ADVANCED COMPUTING SESSION

MULTI-LAYER PERCEPTRON TRAINING BY GENETIC ALGORITHMS

Luka Gajić*,
Dušan Cvetnić,
Timea Bezdan,
Miodrag Živković,
Nebojša Bačanin

Faculty of Informatics and Computing,
Singidunum University,
Belgrade, Serbia

Abstract:
In this paper, the authors are presenting one of the ways to train Artificial Neural Networks (ANN) using a predefined set of weights as biases as input parameters, generated by the Genetic Algorithm (GA). This approach solves the problem of hyperparameter tuning for ANN, which is an NP-hard space search problem and will be further explained in the paper. The genetic algorithm generates a population of potential solutions in each iteration and then after a series of solutions variables modification (crossover, mutation, etc.) ranks them based on their fitness values. The algorithm itself is tested on a standard Multi-layer Perceptron (MLP) artificial neural network and results are similar compared to other techniques of training.

Keywords:
ANN, neural networks, training, genetic algorithm.

1. INTRODUCTION

Artificial neural networks (ANNs)[1]can learn and is used to overcome very complex problems in science or engineering[2]. ANNs have proven to be a good model and have been widely used in working with problems such as shape recognition, classification, clustering, and also prediction undertakings. For example, for medical purposes, researchers have used different types of ANNs to master troublesome classification and disease detection assignments [3], [4]. They likewise utilized ANNs to order biomedical data, for example, coronary heart disease and diabetes. The huge accomplishment of those techniques can be attributed to the ability of neural networks to process vast amounts of information during the training stage and to reduce the time needed for the diagnosis [5].

For each problem, it is necessary to do network training. Network training means adjusting weights and biases, which is why it is an NP-hard problem because the search space is too large.

Correspondence:
Luka Gajić

e-mail:
luka.gajic.17@singimail.rs

DOI: 10.15308/Sinteza-2020-301-306

Nature-inspired algorithms can be used to train the network. They are partitioned within two groups. The primary group comprises of Evolutionary Algorithms (EA) while the rest is known as Swarm intelligence algorithms (SI). The most famous representative of Evolutionary Algorithms is the Genetic Algorithm (GA).

The second group of algorithms consists of algorithms based on the observation of the social behavior of swarms of insects, flocks of birds, packs of animals. What all these animals have in common is that they do not show excessive intelligence as individuals, but when they join groups due to, for example, finding food, they show a high level of intelligence and that encouraged researchers to make mathematical models of these behaviors and apply them to real-life problems [6], [7], [8], [9], [10].

In this paper, we utilized an MLP neural network that we hybridized by a genetic algorithm with our selection model to get better results. When training and testing the network we used publicly available breast cancer datasets. The accuracy of such a system is between 96-98%. We compared our results with the results of the author of the paper in which the Swarm Intelligence algorithm was used together with the MLP neural network [11].

ANN training

The artificial neural network is a collection of nodes, also called artificial neurons, that mimic neurons in the organic mind. Every association, just as neurotransmitters in the mind, can send signals to other neurons. They receive these signals and process them. These signals are represented by real numbers and the yield of every neuron is determined based on some nonlinear function. Neurons and connections usually have their weights that are adjusted during network learning. During these adjustments, the neural network will begin to produce results that are very similar to the desired output. After a sufficient number of weight adjustments, the training can be stopped based on some criteria.

One of the most famous types of ANN models that can detect and estimate computational models utilizing their superior parallel layered structure is Feedforward neural networks (FFNNs) [12].

A multilayer perceptron (MLP) is a type of FFNNs. MLP comprise at least three layers that are sorted out in a one- directional mode. In the beginning, we have an input layer, at that point at least one hidden layers, and finally an output layer. Each of these layers consists from nonlinear-activation nodes. Each node in one layer connects with a certain weight with all other nodes in the next layer. MLP with a solitary hidden layer, as per Kolmogorov's theorem, is equipped for approximating different continuous functions [13]. Figure 1 shows an MLP network with only one hidden layer. The associations among the layers ought to be portrayed by certain weights that are situated within [−1, 1]. Every node in the MLP can perform two purposes: summation and activation. An activation function used in this paper is the S-shaped curved sigmoid function. This function is described as:

$$fj(x) = \frac{1}{1+e^{-sj}} \quad (1)$$

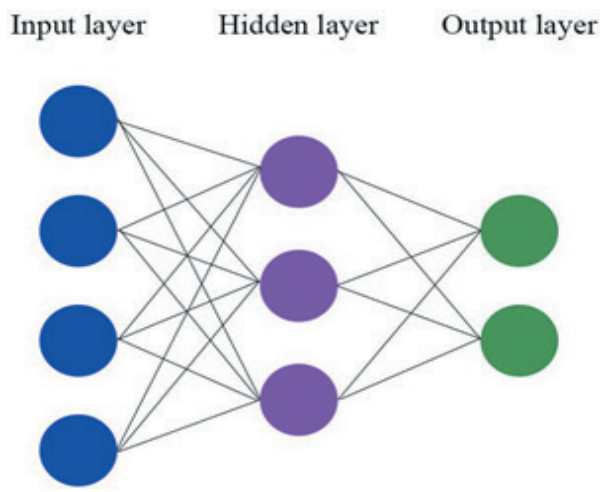

Figure 1. MLP neural network.

When talking about the optimization algorithm, the way we can assess a solution that can be a candidate (in the case of this paper we will be using a set of weights) is utilizing one type of a function that in literature us called the objective function. The objective function can be maximized or minimized, depending on the problem, implying that we search for a solution that has the best or the worst score. Typically, when using neural networks, we try to reduce the error. In essence, the objective function is often called loss function. In this paper, we used Mean Squared Error loss (MSE), which is computed as the average of the squared differences among prognosticated and present values. The result is invariably positive regardless of the sign of the prognosticated and present values. A mathematical perfect value is zero. If we have optimization for a maximizing problem, we can still minimize the loss value by making the score negative.

2. PROPOSED METHOD

Genetic algorithm (GA) [14] is a method propelled by the cycle of natural selection that has a place with the bigger class of evolutionary algorithms (EA). Genetic algorithms are regularly employed to create excellent solutions to optimization and search problems by depending on operators like selection, mutation, crossover which are driven by nature. In genetic algorithms, the population of candidate solution is evolving towards better solutions. Each member of the population has its own set of traits that can change and mutate. Evolution usually begins with the initial formation of a population composed of individuals that are randomly generated. Evolution is by its nature an iterative process and on account of genetic algorithms, a population in each iteration has named a generation. The fitness of each member of the group is calculated. In GA fitness is normally the value of the objective function in the issue to be unraveled. After that, members with greater fitness are chosen from the present population and their properties are combined in order for the offspring to inherit those properties.

Regarding the model of a selection of parents from the population of solutions, we decided on the selection according to the principle of the two best. In this type of selection, two parents are chosen who have the highest fitness values in the current population. This selection is good because the highest- quality specimens of the population are selected and reproduced to improve the genetic algorithm and its convergence.

A dose of mutation is also inserted in order to increase the uncertainty and in some cases to get better solutions. The mutation is achieved by giving a certain number in the range that serves as the certainty of whether the mutation will occur or not. In our case, we called that variable mutation rate. The higher the mutation rate, the more probable it is that the given solution will mutate. If a mutation occurs, the properties of the solution change randomly. The offspring obtained after one iteration changes the existing population and the next iteration begins. The genetic algorithm can be described by the following pseudo-code:

(1) Begin

(2) Initialize maximum number of generations

(3) Generate a random population of n solutions

(4) while(!maximum number of generations)

(5) Calculate fitness f(x) for each solution in the population

(6) while(!number of offspring created)

(7) select the pair of parents from the current population (roulette wheel)

(8) apply the crossover operator to the selected parents with the probability p and create two offspring

(9) Utilize mutation operator to the produced offspring with mutation rate as the chance for it to happen

(10) End while

(11) substitute the current population with the new

(12) End while

(13) End

In this part, the suggested GA-based MLP trainer (GAMLP) algorithm is portrayed in detail. As referenced previously, the single hidden layer MLP network is being trained with GA. Population members are represented as one-dimensional vectors of randomly generated real numbers within[1, 1].

Every solution depicts a candidate ANN. The proposed encoding vector includes three parts which are a set of bias terms and two collections of association weights between the layers. For the absolute quantity of biases and weights in the network, we can get the range of those vectors. A related encoding procedure is used for GAMLP. We must also pay attention to which functions that will represent fitness we will select. To successfully get results of solutions fitnesses, we need to send them to the MLP network as the association weights. The network can assess those numbers as indicated by a dataset we used for training. At long last, the network will acquire the fitness estimations of the analogous solutions. In this paper, the MSE is utilized as the loss function in the GAMLP trainer for evaluating networks fitnesses. As for the training units, the MSE metric can be gotten utilizing the fluctuation of the real and anticipated solutions by the produced members of population (MLPs). The most important thing is to reduce the value of the MSE however much as could reasonably be expected [11].

General steps for GAMLP trainer can be represented by:

(1) Initialization the GAMLP begins by making a random population

(2) Map the solutions the members of the GA population are attached to the weights and biases of a possible MLP network.

(3) Evaluation of the fitness the nature of the produced MLPs is assessed utilizing the MSE function for all samples in the training dataset.

(4) The GAMLP should discover the MLP with the most minimal MSE value. These MLPs with smaller MSEs are better than those with bigger MSEs.

(5) Selection and crossover operators of GA

(6) Iterate rounds 2-4 till the end

(7) Ending and reviewing the cycle is ended and MLP with the smallest MSE is being examined on the test dataset.

The general workings of the GAMLP system are exhibited in Figure 2.

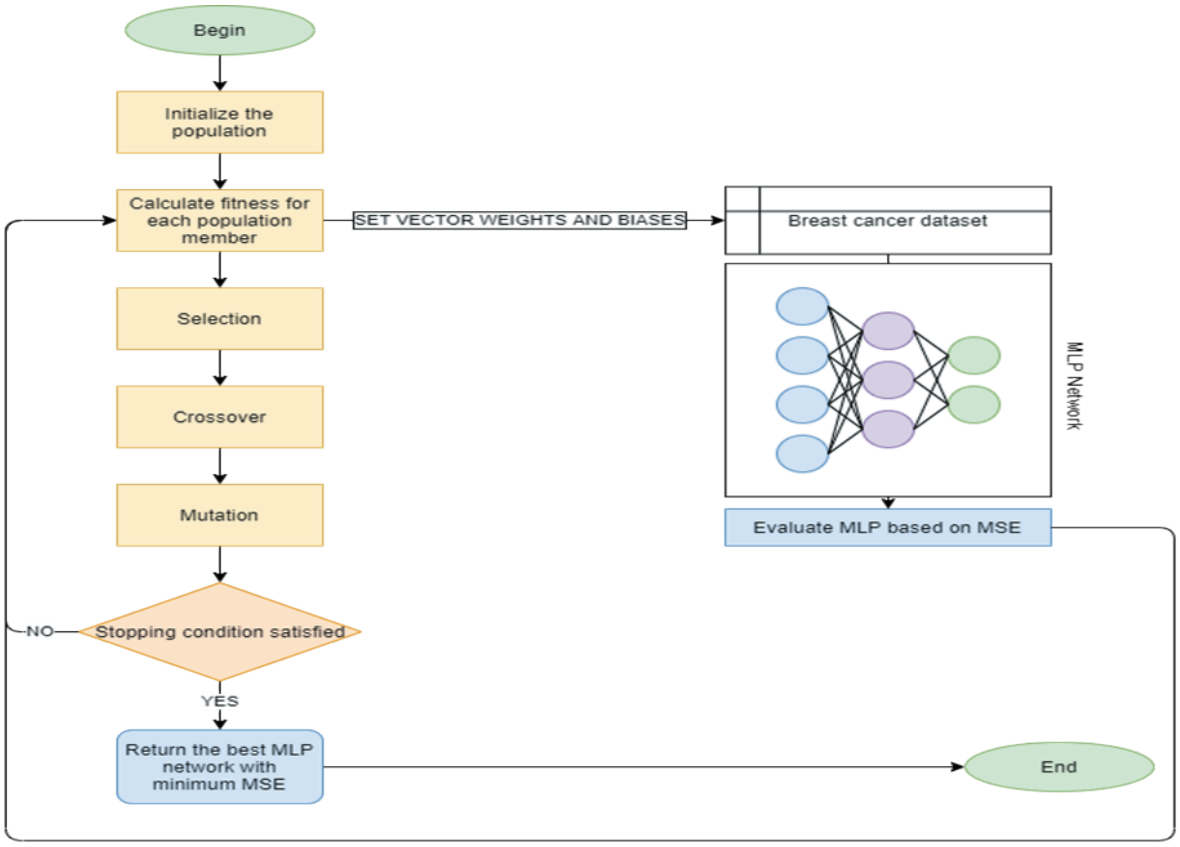

Figure 2. Optimizing MLP networkusing GA

3. EXPERIMENTS AND DISCUSSION

Publicly available breast cancer dataset was used during network training and evaluation of results. This set of information is initially gotten from Dr. William H. Wolberg, the University of Wisconsin Hospitals, Madison. This dataset consists of 699 cases where each case depicts a patient that had gone through the medical procedure for breast cancer. Four factors are estimated for every patient and marked as either malignant or benign [11][15]. To validate the findings of the advised GA-based trainer, it is correlated with a collection of known and established algorithms. The well- established algorithms are the PSO, bat algorithm (BAT/BA) [16], the firefly algorithm (FF/FA) [17], and artificial bee colony (ABC) [18] algorithm. In all trials, the size of the population and number of iterations are set to 5 and 100, respectively [11].The number of runs of the GAMLP is set to 10.

To assess the execution of the classification models that are developed in those analyses, we utilized accuracy. Accuracy of the rate of classification calculates the rate of the rightly classified specimens (either negative or positive) to the real absolute number of soecimens. The accuracy is determined by the accompanying condition [11]:

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (2)$$

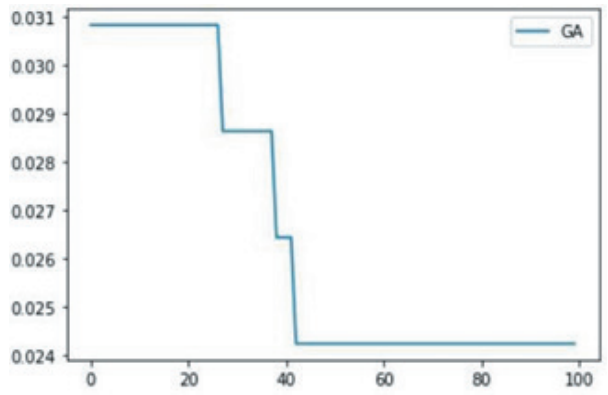

Figure 3. Convergence graph.

Figure 3 shows a graph of the convergence of the training system after the evaluation of the results.

Results of the evaluation of the intended GAMLP and other MHA-based MLP structures can be viewed in Table 1. The standard deviation, worst, best and average of each audit measure is and labeled as STD, WORST, BEST, and AVG. Giving the achieved outcomes, it very well may be seen that every algorithm can accomplish huge rates in all analyses [11].

Table 1. Breast cancer dataset results of the experiments of training MLP networks

	Metric	*Accuracy*
PSO	AVG	0.97045
	STD	0.00752
	BEST	0.97899
	WORST	0.95378
ABC	AVG	0.96891
	STD	0.00793
	BEST	0.98319
	WORST	0.94958
BAT	AVG	0.96218
	STD	0.01422
	BEST	0.98319
	WORST	0.92437
FA	AVG	0.97311
	STD	0.00324
	BEST	0.97899
	WORST	0.96639
GA	AVG	**0.95175**
	STD	**0.01195**
	BEST	**0.97807**
	WORST	**0.92982**

CONCLUSION

This paper used a well - known and exploited genetic algorithm to locate the ideal values of the MLPs parameters. Based on the training and testing of the entire training system on the breast cancer dataset, it was shown that such a system is as competitive and efficient as the others on which a comparative analysis was performed.

REFERENCES

[1] McCulloch, Warren S., and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." *The bulletin of mathematical biophysics* 5.4 (1943): 115-133.

[2] Heidari, Ali Asghar, and Rahim Ali Abbaspour. "Enhanced chaotic grey wolf optimizer for real-world optimization problems: A comparative study." *Handbook of Research on Emergent Applications of Optimization Algorithms*. IGI Global, 2018. 693-727.

[3] Esteva, Andre, et al. "Dermatologist-level classification of skin cancer with deep neural networks." *Nature* 542.7639 (2017): 115-118.

[4] Bezdan, Timea, et al. "Glioma Brain Tumor Grade Classification from MRI Using Convolutional Neural Networks Designed by Modified FA." I*nternational Conference on Intelligent and Fuzzy Systems*. Springer, Cham, 2020.

[5] Faris, Hossam, et al. "Optimizing the learning process of feedforward neural networks using lightning search algorithm." *International Journal on Artificial Intelligence Tools* 25.06 (2016): 1650033.

[6] Bezdan, Timea, et al. "Multi-objective Task Scheduling in Cloud Computing Environment by Hybridized Bat Algorithm." *International Conference on Intelligent and Fuzzy Systems. Springer*, Cham, 2020.

[7] Bacanin, Nebojsa, et al. "Whale Optimization Algorithm with Exploratory Move for Wireless Sensor Networks Localization." *International Conference on Hybrid Intelligent Systems. Springer*, Cham, 2019.

[8] Strumberger, Ivana, et al. "Dynamic search tree growth algorithm for global optimization." *Doctoral Conference on Computing, Electrical and Industrial Systems*. Springer, Cham, 2019.

[9] Zivkovic, Miodrag, et al. "Wireless Sensor Networks Life Time Optimization Based on the Improved Firefly Algorithm." 2020 *International Wireless Communications and Mobile Computing* (IWCMC). IEEE, 2020.

[10] Zivkovic, Miodrag, et al. "Enhanced Grey Wolf Algorithm for Energy Efficient Wireless Sensor Networks." 2020 *Zooming Innovation in Consumer Technologies Conference* (ZINC). IEEE, 2020.

[11] Heidari, Ali Asghar, et al. "An efficient hybrid multilayer perceptron neural network with grasshopper optimization." *Soft Computing* 23.17 (2019): 7941-7958.

[12] Ojha, Varun Kumar, Ajith Abraham, and Václav Snášel. "Metaheuristic design of feedforward neural networks: A review of two decades of research." *Engineering Applications of Artificial Intelligence 60* (2017): 97-116.

[13] Cybenko, George. "Approximation by superpositions of a sigmoidal function." *Mathematics of control, signals and systems* 2.4 (1989): 303- 314.

[14] Mitchell, Melanie. *An introduction to genetic algorithms*. MIT press, 1998.

[15] Wolberg, William H., and Olvi L. Mangasarian. "Multisurface method of pattern separation for medical diagnosis applied to breast cytology." *Proceedings of the national academy of sciences* 87.23 (1990): 9193-9196.

[16] Yang, Xin-She, and Amir Hossein Gandomi. "Bat algorithm: a novel approach for global engineering optimization." *Engineering computations* (2012).

[17] Yang, Xin-She. "Firefly algorithm, stochastic test functions and design optimisation." *International journal of bio-inspired computation* 2.2 (2010): 78-84.

[18] Karaboga, Dervis, Bahriye Akay, and Celal Ozturk. "Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks." *International conference on modeling decisions for artificial intelligence*. Springer, Berlin, Heidelberg, 2007.