A proactive metaheuristic model for optimizing weights of artificial neural network

Ali Hakem Alsaeedi[1], Ali Hussein Aljanabi[2], Mehdi Ebady Manna[3], Adil L. Albukhnefis[4]
[1,4]College of Computer Science and Information Technology, University of Al-Qadisiyah, Iraq
[2]Al-Qadisiyah Education Directorate, Iraq
[3]College of Information Technology, University of Babylon, Iraq

Article Info

Article history:

Received Jan 14, 2020
Revised Mar 17, 2020
Accepted Mar 31, 2020

Keywords:

Artificial neural network
Backpropagation algorithm
Data mining
Particle swarm optimization
Supervising learning

ABSTRACT

This paper proposes the Particle Swarm Optimization model for enhancing the performance of an Artificial Neural Network. The learning process of Artificial Neural Network requires a long time to satisfy requirements because of processing complexity of the backpropagation algorithm that has been used in training Artificial Neural Network. It is a nonlinear complex model that can be used to configure and train an artificial neuron system. Both Artificial Neural Network and Particle Swarm Optimization model have been managed to solve and optimize several nonlinear models. Heuristic Optimization Weight of Artificial Neural Network (HNN) is a proactive metaheuristic model proposed to optimize the performance of Artificial Neural Network. The proposed system applies Particle Swarm Optimization to find the optimum weights of the Artificial Neural Network instead of using the Backpropagation algorithm. Experimentally, the proposed system has required less processing time (average of 76.91 Sec.) than Backpropagation (average of 93.32 Sec). Furthermore, It has provided better classification accuracy (start from 80% to 97.20%) comparing with Backpropagation (start from 75.32% to 94.32%).

Corresponding Author:

Ali Hakem Alsaeedi,
College of Computer Science and Information technology,
University of Al-Qadisiyah,
Al Diwaniyah, Qadisiyyah Province, Iraq.
Email: ali.alsaeedi@qu.edu.iq

1. INTRODUCTION

Artificial Neural Network (ANN) technically simulates the brain behaviour to develop an artificial system that can be used to manage complex prediction problems [1]. It is originally trained based on Backpropagation algorithm (BP) concepts [2]. The ANN consists of several neurons (nodes) distributed on layers (input, hidden, and output) and uses synaptic weights for labelling links that connect neurons. These weights help artificial neural system to train on predicting the class of corresponding input. BP updates ANN weights based on calculating errors that result from the feedback of previous layers [3].

Particle Swarm Optimization (PSO) is a stochastic optimization algorithm inspired by neutral's behaviour of social animals [4]. It technically the property of systems composed of unintelligent agents with limited individual capabilities but with an intelligent collective behaviour [5]. It is an efficient technique for stochastic search in a multidimensional space, has so far been applied successfully to solve the different complex problems [6-9]. It is a guide that use randomness to find the best possible solutions swiftly. ANNs are powerful models that have a significant application in a range of different fields in medicine, multimedia processing, security, Economic, and so on [10-14].

Furthermore, deep learning models mainly based on principles of ANN. Generally, ANN when training by PB, significantly suffering from a delay in learning rate because complex and long steps that applied [1]. PB trains ANN by tuning modification of weights after each train-cycle, therefore learning rate will be improved slowly. As a result, ANN needs a markable tremendous time to train on a particular problem. Evidently, significant remarkably time required to train ANN by BP algorithm and necessity of improving the prediction accuracy of ANN motives as to proposed new train model to optimize the performance of ANN in terms of accuracy and time. The PSO has a good acceptance by the Artificial Intelligence community because it is a tremendous optimization algorithm and can solve variant complex optimization problems. For a given problem, PSO can explore big multimodal and noncontinuous search spaces and can find the best possible solution, near the optimum value [15]. Therefore, the proposed system (HNN) applies the PSO to find the optimal weights of ANN.

In this paper, a new model of the artificial neural system is proposed based on the stochastic search optimization by PSO and principle of ANN. The contributions of this research mainly face on improving the performance of the ANN system in three aspects: Training time required, and the complexity of training procedure, and accuracy. The proposed HNN model consumes remarkably less time to find optimal weights of an ANN. Technically, this fact is explained by the PSO search strategy used by the proposed model rather tuning modification used in BP. The simplicity of PSO [8] and using the few numbers of parameters in the proposed system reduce the complexity of the proposed system. Therefore, the proposed system does not require an additional process to achieve the established system on the final set of weights. Furthermore, The proposed technique aims to create an efficient supervise model with a very high degree of prediction accuracy. Experimental results show significant predictions for input data by the proposed method in comparison with PB in ANN. The accurate predictions of the proposed technique are capable of achieving better prediction ratios of ANN than ANN when training by BP. Herein, we proposed PSO as a training model of ANN to search on optimal weights that corresponding the particular problem. Figure 1 illustrates the scenario used to update the weights of the proposed system.

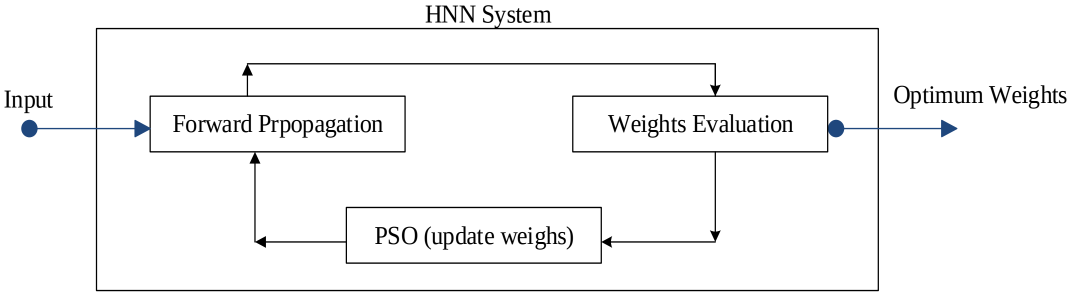


Figure 1. Scenario of weight updating

To develop the ANN algorithm, several studies related to our research, have been proposed for supervised machine learning to classify several objects of data. Most of the studies in the approach of the weights optimization for the neural network have been based on standard algorithms (Backpropagation). Fangcheng *et al* [16], the authors proposed the new modelling of artificial net combines both PSO and backpropagation. The PSO used to optimize the initial weights of ANN and the rest of the learning processing are achieved by the backpropagation algorithm. The proposed system got on MSE ratio reaches 0.5. Gaxiola *et al* [17]. authors introduce both algorithm and particle swarm optimization to type-2 fuzzy weight adjustment. The classification in a proposed system technically depends on the type-2 fuzzy and backpropagation to enable the neural network to learn. Varun *et al* [18]. this study aims to optimize weight and parameters of the objective function. Various types of transfer functions were chosen for the purpose of the Rigorous analysis of the influence of the transfer functions optimization together with the NN weights. Apart from the metaheuristic algorithms, the backpropagation algorithm was used for the comprehensive comparison and validation of the significance of the NN-TFs model. Yu *et al* [7]. have applied the optimization technique to set optimal neurons in the hidden layer of Artificial Neural Network. Particle Swarm Optimization used with proposed system to determine optimal numbers of neurons in the hidden layer. The proposed system achieve the best result when the five neurons in the hidden layer.

2. RESEARCH METHOD

In this section explain the main mechanism of BP to train ANN, PSO, and the last part illustrates the proposed HNN system.

2.1. Artificial neural network (ANN)

ANN system basically consists of neurons and weights. The neurons represented as learning unit is distributed on ANN' layers(input, hidden, and output) [1]. Weights define the value of the interconnection line that connect neurons together. Generally, the architecture of the ANN set by depth, width, and activation functions used on each layer [19]. Depth determines the deep learning level in how many layers inside hidden layers. The width defines the number of neurons in each hidden layer (s). The activation function is applied to define the non-linear relationship between the inputs and the input information into a more useful output [20]. Figure 2 shows the architecture of ANN.

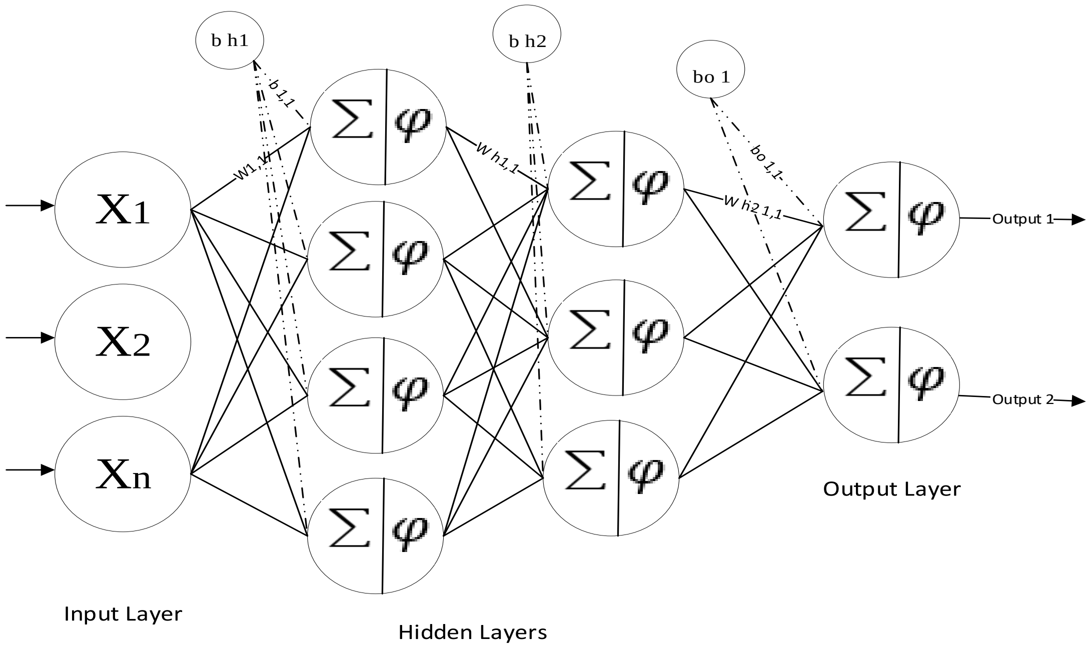


Figure 2. The architecture of ANN

Technically, the processing problem of ANN is based on two operations prediction and training [21, 22]. First, a prediction is achieved by Forward Propagation algorithm, it forward summation of multiplication inputs of neurons with connecting weights to activation function for predicting the class of input. In (1) and (2) calculates the output of Forward Propagation. The net summation Net_i and its output for the activation function Y_i' [23].

$$Net_i = \sum_{j=1}^{n} w_{ij}\, x_i + \mu_i \text{ , j=1, 2, 3,............n} \tag{1}$$

$$Y_i' = \varphi\ (Net_i) \tag{2}$$

where Net_i is the result of net inputs xi multiplied by interconnection weight w_{ij}, φ is the activation function, and Y_i' the actual output of the net.

Second, The ANN trains its weight originally by a Back Propagation (BP) algorithm [1]. The Back Propagation (BP) updates weights of ANN after each train-cycle (epoch) according to product error. Generally, the error is calculated in (3):

$$E = \frac{1}{n} \sum_{i=1}^{n} (y_{Actual\, i} - y_{desired\, i})^2 \tag{3}$$

Where: n number of outputs
Algorithm I illustrates training ANN by concepts of BP:

Algorithm (I): Backpropagation algorithm (PB) .
Input: Dataset

```
Output: optimum Weight
1. Set network layers and neuron ( $n_{input}$ nodes, $h$ hidden nodes, and $onet$ output nodes)
2. Initialize all network weights with random values.
3. Repeat until the termination condition is met:
4. For each training example (x,t) ϵ D do
5. Input x to the network and compute the output ak of units in the output layer.
6. For each network output unit k, calculate its error term δk:
   δk ← ak (1- ak) (tk- ak)                                        (4)
7. For each hidden unit h, calculate its error term δh
   δh ← ah(1- ah)$\sum_k \delta_k$ $w_{kh}$                        (5)
8. Update each network weight wji
   $w_{ji} \leftarrow w_{ji} + a\delta jxji$
9. Return optimum Weight
```

2.2. Particle swarm optimization (PSO)

PSO is a public metaheuristic optimization algorithm inspired by the social behaviour of animals, which live in groups. It's introduced by (R. Eberhart and J. Kennedy) [4]. The main characteristics of PSO are the simplicity of implementation, few numbers of parameters, and successes in optimizing different fields such as data mining, medical application, speech recognition, and image processing, therefore the PSO is an interesting topic for researchers[8]. The standard search mechanism of the PSO starts with selecting the first local optimum (P_{bset}) by each particle then it selects the best P_{bset} as global optimum (g_{best}). The velocities of particles are updated dynamically during the search process and added to the current particle's value for updating. The initial velocity of particles set by zeros. The value of g_{best}, and P_{bset} are updated iterativlly (6) calculates the particle velocity[5].

$$V_i^d(t+1) = \omega(t)V_i^d(t) + c_1 r_1 \left(pbest_i^d - x_i^d(t)\right) + c_2 r_2 \left(gbest^d - x_i^d(t)\right) \tag{6}$$

Where: r1 and r2 are random variables in the range [0, 1], c1, and c2 are positive constants (acceleration coefficients), and ω is the inertia weight. The$V_i^d(t)$, $x_i^d(t)$ is indicating the velocity and position of i^{th} the particle at iteration t in $d^{\ th}$dimension, respectively. The PSO uses the (7) to update the value of the particles (candidate solution) [5].

$$x_i^{t+1} = x_i^t + V_i^{t+1} \tag{7}$$

where: x_i^t is old particle value, x_i^{t+1} is a new particle value.

2.3. Proposed heuristic optimization weight of artificial neural network (HNN)

The limitation of ANN needs a lot of time for training network on objective function [24]. The weights that connect the neurons contribute primarily to the object for which the ANN is used. The proposed method focuses on finding optimum weights during the minimum possible time. The stochastic optimization technique supersedes the backpropagation in original ANN. According to [24, 25], the PSO succeeds in optimizing several problems, therefore, it is used as an optimizer to enhance the performance of ANN. It is clearly shown from the Figure 3 that the proposed system receives the data after initializing the parameters then it optimizes the ANN's weights. Testing data evaluate the optimum weight obtained from the training progress.

2.3.1. Initialization

Firstly, the algorithm has to set the population size which corresponding the candidate weights to adapt PSO with ANN. To increase the exploration of PSO, the weight inertia (ω) is calculated according to [8]. It is calculated iteratively by using (8).

$$\omega = \frac{2+\varepsilon}{2} \tag{8}$$

where $\varepsilon\ is$ a $random\ value\ btween$ [0,1].

2.3.2. Fitness or cost function

The value of the Mean Square Error (MSE) is used as an indicator to evaluate the performance of the proposed system. As shown in 9 calculates the MSE.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(E_i)^2 \qquad (9)$$

where: E_i is different between i actual output and i desired output of.

2.3.3. Weights update

After calculating the fitness of candidate weight, proposed algorithm updates the velocity to be added to weights and produce a new candidate weight. The weights matrix (w) is updated as shown in Algorithm II frequently until the HNN achieves the termination criteria. The population size of PSO is adaptive to correspond to the weight's matrix of ANN. The proposed system combines weights of whole ANN in one matrix as shown in Formula 7:

$$P_i = \bigcup_i^n \bigcup_j^m w_{i,j} \qquad (10)$$

For example, assume the n, m is dimensional of weight matrix (w) of ANN

$$w = \begin{bmatrix} w_{1,1} & \cdots & w_{1,n} \\ \vdots & \cdots & \vdots \\ w_{n,1} & \cdots & w_{n,m} \end{bmatrix}_{n\times m}$$

The domination of population (P) and velocity (v) that generated is s the value of s is calculates by square n of the weight matrix.

$$v = \begin{bmatrix} v_{1,1} & \cdots & v_{1,s} \\ \vdots & \cdots & \vdots \\ v_{z,1} & \cdots & v_{z,s} \end{bmatrix}_{z\times s} \;, \; P = \begin{bmatrix} P_{1,1} & \cdots & P_{1,s} \\ \vdots & \cdots & \vdots \\ P_{z,1} & \cdots & P_{z,s} \end{bmatrix}_{z\times s}$$

To control on learning orientation, the proposed system modifies biases (b) value based on the error of the system. In (11) modifies biases.

$$b = \frac{rand}{2\times n} E \qquad (11)$$

After HNN modifies weights, the proposed system reconstructs the weights matrix to be suitable for forwarding propagation. As shown in (12) redistributed weight generated by POS.

$$w_i = \bigcup_{i=1}^{n} \bigcup_{j=m(i+1)-1}^{i\times m} p_j \qquad (12)$$

2.3.4. Termination criteria

The proposed model stops when one of the stopping criteria used in the stochastic algorithm is satisfied. These criteria are: exceed of optimization time, no change of optimization in n iteration, or user's pre-defined condition [26, 27]. Figure 3 shows the main steps of the proposed system.

The Algorithm (II) illustrates the pseudocode of the HNN.

Algorithm (II): Heuristic optimization weight of artificial neural network (HNN)
Input: Dataset
Output: Optimum Weight

```
1. Set network layers and neuron ( $n_{input}$ nodes, $h$ hidden nodes, and $onet$ output nodes)
2. Initialize PSO
3. Repeat until the termination condition is met:
4. For each $w, t$ (candidate weights) ϵ population do
5.     Convert $w$ to correspond ANN weights in each layer (12)
6.     Evaluate $w$
7.     Update velocity of $w_{t+1}$ , $g_{best}$ , $p_{best(t+1)}$ of $w$ (6), (7)
8. Return optimum Weight
```

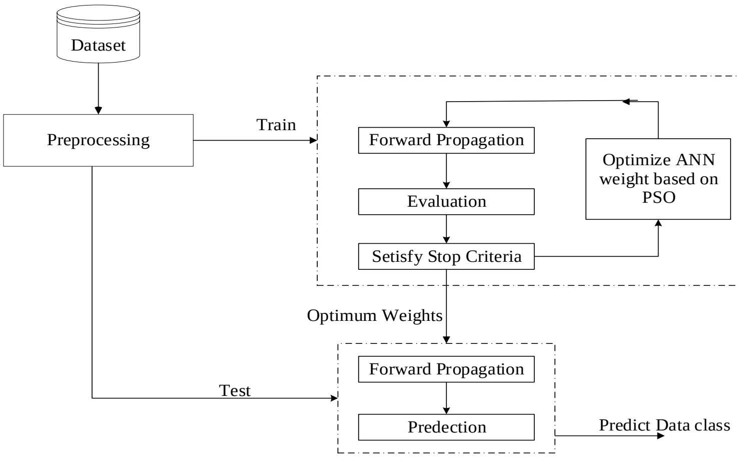


Figure 3. The main steps of the proposed HNN model

3. RESULTS AND DISCUSSION

The evaluation strategy is based on testing the proposed model using the ten datasets of UCI with different attributes [28]. The proposed HNN is compared with BP in training ANN. Table 1 illustrates the dataset description:

Table 1. UCI dataset descriptions [19]

No.	Dataset Name	Size (Attribute x Feature x Class)
1	Breast cancer1	699 x 8 x 2
2	Wine	178 x 13 x 3
3	Spectf	267 x 44 x 2
4	Ozone level detection	2536 x 72 x 2
5	Breast Tissue	106 x 9 x 4
6	Car evaluation	1728 x 6 x 4
7	Drug consumption	1885 x 32 x 7
8	HTRu-2	16384 x 8 x 2
9	Burst Header Packet	1075 x 22 x 4
10	Avila	20867 x 10 x 12

The proposed HNN model is tested over ten datasets to classify the problem in the UCI machine learning. The datasets have been explained in the Table 2 which is used as a benchmark in order to evaluate the proposed algorithm and compare the results with the standard ANN. The parameters of algorithms we are compared set as shown in Table 2:

Table 2. ANN and PSO algorithms parameters

Algorithm	Parameters
ANN with BP	$Epochs\ = 25\ , validition\ check\ = 6, Mu\ =\ 0.001, \quad gradient\ =\ 1 \times e^{-7}$
HNN	$c_1, c_2 = 1.25, Population\ size\ = 50, Max\ iteration\ =\ 25$

The experiments use the logistic and the tangent hyperbolic functions as an activation function for both the HNN and BP algorithms. Table 3 shows the results of the experiments of both the mentioned algorithms. The results of HNN show a better performance in the sample datasets Breast cancer1, Wine, Ozone level detection, Car evaluation, HTRu-2, Burst Header Packet, and Avila. While the ANN with BP recorded the best results in dataset Spectf and Breast Tissue. Both algorithms obtained equal accuracy in dataset DC. The training time of HNN required less time than ANN. It is clearly shown in the Table 3 that the metaheuristic function helps in updating the weight as fast as possible compared with the original ANN -with BP-. Figure 3 shown compare performance of ANN by using BP and HNN (dataset breast cancer). Figure 4 shown compare performance of ANN by using BP and HNN (dataset wine). Figure 5 shown the performance of algorithms by using the MSE as compared criteria. The results indicate that BP has better accuracy:

Table 3. The evaluation result of the HNN and ANN

No.	Dataset Name	Performance of the ANN		Performance of the HNN	
		Accuracy	Time in sec	Accuracy	Time in sec
1	Breast cancer1	94.32	15.25	96.72	10.28
2	Wine	75.32	5.98	92	3.22
3	Spectf	79.4	18.95	76.6	14.37
4	Ozone level detection	90.37	128.5	97.12	98.22
5	Breast Tissue	85.9	18.99	80	13
6	Car evaluation	82.31	68.32	84.22	48.47
7	Drug consumption	84.2	95.45	84.2	80.36
8	HTRu-2	88.6	137.9	97.20	122.95
9	Burst Header Packet	89.44	185.73	93.51	168.02
10	Avila	79.32	258.15	86.31	210.17

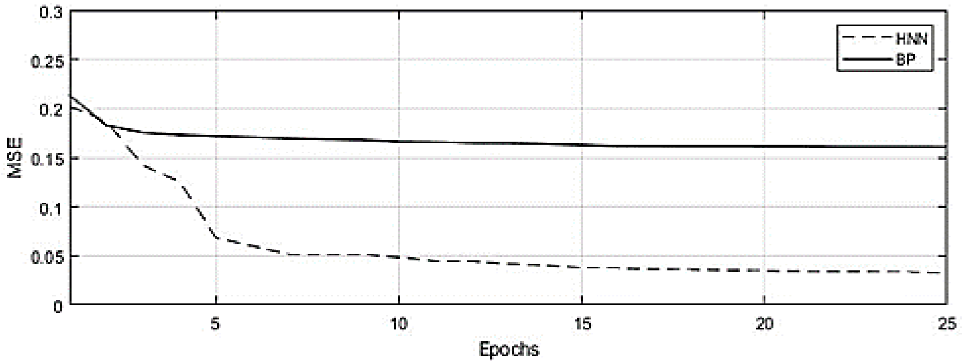


Figure 3. Compare performance of ANN by using BP and HNN (dataset breast cancer)

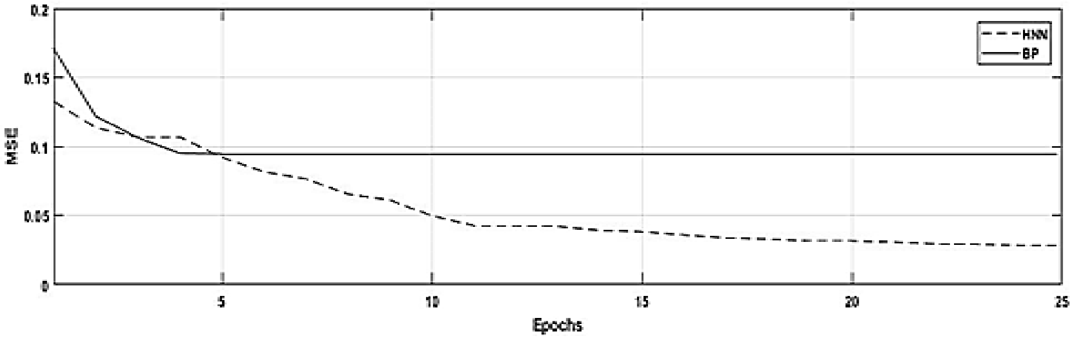


Figure 4. Compare performance of ANN by using BP and HNN (dataset wine)

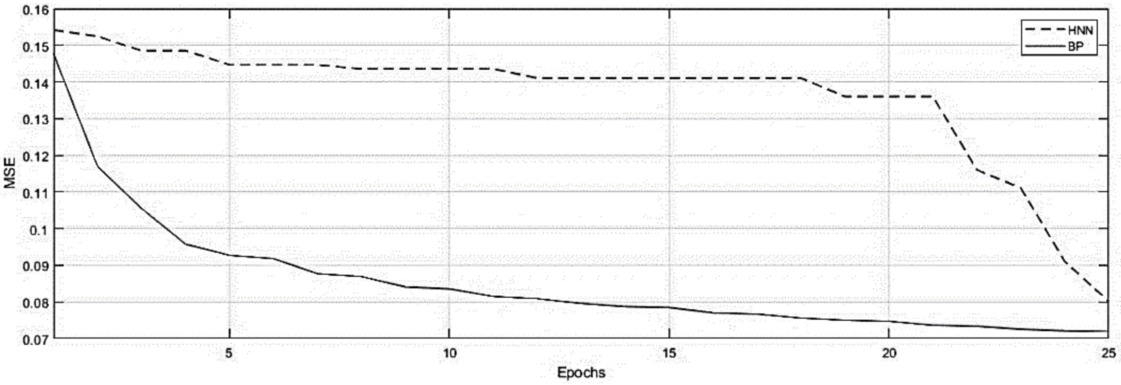


Figure 5. Compare performance of ANN by using BP and HNN (dataset spectf)

The proposed system shows that the stochastic techniques (PSO) overcome on the stagnation in ANN when using the BP algorithm; also, it optimizes the accuracy of ANN prediction. The accuracy of HNN is less than BP as shown in Figure 5, although the performance of HNN is better because it could be jumping from the stagnation stage at last quarter from search progress.

4. CONCLUSION

Artificial Neural Network is a nonlinear computing system inspired by the biological neural networks of the brain. It consists of several layers and each layer has several neurons (nodes). The learning behaviour of ANN is based on errors that result from the feedback of the previous layer to update the weights of the neuron layers. The weights are the outcome of the trained network on a specific problem that is used to predict the class of inputs. They start randomly and update dynamically by the backpropagation algorithm during ANN training. The training process of ANN requires tremendous time, in addition to the complex procedure that has been followed when using backpropagation to train ANN. PSO is one of the popular stochastic methods for features selection. It has been recognized as fast, flexible, easy to implement, and succeed in optimizing different fields. Therefore, the PSO is used to find the optimal weight of ANN. By using the metaheuristic (PSO), the proposed system enhances the performance of ANN in both accuracy and processing time. The training time is reduced because of the simplicity of processes in the training stage of metaheuristic. In future work, we suggest enhancing the exploration of metaheuristic model before using to optimize ANN weight to reduce stagnation in the advance search process.

REFERENCES

[1] J. Too, A. R. Abdullah, N. M. Saad, N. M. Ali, and T. N. S. Tengku Zawawi, "Featureless EMG pattern recognition based on convolutional neural network," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 14, no. 3, pp. 1291–1297, 2019.

[2] S. Ding, H. Li, C. Su, J. Yu, and F. Jin, "Evolutionary artificial neural networks: A review," Artif. Intell. Rev., vol. 39, no. 3, pp. 251–260, 2013.

[3] S. A. Mohamed, M. Othman, and M. H. A. Abdullah, "A review on data clustering using spiking neural network (SNN) models," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 15, no. 3, pp. 1392–1400, 2019.

[4] R. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," *Proc. Int. Symp. Micro Mach. Hum. Sci.*, pp. 39–43, 1995.

[5] S. Mirjalili and A. Lewis, "S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization," *Swarm Evol. Comput.*, vol. 9, pp. 1–14, 2013.

[6] R. Islam, M. N. Akhtar, B. R. Ahmad, U. K. Das, M. Rahman, and Z. I. A. Khalib, "An approach to building energy clusters using particle swarm optimization algorithm for allocating the tasks in computational grid," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 14, no. 2, pp. 826–833, 2019.

[7] L. B. Khuzyatova and L. A. Galiullin, "Optimization of parameters of neuro-fuzzy model," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 17, no. 3, pp. 1206–1209, 2019.

[8] A. H. Jabor and A. H. Ali, "Dual Heuristic Feature Selection Based on Genetic Algorithm and Binary Particle Swarm Optimization," *J. Univ. BABYLON pure Appl. Sci.*, vol. 27, no. 1, pp. 171–183, 2019.

[9] M. B and G. Holi, "An optimal and secure watermarking system using SWT-SVD and PSO," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 18, no. 2, p. 917, 2020.

[10] S. Shindo, T. Goto, T. Kirishima, and K. Tsuchida, "An optimization of facial feature point detection program by using several types of convolutional neural network," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 2, pp. 827–834, 2019.

[11] M. Yusoff, F. M. Dnajib, and R. Ismail, "Hybrid backpropagation neural network-particle swarm optimization for seismic damage building prediction," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 14, no. 1, pp. 360–367, 2019.

[12] R. T. C and R. Sivramakrishnan, "Fuzzy Neuro-Genetic Approach for Feature Selection and Image Classification in Augmented Reality Systems," vol. 8, no. x, pp. 194–204, 2019.

[13] A. Sarkar, J. Dey, and A. Bhowmik, "Multilayer neural network synchronized secured session key based encryption in wireless communication," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 14, no. 1, p. 169, 2019.

[14] S. Verma, G. T. Thampi, and M. Rao, "ANN based method for improving gold price forecasting accuracy through modified gradient descent methods," *IAES Int. J. Artif. Intell.*, vol. 9, no. 1, pp. 46–57, 2020.

[15] M. Lones, *Sean Luke: essentials of metaheuristics*, vol. 12, no. 3. 2011.

[16] R. Yu, X. An, B. Jin, J. Shi, O. A. Move, and Y. Liu, "Particle classification optimization-based BP network for telecommunication customer churn prediction," *Neural Comput. Appl.*, vol. 29, no. 3, pp. 707–720, 2018.

[17] F. Gaxiola, P. Melin, F. Valdez, and O. Castillo, "Optimization of type-2 fuzzy weight for neural network using genetic algorithm and particle swarm optimization," *2013 World Congr. Nat. Biol. Inspired Comput. NaBIC 2013*, vol. 1, no. 1, pp. 22–28, 2013.

[18] V. K. Ojah, A. Abraham, and V. Snase, "Simultaneous Optimization of Neural Network Weights and Active Nodes using Metaheuristics," in *14th International Conference on Hybrid Intelligent Systems*, 2014, pp. 248–253.

[19] N. Z. Mohd Ali, I. Musirin, and H. Mohamad, "Clonal evolutionary particle swarm optimization for congestion management and compensation scheme in power system," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 16, no. 2, pp. 591–598, 2019.

[20] C. Gulcehre, M. Moczulski, M. Denil, and Y. Bengio, "Noisy activation functions," *33rd Int. Conf. Mach. Learn. ICML 2016*, vol. 6, pp. 4457–4466, 2016.

[21] R. Kaubruegger, L. Pastori, and J. C. Budich, “Chiral topological phases from artificial neural networks,” *Phys. Rev. B*, vol. 97, no. 19, pp. 1–8, 2018..

[22] N. F. Fadzail, S. M. Zali, M. A. Khairudin, and N. H. Hanafi, “Stator winding fault detection of induction generator based wind turbine using ANN,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 19, no. 1, pp. 126–133, 2020.

[23] S. Lawrence, A. C. Tsoi, and A. D. Back, “Function Approximation with Neural Networks and Local Methods : Bias, Variance and Smoothness,” *Aust. Conf. Neural Networks*, pp. 16–21, 1996.

[24] A. Rakitianskaia and A. Engelbrecht, “Weight regularisation in particle swarm optimisation neural network training,” *IEEE SSCI 2014 - 2014 IEEE Symp. Ser. Comput. Intell. - SIS 2014 2014 IEEE Symp. Swarm Intell. Proc.*, pp. 34–41, 2015.

[25] M. D. Sulistiyo, R. N. Dayawati, and Nurlasmaya, “Evolution strategies for weight optimization of Artificial Neural Network in time series prediction,” *Proc. 2013 Int. Conf. Robot. Biomimetics, Intell. Comput. Syst. ROBIONETICS 2013*, no. November, pp. 143–147, 2013.

[26] A. H. Al-saeedi, “Binary Mean-Variance Mapping Optimization Algorithm (BMVMO),” *J. Appl. Phys. Sci.*, vol. 2, no. 2, pp. 42–47, 2016.

[27] A. O. Topal and O. Altun, “A novel meta-heuristic algorithm: Dynamic Virtual Bats Algorithm,” *Inf. Sci. (Ny).*, vol. 354, pp. 222–235, 2016.

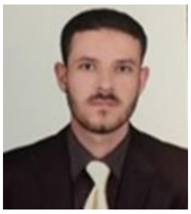
[28] D. Aha, E. Keogh, Patrick Murphy Christopher Merz, C. Blake, S. Hettich, and D. Newman, “The UCI Machine Learning Repository,” *National Scientific Foundation*, 2007.

BIOGRAPHIES OF AUTHORS

Ali Hakem Alsaeedi is completed B.Sc. in Computer Sciences in 2006 from the college of sciences at University of Al-Qadisiyah, Diwaniya, Iraq. Received his M.Sc. (master) in computer sciences in the year 2016 from the college of computer sciences at the Yildiz Technical University (YTU), Istanbul, Turkey. He has worked as a lecturer at a number of the Iraqi Universities in the areas of Artificial Intelligent, Data mining, and signal processing. He currently works as a lecturer in the University of Al-Qadisiyah. His research interests machine learning, smart optimization algorithms, and optimization of Big Data. Ali has several publications in the areas of the binary of metaheuristic optimization and data mining

Ali Hussein Aljanabi is completed B.Sc. in Computer Sciences in 2006 from the college of sciences at University of Al-Qadisiyah, Diwaniya, Iraq. Received his M.Sc. (master) in computer sciences in the year 2014 from the college of computer sciences at the Babylon University, Babil, Iraq. He is currently working as a lecturer in the Al-Qadisiyah Education Directorate. His research interests machine learning, Security information, and soft computing. Ali has several publications in the areas of data mining and network identification.

Mehdi Ebady Manaa is currently an assistant professor in the department of information network, College of Information Technology, University of Babylon. He received his bachelor's degree from the University of Babylon, college of science in 1999-2000. His master of science from University Utara Malaysia (UUM), Malaysia in 2012. He received his PhD in Computer Science and in the field Network Security and Data Mining using Cloud Computing from the University of Babylon - College of Information Technology in 2016. He is currently focusing on the detection of the attacks using data mining techniques. The main interesting fields are network administration and distributed object-oriented topics, network protocols TCP/IP suite, Mobile Ad-hoc network (MANet), Data Mining Techniques (clustering and classification), communication software, network security, cloud computing, Internet of Things, and unstructured data



Adil received his B.Sc. in Computer Science in 2009 from the College of Science at University of Al-Qadisiyah, Diwaniya, Iraq. He received his M.Sc. in Computer Science in 2016 from Kent State University, Ohio, United States. He is working as a lecturer at the University of Al-Qadisiyah. His teaching and research interests include Computer Vision, Biomedical Imaging, Image Processing, machine and deep learning.