International Journal on Electrical Engineering and Informatics - Volume 12, Number 1, March 2020

Optimization of Neural Networks Based on Modified Multi-Sonar Bat Units Algorithm

Mohammed Ali Tawfeeq

Computer Engineering Department, College of Engineering, Mustansiriyah University
Baghdad, Iraq
drmatawfeeq@uomustansiriyah.edu.iq

Abstract: The motivation behind this paper is to explore an algorithm that has the ability to optimize the free parameters required to design a neural network without being diligent in determining its values. The method uses modified version of Multi-Sonar Bat Units (MSBU) algorithm which is based on the idea that bats leave their roost shelter in groups and spread in space searching for their best food targets. The algorithm considers the population of the bats is equal to the training samples, each bat flies to the location of one training sample which represents its first target. After that, there will be frequent competitions between the bats to capture the next targets using their sonar beam. The bat got the highest association value with the target wins the competition. Each winning bat modifies the architecture of the designed network, and reflects its effect on the network weight values and on the positions of the remaining bats. The proposed method has been used to design several classification neural networks which are then utilized to diagnose various pathologies. Although some of the used training datasets are of nonlinear and heterogeneous nature, the designed networks using the proposed algorithm showed high performance outputs. The results obtained were compared with pattern recognition NNs. The comparison showed very promise output using the proposed algorithm.

Keywords: Artificial neural network, classification, Multi-sonar bat units, Pattern recognition network

1. Introduction

Artificial neural networks (ANN) are characterized by their ability to address complex nonlinear problems. This characteristic has made these networks of great interest allowing them to be used in applications that encompassed various areas of life. Although there are many algorithms already used for training and learning neural networks to adjust their weights, but the biggest problem remains in two aspects. The first one is how to determine or choose the optimum network architecture, which is often done by trial and error, while the second one is pointed to the long time required to train the networks. Over the past years, many attempts with different approaches have been done to solve these dilemmas. In general these approaches can be categorized according to their principles to: brute- force, pruning, regularization, probability optimization, sensitivity analysis, and network construction techniques [1]. Metaheuristic optimization techniques can be considered as another category in this field, such as: ant colony optimization, particle swarm optimization, genetic algorithms, etc.

Several researches have been proposed to optimize neural networks using such techniques, among these; Ghosh et al. [2] compared the results of using Backpropagation (BP), Simulated Annealing (SA), and Genetic Algorithm (GA) in optimizing NNs. They concluded that although SA algorithm performs better than BP, but both algorithms required some parameters to be determined by user in which it may significantly affects the solution, meanwhile GA can obtain superior solutions for optimizing NNs. Zhao et al. [3] used Artificial Bee Colony (ABC) algorithm with backpropagation neural network to predict rolling force in aluminum hot tandem rolling. The ABC algorithm is used to optimize the architecture of the neural network. Their results showed that the output of the optimized network is close to the practical values. Nimbark

Received: October 16th, 2018. Accepted: March 18th, 2020
DOI: 10.15676/ijeei.2020.12.1.9

et al. [4] optimized weights and transfer function of the neurons in ANN by using modified Artificial Bee colony algorithm (ABC). In this algorithm, the initialization is based on utilizing opposition-chaos method with balanced exploitation and exploration abilities. Idrissi, et al. [5] treated the problem of optimizing the architecture of neural network layers, connections, and weights in MLP by proposing a multi-objective formula which is extended later by introducing a regularized objective function [6]. The authors use GA and backpropagation techniques to solve their mathematical model. Their methodology has been tested using classification data, such as iris, wine, seed, and medical data. Kaviani and MirRokni [7] also used GA to obtain the optimum number of hidden neurons and weights in a neural network which is utilized in predicting average daily temperature. They showed that the GA approach can replace the trial and error methods in determining optimal state of ANN architecture. Melin and Sánchez [8] proposed Multi-objective Hierarchical GA for modular NN optimization using a granular approach. In this approach, different NN parameters are performed, such as: size of the training dataset, number of sub-granules, number of hidden layers with their neurons, and the goal error. Chhachhiya et al. [9] solve the problem of finding optimum architecture of neural network by using hybrid of Particle Swarm Optimization (PSO) technique with back propagation algorithm. The parameters considered to optimize the network are the number of hidden neurons, learning rate, and activation function, while the applied fitness function is RMS error. KAMAL and KODAZ [10] used the bat algorithm which is proposed by Yang [11] as a learning algorithm to optimize the weight values of neural networks. The algorithm is based on that each bat has its position, velocity, frequency, wavelength, loudness value, and pulse emission. The developed NNs were trained and tested for classification purposes. The recorded classification accuracy for different datasets were between 56.44 up to 96.26. Kusy and Kowalski [12, 13] applied sensitivity analysis (SA) procedure in probabilistic neural networks (PNNs) to calculate the weights between the pattern and summation layer, whereas Kowalski and Kusy [14] used the SA procedure to reduce the input layer units in PNNs by removing some features from the training dataset which leads to reduction in the network structure.

This work introduces a neural network optimization technique using a metaheuristic method based on modified Multi-Sonar Bat Units (MSBU) algorithm. MSBU deals with the principle of network construction technique by incrementally increasing the number of the hidden neurons. The next section introduces the details of the proposed algorithm, while section 3 describes the used datasets. In section 4 the results and the performance evaluation of the proposed algorithm is presented. Last section introduces the conclusion of this work.

2. Modified Multi-Sonar Bat Units Algorithm

Bat MSU algorithm in its original form [15] is used to find the optimum solution of a problem in which multi bats (multi-sonar bat units) spread randomly in the state space of the problem. Each bat emits sonar signals using its ecolocating system and fly towards the discovered prey to capture it. Bats continues searching for best target (solution) and keeps flying toward these targets (unless the positions where they were are better) until it settled on the optimum solution. In this work the MSU algorithm is used with some modification to accommodate the requirements of determining the optimum architecture and weights of neural networks. The state space of the problem is assumed to be the Euclidean space R^n, where n is the number of input attributes to the neural network. The training data set vector pairs p:t are of Q length. The input vectors p are represented as points in the Euclidean space R^n. The neural networks to be optimized have the architecture n-H-m, where n is the number of the input source node, H is the number of neurons in the hidden layer, and m is the number of output neurons which can be obtained from the target dataset. The activation function of the hidden layer is considered to be the hyperbolic tangent, while for the output layer is the identity function. The motivation behind using the hyperbolic tangent as an activation function for the hidden neurons is that it combines the characteristics of several activation functions, in which it behaves in its mid part approximately like the linear function that has the ability of storing the pattern perfectly, and in its lower part it conducts like faster-than-linear activation function that is suitable to deal with small signals, while in its upper part it acts just like slower-than-linear function which can perform large signals. In other words; it has the ability to enhance contrast, attenuates small signals, while the larger values are amplified. This helps in minimizing the effects of noise signals. In addition this function covers the range from -1 to+1, which gives freedom to deal extensively with various weights values.

The problem is to determine the optimum values of the weights and biases of the networks in addition to the number of the hidden neurons *H* that can classify each input perfectly.

In nature bats leave their roost shelter in groups to start their hunting period. Each bat has its echo sonar unit, in other words, group of bats have Multi Sonar Units MSU. In this work the population of bats is considered to be of *Q* size, i.e., the size of MSU is equal the number of the training input vectors *Q*. The Modified MSBU algorithm can be described as follows:

- The bats leaves their roost in a population of size *Q*.
- Each bat navigates for a prey using its sonar unit, flying towards it. The positions of the preys are assumed to be the end point x_i of the length of each input training vector p_i starting from the roost and spread in the Euclidean space. As a result, each bat is settled on one end point of the p's vectors. The locations of the bats are considered as the initial position. Figure-1 illustrates the concepts of this step.
-

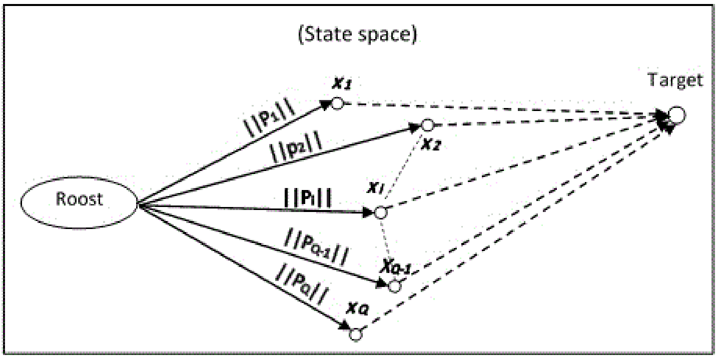


Figure-1 Deployment of bats in the state space

- In the next step, the bats start a competition to navigate a new prey located at the target position and fly towards it. The bat with largest associated value between its position x_i and the target t is assumed to be the winner. This is just like what happen in food competition between big brown bats in nature, in which these bats produce an ultrasonic signal that effectively jamming the echolocation signals of other competitors' bats to make them miss the target, the bat with the most high power signal wins the competition [16, 17].
- The winner bat puts its impression on the neural network and stimulates the other MSU to change their position. This mimics the behavior of big brown bats in which the jamming signal does not cause bats to leave away from prey, instead they change their position and wrap around the site of foraging area [18].
- The remaining bats continues the competition with new iteration, the new winner bat repeats what the former winner did on the NN. This process is continued until the NN reaches its optimum architecture and weights.
- The competition is settled either when the network converged beyond the allowed mean square error or it reaches the maximum pre-specified number of neurons.
- The algorithm can be repeated for several times, the neural network with best outputs is picked as the final one.

In more details the MSBU algorithm can be expressed as follows:

1. Define the main parameters:
   Maximum allowed Mean Squared Error MSE, maximum number of hidden neurons *H*, and

Max number of Iterations *Max_It*
2. While iteration *Iter* < *Max_it* do step 3-17
3. Select randomly 70% of the dataset as training set pairs p:t (Q pair of vectors), and 30% as test data. Where $Q >> H$.
4. Deployed bats of Q population in the state space and let them settled in their initial position points x_i, where $x_i = \|p_i\|$, (roost is assumed to be the point of origin).
5. While ($h \leq H$ or performance>MSE) repeat steps 6-15 ((where h is a counter for temporary number of hidden neurons initiated with 0 value)
6. Calculate the Associated Values (Av) between the bat positions x and the targets t
$$Av = \frac{\sum(x_i t_i)^2}{\sum x_i^2 \sum t_i^2} \tag{1}$$
7. Determine the winner bat:
$$x_{win} = \max|Av| \tag{2}$$
8. Store the value of x_{win} in matrix W_{win} and its sequence in a list array of the winners L_{win}.
9. Add a neuron to the hidden layer and calculate the weights between it and the input source nodes by considering the input vector of the winner bat p_{win} as the output of this neuron, taking in consideration that the activation function of this layer is the hyperbolic tangent (hint: the algorithm assumes the hidden layer biases is a constant small value $b \leq 1$):
$$S = S + 1 \tag{3}$$
$$\boldsymbol{w^1} = \tanh^{-1}(\boldsymbol{p}(L_{win})) \tag{4}$$
Where w^1 is the weights of layer 1.
10. Calculate the output of layer 1:
$$\boldsymbol{a^1} = \tanh(w^1 p + b^1) \tag{5}$$
11. Since the activation function of the output is linear, thus computing the weights and the biases of layer 2 can be easily done. The input to this layer is a^1 concatenated with 1's (value of the inputs to the biases).
$$\boldsymbol{w^2} = \boldsymbol{t}/\boldsymbol{a^1} \tag{6}$$
The last column in w^2 represents the values of the biases b^2 of this layer.
12. Remove the winner bat from the population (remove x_{win} from the Q population data).
13. Evaluate the stimulation factor *St* of the winning bats on the other MSBU:
$$St = \frac{\sum(Av_i\, x_i)^2}{\sum Av_i^2} \tag{7}$$
14. Calculate the new positions of the remaining MSBU due to the stimulation of the winner ones:
$$x(new) = x(old) - Av\, St \tag{8}$$
15. Compute the total mean squared error:
$$\boldsymbol{a^2} = \boldsymbol{w^2 a^1} + \boldsymbol{b^2} \tag{9}$$
$$mse = \sum(\boldsymbol{t} - \boldsymbol{a^2})^2 \tag{10}$$
where a^2 is the output of the network.
16. Save the network with its parameters
17. Increase the number of iteration *Iter* by 1
18. Calculate the performance of the saved networks
19. Select the network with best performance and optimum number of hidden neurons

Note that, the randomness selection of training dataset in each iteration leads to different network parameters.

3. Dataset Selection

Different datasets have been selected to train and test several neural networks using the proposed algorithm. All the data sets are downloaded from the "University of California, Irvine UCI Machine Learning Repository" [19]. These datasets are: (i) Breast cancer Wisconsin, (ii) Acute inflammations, (iii) Dermatology, (iv) Hepatitis, (v) SPECT heart (i.e. diagnosis of cardiac Single Proton Emission Computed Tomography images). The associated task of the five selected datasets is "classification". The datasets are preprocessed by eliminating the missing values records, converting some categorical attributes to numeric values in addition to that, the datasets are normalized.

4. Results and Performance Evaluation

In this work, several classification neural networks have been designed using MSBU algorithm. In order to check the generalization of these networks they were all subjected to a different measurements and performance evaluations. In addition, the outputs of these networks are compared with the results obtained from equivalent architecture feedforward networks which are specialist in pattern recognition classifications problems using the same training datasets. The pattern classification (PR) networks are created by calling the built-in function "patternnet" in MATLAB 2017b, and trained by using scaled conjugate gradient backpropagation training algorithm and categorical Cross-Entropy loss function (CE) to calculate the performance.

The following testing measurements and metrics are used to check the classification quality of each network:

- Performance (MSE and CE), mean of the error, and the error standard deviation.
- linear regression of the testing data targets relative to the network outputs
- Classification confusion matrix
- The receiver operating characteristic (ROC) which shows how well the neural network fits the testing data

Two samples of graphical representation for these metrics are illustrated in the following figures.

For Breast Cancer Wisconsin the dataset is of 683 instances, 9 attributes, and 2 classes. The architecture of the best determined NN for this problem is of 9-4-2, in which, the recorded classification accuracy using the test data is 99%. The upper part of Figure-2 shows sample index of the errors, while the lower part of Figure-2 shows a histogram of error in classification and fits a normal density function. As it is indicated in this figure, the MSE is 0.0097, the error mean is -0.0097, and the error standard deviation is about 0.098.

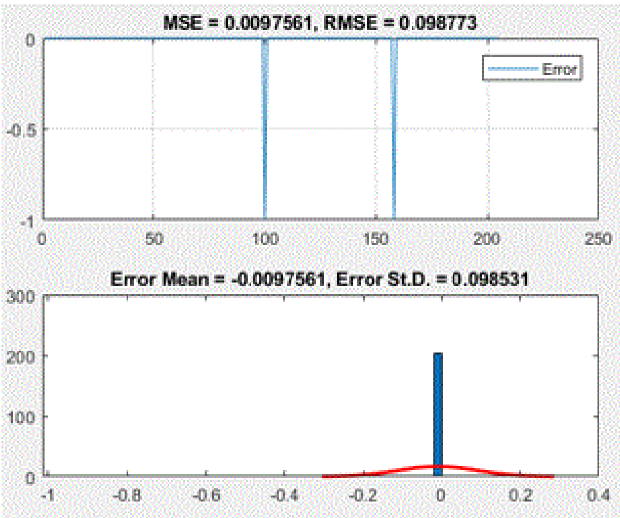


Figure 2. Classification errors in Breast Cancer Wisconsin NN

The linear regression of the targets relative to the outputs is shown in Figure-3. The calculated correlation coefficient is about 0.978.

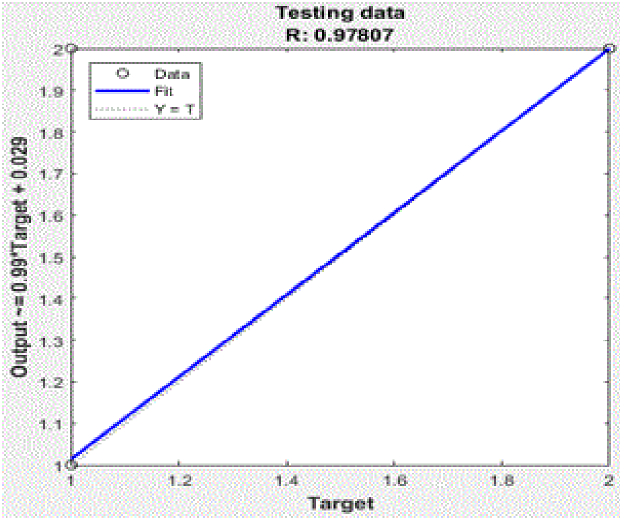


Figure 3. Regression of Breast Cancer Wisconsin NN

The left part of Figure-4 shows the classification confusion matrix using MSBU algorithm, while the right part shows the confusion matrix of the PR net trained using backpropagation algorithm. Knowing that these two NNs have the same architecture and trained using the same dataset. Comparison between the two NNs shows that the classification accuracy using MSBU bat NN is about 99%, while it is 98.5 in PR net.

MSBU

Output Class	Target Class 1	Target Class 2	
1	**137** 66.8%	**0** 0.0%	**100%** 0.0%
2	**2** 1.0%	**66** 32.2%	**97.1%** 2.9%
	98.6% 1.4%	**100%** 1.0%	**99.0%** 1.0%

PR Net

Output Class	Target Class 1	Target Class 2	
1	**138** 67.3%	**2** 1.0%	**98.6%** 0.0%
2	**1** 0.5%	**64** 31.2%	**98.5%** 1.5%
	99.3% 0.7%	**97.0%** 3.0%	**98.5%** 1.5%

Figure 4. Confusion matrix of Breast Cancer Wisconsin networks

On the other hand, the ROC curves for both networks are plotted as shown in Figure-5. The ROC curves shows that the NN of MSBU is much more fits the testing data.

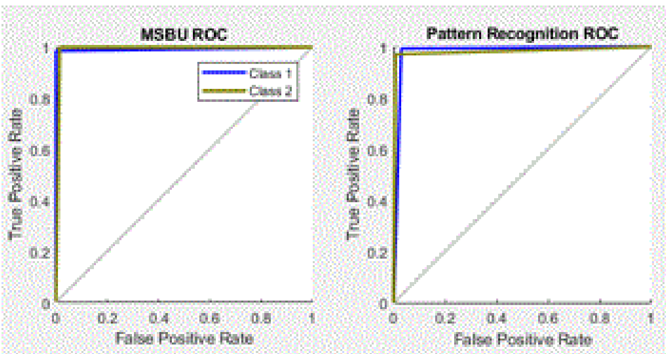


Figure 5. ROC of Breast Cancer Wisconsin networks

For acute inflammations the used dataset is of 120 instances, 6 attributes, and 4 classes. The optimum architecture of the network found by using the proposed algorithm is of 6-4-4. This network achieved classification accuracy of 100%. The MSE is equal to zero as well as the error standard deviation with no sample index of errors as shown in Figure 6.

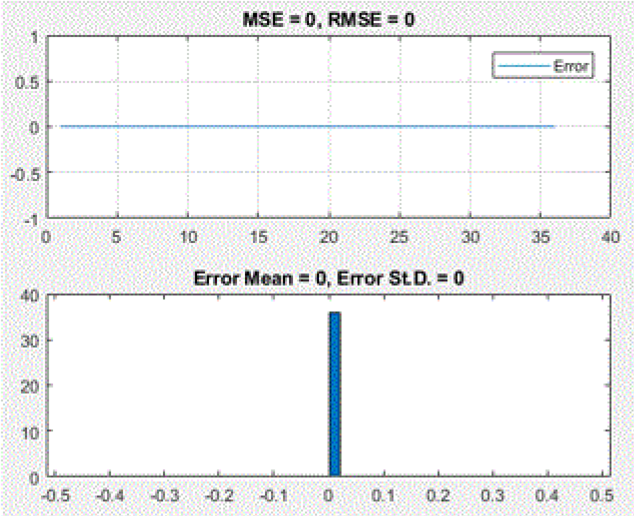


Figure 6. Classification errors in acute inflammations NN

The regression line of the targets vs outputs is almost perfect as shown in Figure 7, with correlation coefficient R =1.

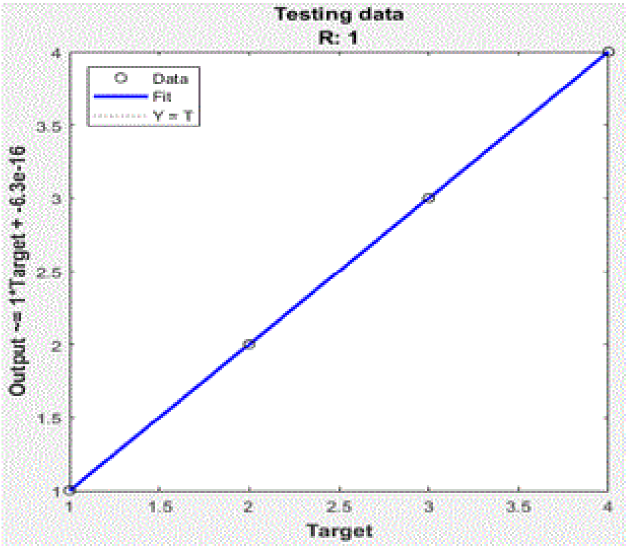
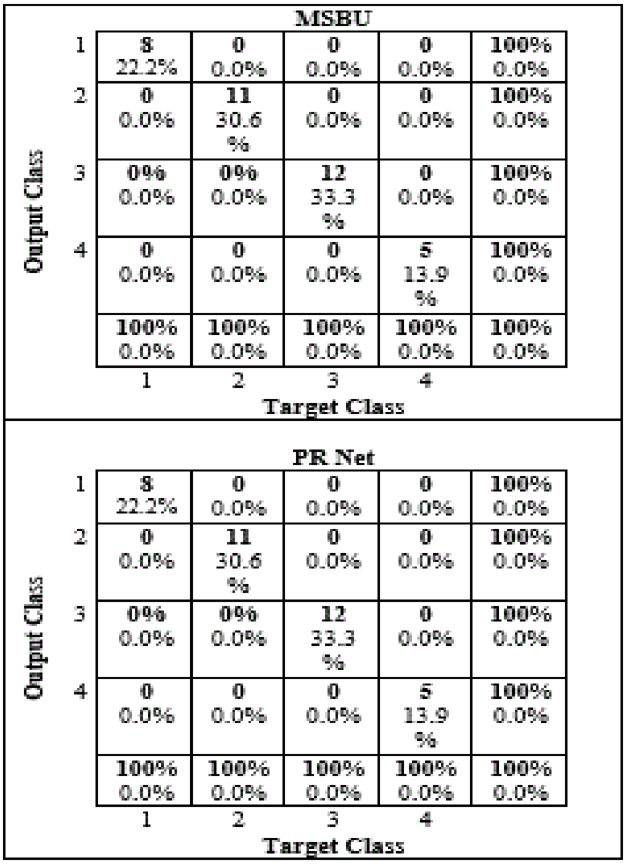


Figure 7. Regression of acute inflammations network

The comparison between the classification results using the optimum NN of the proposed algorithm and PR net can be illustrated using the confusion matrices shown in Figure 8.



MSBU

Output Class	1	2	3	4	
1	8 22.2%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	11 30.6 %	0 0.0%	0 0.0%	100% 0.0%
3	0% 0.0%	0% 0.0%	12 33.3 %	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	5 13.9 %	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%

PR Net

Output Class	1	2	3	4	
1	8 22.2%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	11 30.6 %	0 0.0%	0 0.0%	100% 0.0%
3	0% 0.0%	0% 0.0%	12 33.3 %	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	5 13.9 %	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%

Figure 8. Confusion matrix of acute inflammations networks

The ROC curves for the two networks fit perfectly the testing data as shown in Figure-9. The information of the other dataset cases and their networks classification accuracy results with the early mentioned ones are summarized in Table-1.

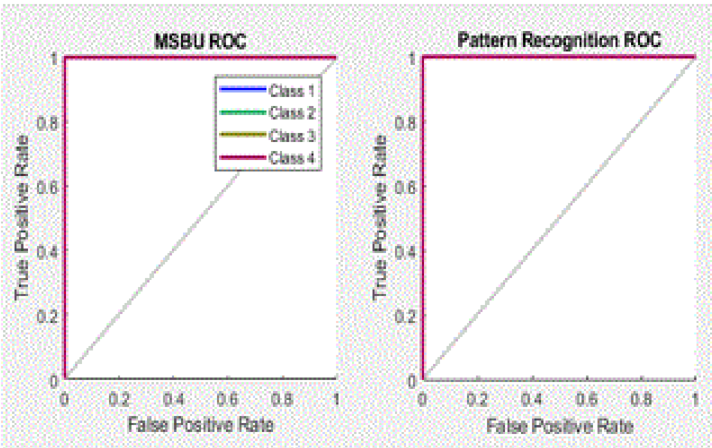


Figure 9. ROC of acute inflammations networks

Table 1. Datasets information with their networks' classification accuracy results

Case	No. of instances	No. of attributes	No. of classes	Best Net architecture	Correct classification	
					MSBU net	PR net
Breast Cancer Wisconsin	683	9	2	9-4-2	99%	98.5%
Acute inflammations	120	6	4	6-4-4	100%	100%
Dermatology	358	34	6	34-30-6	96.3%	97.2%
Hepatitis	148	18	2	18-15-2	97.7%	81.8%
SPECT heart	267	22	2	22-14-2	91.3%	86.3%

It is noticed experimentally that the performance of both types of the networks (MSBU and PR networks) are sensitive to the training dataset. Since the training datasets are randomly selected each time, this leads to different networks architectures and different magnitude of parameters which in turn leads to different performance. Table 2 depicts the others calculated metrics for the networks mentioned in Table 1.

Table 2. Performance and error measurements of the best MSBU networks

Case	MSBU				PR
	Error mean	Error St. D	R	MSE	CE
Breast Cancer Wisconsin	-0.009	0.098	0.978	0.009	0.031
Acute inflammations	0	0	1	0	8.04 e-7
Dermatology	0.028	0.444	0.959	0.196	0.006
Hepatitis	0.022	0.150	0.855	0.022	0.068
SPECT heart	-0.062	0.290	0.613	0.087	0.066

Table 3. Classification accuracy of randomly selected MSBU networks comparing with their corresponding PR net results

Iteration	Breast Cancer Wisconsin ($H = 4$)		Acute Inflammations ($H = 5$)		Dermatology ($H = 30$)		Hepatitis ($H = 15$)		SPECT heart ($H = 14$)	
	MSBU net %	PR net %	MSBU net %	PR net %	MSBU net %	PR net %	MSBU net %	PR net %	MSBU net %	PR net %
1	97.1	96.6	97.2	100	92.5	99.1	93.2	86.4	81.3	75
2	94.6	95.1	100	100	94.4	99.0	86.4	77.3	91.3	86.3
3	98.5	96.1	100	100	88.9	98.1	90.9	77.3	80.0	82.5
4	96.6	96.6	97.2	100	90.7	96.3	95.5	84.1	83.8	82.5
5	97.6	96.1	89.9	89.9	90.7	95.3	84.1	79.5	82.5	82.5
6	97.1	98.5	100	100	96.3	97.2	90.9	88.6	85.0	77.5
7	97.6	97.6	100	100	88.8	94.4	75.0	68.2	80.0	77.5
8	99.0	98.5	97.2	100	90.7	98.1	97.7	81.8	85.0	83.8
9	95.1	94.6	100	100	91.6	100	88.6	88.6	86.3	83.8
10	98.5	98.5	100	100	90.7	99.1	88.6	77.3	86.3	81.3
average	97.17	96.82	98.15	98.99	91.35	97.66	89.09	80.9	84.15	81.3

As it is described earlier, MSBU algorithm looks for and selects the best network performance and architecture, however, the results of ten random iterations are recorded and compared with the corresponding pattern recognition networks' results as shown in Table 3.

The average results for the networks of both MSBU and PR algorithms showed that the classification accuracy for breast cancer Wisconsin and acute inflammations are almost the same, meanwhile, the MSBU networks for Hepatitis and SPECT heart showed better accuracy. On the other side, the dermatology's PR networks record better average classification results.

The average network training times for both algorithms are measured and tabulated as shown in Table-4. The ratio of these training times are calculated by:

$$R_t = \frac{MSBU\ average\ training\ time}{PR\ average\ training\ time} \times 100\% \qquad (11)$$

The values of R_t showed that the required training time using MSBU algorithm is much less than that of the PR algorithm.

Table 4. Average training time

Case	No. of training vectors	Arch.	Average training time (sec)		*Rt*
			MSBU net	PR net	
Breast Cancer Wisconsin	478	9-4-2	0.128	0.400	32%
Acute inflammations	84	6-5-4	0.108	0.415	26%
Dermatology	251	34-30-6	0.270	0.471	57%
Hepatitis	104	18-15-2	0.159	0.393	40%
SPECT heart	187	22-14-2	0.183	0.398	46%

In PR network, the categorical cross-entropy loss between network actual response and target values was used to measure the performance. Using this function leads to trounce the outputs that are extremely inaccurate which in turn gives good classification results. Whereas, the proposed algorithm depends on the computation of MSE for performance evaluation. The two functions seem to be good at assessing networks performances. On the other hand, in PR networks, the weight and bias values were updated using the scaled conjugate gradient function. For the network to improve performance and therefore learn based on this function it requires iterative training process that takes all the training vectors in consideration and updated the network weights and biases each time. In other words, it requires to several number of epochs. Meanwhile in the proposed algorithm, the winning bat adds one neuron at a time, and modify the weight values of the network. Thus, the train of MSBU network requires a fraction of the time it takes to train PR network.

5. Conclusion

This work introduces an algorithm optimizes the architecture and weights of NNs based on the principles of bats competition for food in the wild. Each winner bat adds a neuron to the hidden layer and modify the weights of the network. The competition between the remaining bats is continued until the performance of the network reaches to the desired MSE or the network reached to the predetermined maximum number of neurons.

The algorithm was adopted in optimizing neural networks that were used for classification purposes. These networks showed promising results. The obtained results were compared with the output of standard neural networks trained by using backpropagation algorithm, the results of the two types of networks were fairly close to each other in most cases, and whereas the time required to design networks using the proposed algorithm is much less than the time required to train networks using standard methods.

6. Acknowledgements

The author would like to thank UCI Machine Learning Repository for providing the dataset used in this work.

7. References

[1]. Chhachhiya D, Sharma A, Gupta M, "Designing optimal architecture of neural network with particle swarm optimization techniques specifically for educational dataset", In: *Proceedings. 7th Int. Conf. on cloud computing, data science and engineering—confluence, IEEE, Noida*, 2017, pp. 52–57.
[2]. Corcoran1 A J, Conner W E, "Bats jamming bats: Food competition through sonar interference", Science 346(6210), 2014, pp. 745-747. doi: 10.1126/science.1259512.
[3]. Ghosh R, Ghosh M, Yearwood J, Bagirov A, "Comparative analysis of genetic algorithm, simulated annealing and cutting angle method for artificial neural networks", In: *Proceedings. 4th Int. Conf., MLDM 2005, Leipzig, Germany, July, 2005. Proceedings*, pp. 62–70.
[4]. Idrissi M A J, Ramchoun H, Ghanou Y, Ettaouil M, "Genetic algorithm for neural network architecture optimization", In: *IEEE Proceeding of the 3rd Int. Conf. of Logistics Operations Management*, GOL Morocco, May 2016, pp. 23-25.
[5]. KAMAL L L, KODAZ H, "Training Artificial Neural Network by Bat Optimization Algorithms", *International Journal of Advanced Computational Engineering and Networking*, vol. 5, no. 8, 2017, pp. 53-56.
[6]. Kaviani M, MirRokni S M, "Applying Genetic Algorithm in Architecture and Neural Network Training", *International Journal of Computer Science and Network Security*, Vol.17 No.6, June 2017.
[7]. Kowalski P A, Kusy M, "Determining significance of input neurons for probabilistic neural network by sensitivity analysis procedure", *Computational Intelligence*, Wiley Online Librar. https://doi.org/10.1111/coin.12149, 2017.
[8]. Kusy M, Kowalski P A, "Modification of the probabilistic neural network with the use of sensitivity analysis procedure", In: *Federated Conference on Computer Science and Information Systems (FedCSIS), Poland. IEEE*, 2016, pp. 97-103.
[9]. Kusy M, Kowalski P A, "Weighted probabilistic neural network", Information Sciences, ELSEVIER, Vol. 430-431, 2018, pp. 65-76.
[10]. Lichman, M, "UCI Machine Learning Repository", [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, *School of Information and Computer Science*, 2013.
[11]. Melin p, Sánchez D, "Multi-objective optimization for modular granular neural networks applied to pattern recognition", *Information Sciences, ELSEVIER*, Vol. 460–461, Sep. 2018, pp. 594-610.
[12]. Merzendorfer H, "Bat's jamming sonar tricks echolocation in rivals", *Journal of Experimental Biology*, Vol. 218, Issue 3, 2015, pp. 332-332. DOI: 10.1242/jeb.112052.
[13]. Nimbark H, Sukhadia R, Kotak P P, "Optimizing architectural properties of Artificial Neural Network using proposed Artificial Bee Colony algorithm", *Int. Conf. on Advances in Computing, Communications and Informatics, ICACCI2014*, Delhi, India, September, 2014. IEEE.
[14]. Ramchoun H, Idrissi M A J, Ghanou Y, Ettaouil M, "New Modeling of Multilayer Perceptron Architecture Optimization with Regularization: An Application to Pattern Classification", *International Journal of Computer Science*, Vol. 44, Issue 3, 2017, pp. 261-269.
[15]. Reitermanov´a Z, "Feedforward Neural Networks – Architecture Optimization and Knowledge Extraction", *WDS'08 Proceedings of Contributed Papers*, Part I, 2008, pp. 159–164.
[16]. Tawfeeq M A, "Intelligent Algorithm for Optimum Solutions Based on the Principles of Bat Sonar", *International Journal of Computer Science and Information security*, Vol. 10, No. 10, pp. 11-19.
[17]. Wright G S, Chiu C, Xian W, Wilkinson G S, Moss C F, "Social calls predict foraging success in big brown bats", Curr Biol, Vol. 24, No. 8, 2014, pp.885–889. DOI: 10.1016/j.cub.2014.02.058.

[18]. Yang X S, "A New Metaheuristic Bat-Inspired Algorithm", *Nature Inspired Cooperative Strategies for Optimization (NICSO)*, 2010, pp. 65-74.
[19]. Zhao Z, Yang J, Che H, Sun H, Yang H 2013, "Application of artificial bee colony algorithm to select architecture of an optimal neural network for the prediction of rolling force in hot strip rolling process", *Journal of Chem. Pharm. Res*., Vol. 5, No. 9, 2013, pp. 563-570.

Mohammed Ali Tawfeeq is an associated Professor at Computer Engineering Department, College of Engineering, Mustansiriyah University. He received his B.Sc. in Electrical Engineering from University of Technology/Baghdad, Iraq in 1979 and his M.Sc. degree in Computer Engineering from University of Baghdad in 1989. He obtained his Ph.D. in Computer Engineering from University of Technology, Baghdad, Iraq in 2006. His research interests include, but not limited to, Intelligent Systems, Optimization Techniques, Wireless Sensor Networks, and IoT applications.