-14$
WOA-MLP	100%	$4.61e-24 \pm 7.52e-23$
SCA–MLP	100%	0.000585 ± 0.000749

The Iris dataset contains three different classes, a total of 150 training and test samples, and four different attributes. As a result, the MLP structure for resolving this dataset is of the form 4-9-3, and there are seventy-five variables involved in the issue. The outcomes of training several distinct algorithms are laid out in Table below. In comparison to other algorithms, the results shown in Table demonstrate that the GTO-MLP algorithm achieves the highest classification rate of 92%.

Table 4: Experimental results for the iris-dataset

Algorithm	Classification rate	MSE (AVE± STD)
GTO–MLP	92%	0.0229 ± 0.0032
GWO-MLP	88%	0.089912 ± 0.123638
WOA-MLP	100%	0.009410 ± 0.029500
SCA–MLP	27%	0.084050 ± 0.035945

The breast cancer dataset contains two classes and nine attributes across 599 training samples. Trainers solve this dataset 10 times, and the aggregated results are tabulated below. The table results show that compared to other algorithms, GTO-MLP has the highest classification rate at 99%.

Table 5: Experiments done on breast cancer data

Algorithm	Classification rate	MSE (AVE± STD)
GTO–MLP	99%	$0.0012 \pm 7.4498e-05$
GWO-MLP	98%	0.003026 ± 0.001500
WOA-MLP	98%	0.003026 ± 0.001500
SCA–MLP	85%	0.089912 ± 0.123638

The algorithms' final success came with the heart dataset, a classification problem with twenty-two features, eighty training samples, one hundred eighty-seven test samples, and two classes. These algorithms are training MLPs with

a 22-45-1 structure. The tabulated findings are presented below. As can be seen in the table below, GTO-MLP algorithms achieve the highest classification rate of any algorithm tested, at 90%.

Table 6: Experiments done on breast cancer data

Algorithm	**Classification rate**	**MSE (AVE± STD)**
GTO–MLP	90%	0.0229 ± 0.0032
GWO-MLP	88.75%	0.089912 ± 0.123638
WOA-MLP	37.5%	0.084050 ± 0.035945
SCA–MLP	75%	0.122600 ± 0.007700

6 Conclusion

Recently, the GTO algorithm was proposed, and its first use as an MLP trainer can be found in this paper. This research was driven by a curiosity about the extensive capabilities. The extremely high degree of this algorithm for exploration and exploitation. In the beginning, there was the GTO algorithm. to be presented the difficulty involved in training an MLP. After determining the algorithm's optimal weights and biases, they put it to use. Five common classification datasets (XOR, balloon, Iris, breast cancer, and heart) datasets were used to test the proposed GTO-based trainer. Results from the GTO-MLP algorithms were compared to one another to those from the combination of five different stochastic optimizations simulators for confirmation. The outcomes demonstrated the effectiveness of the proposed approach in training MLPs. The GTO-MLP, for one, avoids local optima at a very high level, increasing the likelihood that good approximate values for the weights and biases that should be used in MLPs will be found. And because the GTO-MLP trainer is so well-utilized, the optimal weights and biases that are derived from it are extremely precise. Strong and poor performances of other algorithms were also identified, and their causes were discussed in this paper. It was found that the GTO algorithm is useful in determining the optimal values for MLP structural parameters like nodes and layers hidden. Further study into the finer points of this algorithm's tuning is warranted.

References

[1] B. Abdollahzadeh, F. Soleimanian Gharehchopogh and S. Mirjalili, *Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems*, Int. J. Intell. Syst. **36** (2021), no. 10, 5887–5958.

[2] I. Al-Badarneh, M. Habib, I. Aljarah and H. Faris, *Neuro-evolutionary models for imbalanced classification problems*, J. King Saud. Univ. Comput. Inf. Sci. 34 (2022), no 6, 2787–2797.

[3] I. Aljarah, H. Faris and S. Mirjalili, *Optimizing connection weights in neural networks using the whale optimization algorithm*, Soft Comput. **22** (2018), no. 1, 1–15.

[4] G. Bebis and M. Georgiopoulos, *Feed-forward neural networks*, Potentials, IEEE **13** (1994), 27–31.

[5] R.K. Belew, J. McInerney and N.N. Schraudolph, *Evolving networks: Using the genetic algorithm with connectionist learning*, Addison-Wesley, 1990.

[6] C. Blake and C.J. Merz, *UCI repository of machine learning databases*, University of California, Department of Information and Computer Science, 1998.

[7] A. Botalb, M. Moinuddin, U.M. Al-Saggaf and S.S.A. Ali, *Contrasting convolutional neural network (CNN) with multi-layer perceptron (MLP) for big data analysis*, Int. Conf. Intell. Adv. Syst. ICIAS **2018**, (2018), 1–5.

[8] F. Ecer, S. Ardabili, S.S. Band and A. Mosavi, *Training multilayer perceptron with genetic algorithms and particle swarm optimization for modeling stock price index prediction*, Entropy **22** (2020), no. 11, p. 1239.

[9] S. Ghosh-Dastidar and H. Adeli, *Spiking neural networks*, Int. J. Neural. Syst. **19** (2009), 295–308.

[10] J. Hamidzadeh, R. Monsefi and H. Sadoghi Yazdi, *DDC: Distance-based decision classifier*, Neural Comput. Appl. **21** (2012), no. 7, 1697–1707.

[11] A.A. Heidari, H. Faris, I. Aljarah and S. Mirjalili, *An efficient hybrid multilayer perceptron neural network with grasshopper optimization*, Soft Comput. **23** (2019), no. 17, 7941–7958.

[12] J. Hertz, A. Krogh and R.G. Palmer, *Introduction to the theory of neural computation*, Avalon Publishing, 1991.

[13] M. Hesami, R. Naderi and M. Tohidfar, *Modeling and optimizing in vitro sterilization of chrysanthemum via multilayer perceptron-non-dominated sorting genetic algorithm-II (MLP-NSGAII)*, Front. Plant Sci. **10** (2019), 1–13.

[14] T. Kohonen, The self-organizing map, Proc. IEEE **78** (1990), 1464–1480.

[15] S. Lee and J.Y. Choeh, *Predicting the helpfulness of online reviews using multilayer perceptron neural networks*, Expert Syst. Appl. **41** (2014), no. 6, 3041–3046.

[16] W.S. McCulloch and W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, Bull. Math. Biophys. **5** (1943), 115–133.

[17] S. Mirjalili, *How effective is the Grey Wolf optimizer in training multi-layer perceptrons*, Appl. Intell. **43** (2015), no. 1, 150–161.

[18] S. Mirjalili, S.M. Mirjalili and A. Lewis, *Grey wolf optimizer*, Adv. Eng. Softw. **69** (2014), 46–61.

[19] S. Mirjalili, S.M. Mirjalili and A. Lewis, *Let a biogeographybased optimizer train your multi-layer perceptron*, Info. Sci. **269** (2014), 188–209.

[20] J. Park and I.W. Sandberg, *Approximation and radial-basisfunction networks*, Neural Comput. **5** (1993), no. 2, 305–316.

[21] H. Ramchoun, M. Amine, J. Idrissi, Y. Ghanou and M. Ettaouil, *Multilayer perceptron: Architecture optimization and training*, Int. J. Interact. Multimed. Artif. Intell. **4** (2016), no. 1.

[22] S. Samadianfard, S. Hashemi, K. Kargar, M. Izadyar, A. Mostafaeipour, A. Mosavi, N. Nabipour and S. Shamshirband, *Wind speed prediction using a hybrid model of the multi-layer perceptron and whale optimization algorithm*, Energy Rep. **6** (2020), 1147–1159.

[23] Y. Xiao, X. Sun, Y. Guo, S. Li, Y. Zhang and Y. Wang, *An improved gorilla troops optimizer based on lens opposition-based learning and adaptive 13-hill climbing for global optimization*, CMES-Comput. Model. Eng. Sci. **131** (2022), no. 2, 815–850.