TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 14, No. 1, April 2015, pp. 163 ~ 172
DOI: 10.11591/telkomnika.v14i1.7233

Near Optimal Convergence of Back-Propagation Method using Harmony Search Algorithm

Abdirashid Salad Nur*[1], Nor Haizan Mohd Radzi[2], Siti Mariyam Shamsuddin[3]
Soft Computing Research Group, Faculty of Computing, Universiti Teknologi Malaysia (UTM), 81310 Skudai, Johor, Malaysia
*Corresponding author, e-mail: salaadnuur@yahoo.com[1], haizanradzi@utm.my[2], sitimariyams@gmail.com[3]

Abstract

Training Artificial Neural Networks (ANNs) is of great significanceand a difficult task in the field of supervised learning as its performance depends on underlying training algorithm as well as the achievement of the training process. In this paper, three training algorithms namely Back-Propagation Algorithm, Harmony Search Algorithm (HSA) and hybrid BP and HSA called BPHSA are employed for the supervised training of Multi-Layer Perceptron feed forward type of Neural Networks (NNs) by giving special attention to hybrid BPHSA. A suitable structure for data representation of NNs is implemented to BPHSA-MLP, HSA-MLP and BP-MLP. The proposed method is empirically tested and verified using five benchmark classification problemswhich are Iris, Glass, Cancer, Wine and Thyroid dataset on training NNs. The MSE, training time, and classification accuracy of hybrid BPHSA are compared with the standard BP and meta-heuristic HSA. The experiments showed that proposed method has better results in terms of convergence error and classification accuracy compared to BP-MLP and HSA-MLPmaking the BPHSA-MLPa promising algorithm for neural network training.

Keywords: *artificial neural networks, harmony search, backpropagation, classification problem*

1. Introduction

Artificial Neural Networks (ANNs) are considered to be powerful tools in pattern classification, prediction of future events, adaptability, noise filtering and ability to learn from its surroundings [2], [5-6], [11, 24].

The process of training neural network is an optimization task in which a set of connection weights of ANN is determined in order to minimize the error. The connection weights are initially given indiscriminately to every neuron and later these weightsare modifiediteratively until the desired or near targeted output value is obtained through altering the network weights accordingly [14]. When the training process is ended, unseen data known as test dataset is used to test the generalization ability of the classifier. The training problem of ANN requires more powerful optimization methods since the determination of connection weight is considered crucial task and significantly contributed to the output value [24].

Back-Propagation (BP) is a gradient based algorithmand has been widely used for training artificial neural networks. The BP algorithm computes the network's output and reduces the mean square error (MSE) between the actual output and the targeted output by adjusting weights [3, 6, 8, 14, 24]. The gradient-based algorithms are designed for local search where the search area is largely attributed by the search starting point. Although there are no doubts concerning BP performance in some non-linear separable problems, It has lead to slow convergence opening the possibility of getting stuck in local minima.

In order to overcome the local minimum problem, researchers have applied meta-heuristic global optimization algorithms such as Genetic Algorithms (GA) [7-9], [12], Particle Swarm Optimization (PSO) [9, 11, 13, 22], Ant Colony Optimization (ACO) [1, 23] and recently introduced Harmony Search algorithm (HSA) [6, 14, 20] to seek the optimal network weights. These global search algorithms are claimed have produced better results since they are able to expand the search space in order to avoid the local minima problem. Apart from training ANN with global optimization algorithm alone BP has also been combined with global searchtechniques such as GA and PSO as can be seen in the work conducted by [1, 4, 10], [15-

Received December 24, 2014; Revised March 20, 2015; Accepted March 28, 2015

16], [18]. In these hybrid forms of BP-GA and BP-PSO, the GA and PSO are used to initialize and modify the weights of BP network.

In this paper, a hybrid BP and HSA or known as BPHSA is employed to train the feed forward neural network (FFNN). This algorithm will exploit the capability of HSA in global search to avoid local minima problem faced by BP. HSA is employed to adjust the weights and biases of the network whenever BP fails to generate near optimal weights. The scheme is empirically tested and verified with five benchmark datasets from the University of California at Irvine (UCI) Machine learning Repository. These datasets include: Iris, Cancer, Glass, Wine and Thyroid datasets. The experimental results of the proposed scheme showed that BPHSA is an efficient and good candidate for training feed forward neural network. The paper is organized as follows: Section 2 describes the training algorithms such as BP, HSA and the proposed technique of hybrid BPHSA. The experimental setups, datasets, structure of the network and its implementation as well as parameter tuning are presented in Section 3.Section 4 explains the experimental analysis and the performance of the proposed model. Finally, the conclusion of this work is summarized in Section 5.

2. Training Algorithms

In this study several algorithms are used to train ANNs: the Back propagation (BP) algorithm, Harmony Search Algorithm (HSA) and a hybrid of BP and HSA known as BPHSA. A brief description on each algorithm is given in the following paragraphs.

2.1. Back Propagation Algorithm

Back-propagation (BP) is one of most famous supervised training algorithms for FFNN [8, 15]. BP aims to minimize the total MSE (Mean Square Error) between the expected and actual output. This MSE is applied to monitor the exploration of the BP algorithm in the weight space. In addition to the usage of BP for training ANN, it is unnecessary to predetermine the exact design of ANN network architecture and parameters (11). Equation (1) presents the formulain which is used to adjust the weights using the BP algorithm:

$$\mathrm{w_{lji}(k+1) = wlji(k) - \mu\ (\partial(ji(w))/\ \partial)_{lji}} \quad (1)$$

Where $\mathrm{w_{lji}}$ is the connection weight to the neuron l in l-1 layer and neuron j in layer l and μ is the learning rate that is a positive number which is utilized to supervise the learning steps and it is usually a minor positive number [19].

The training process of BP consists of two mechanisms which are: (1) forward and (2) back propagation. In the forward pass, the input information units are transmitted from the input layer via the hidden layer and then transmitted to the output layer. In the backward pass, the errors are back-propagated along the original connection path. Modifying the weights of neurons in each layer can reduce the error. BP has the capability of local search that generates near local optimal weights; however, its ability to global search is too weak. Therefore, to acquire the benefits of both techniques and to avoid their weakness to enhance the learning ability of the Multi-Layer type of feed forward neural network, we are focusing on changes in the weights.

2.2. Harmony Search Algorithm (HSA)

HSA is a new meta-heuristic optimization algorithm derived from the improvisation process of the musician in a musical collection. It has been proposed by Geem et al in 2001. The solution vector is analogous to harmony in music. Both schemes of local and global search are analogous to musical requirements. Thus HSA can be easily applied for solving optimization problems [25]. For instance, the musicians test and play a tone on their instrument to select the perfect tone (or outcome) in harmony with the rest of the band. For creating a new harmony, either a tone from harmony memory is played with modifications, an existing tone from the memory is played, or a totally new tone from a range of acceptable tones is used (played). Only the best harmonies in the memory are saved and remembered until better ones are discovered and exchangedwith the worst harmonies in the memory. The following steps of HSA demonstrate the perfect solution vector that has been obtained during search component value which was optimal using some optimal objective functions assessed for this solution vector [6, 20].

***Step 1**: Define and Initialize all HS parameters (HMS, HMCR, PAR and NI) and the problem.*

***Step 2**: Initialize Harmony Memory (HM) with random values of vectors.*

$W_i(j)= LB_j+r(UB_j-LB_j)$ *for* $j=1,2,\ldots,n$; $i=1,2,..HMS$ *where* $r\in (0, 1)$

***Step 3**: Use objective function to improvise New Harmony vector.*

While $(j \leq n)$ *do*

If $(r_1< HMCR)$ *then* $H_{new}(j)=H_a(j)$ *where* $a \in (1,2\ldots,HMS)$ *and* $r_1 \in (0, 1)$

If $(r_2<PAR)$ *then* $H_{new}(j)= H_{new}(j)\pm r3*BW$ *where* r_2 *and* $r_3 \in (0, 1)$

Else $H_{new}(j)=LB_j(j)+r(UB_j(j)-LB_j(j))$ *where* $r \in (0, 1)$

End loop

***Step 4**: If the created harmony is better than the one in the memory, then replace it.*

***Step 5**: If the termination criteria were met then stop; otherwise repeat steps 3 & 4.*

***Step 6**: The best solution of the obtained harmony is stored in the harmony memory (HM).*

2.3. The proposed Algorithm (BPHSA)

Hybridization refers to the presence of problem dependent information in an overall search sample. Hybridization can be distinguished into strong and weak hybridization [21, 26]. The first one refers knowledge representation using a specific operator whereas weak hybridization is used to combine several algorithms to improve the result of another one separately or it can be used as an operator of the other. The hybridization approach used in this work is the combination of two algorithms (weak hybridization) where one of them acts as an operator on the other. Hence, we combine HSA with BP algorithm known as BPHSA. HSA is used to generate new weights for MLP whenever BP failed. The hybrid BPHSA will exploit benefits obtained from both algorithms and avoid weakness to enhance the learning ability of the Multi-Layer type of feed forward neural network. The following are the steps which must be followed to use of the hybrid form of BPHSA named “BPHSA-MLP” in the training of the supervised NNs:

Step 1: *Preprocess dataset*: The dataset is normalized in order to scale all data in the range of [0, 1]. The normalized data are randomly divided into training and testing samples; therefore the chosen data are removed from initialized data while the remaining data are assigned as training sets to investigate the effectiveness of the model towards the MLP learning.

Step 2: *Determine number of input, hidden and output nodes for MLP architecture*: a three layer network architecture is created corresponding to the dataset requirements. For instance, number of input nodes are determined based on dataset attributes (variables). The output nodes also depend on the dataset’s desirable output while the number of hidden nodes are determined based on equation 2; therefore the number of input layer nodes, hidden layer nodes and output layer nodes in each dataset differ from each other.

Step 3: *Initialize weights & bias randomly*: weights and bias of MLP are randomly initialized. Input patterns are presented to the network to start the training.

Step 4: *Generate net output of MLP and evaluate error*: Here, the network output is calculated and compared to the target value and then errors are evaluated by comparing to the threshold value. If the error is greater then the threshold value and steady stateof BP is not met; hence weights and bias of MLP are adjusted using BP. However if the errors are not reduced gradually to contribute to the network’s generalization ability after six attempts without change, the BP is unable to train the network efficiently due to local minima or over fitting problem. We choose six iterations as the value of steady state parameter because if BP attempts to decrease the error after six iterations and the errors remain unchanged, the BP will surely be stuck in local minima which consumes convergence time before finally reaching the maximum number of learning iterations unsuccessfully; therefore, it does not make sense to wait until BP reaches the maximum iterations without achieving the training goal or reducing the error rate.

Step 5: *Run HSA to generate new values to adjust the weights and bias of MLP*: The Harmony Search Algorithm is called only when BP fails to train the network, has confusion of local minima, or the generalizing ability is poor caused by sub optimal weights or over fitting problem. Therefore, early stopping criteria is required to prevent the network being trapped in local minima or over fitted. The BP steady state mechanism is used to exchange the training scenario from BP to HSA to stop the training earlier if BP does not pass this process control after five iterations which supports the network to decide whether to continue or stop the training. The error rate is rising instead of minimizing or the error rate is unchanged after six iterations whichshows the failure of BP; this is what we call the steady state of BP. If steady stateof BP is met then HSA is called to produce new weight values to continue learning process of MLP and prevent network training disturbance and failure.

Step 6: *Update weights & bias for MLP*: The values in the Harmony Memory generated by HSA are used as weights and bias of MLP neurons. Network weights and bias are then adjusted probably due to BP's failure; the updated weights are used in order to decrease the errors and to finish the learning process successfully.

Step 7: *Stop training and Save results*: The training is stopped if the training goal is met or the maximum iterations is reached. And then, the results obtained from the training process are returned to prepare the network for testing and to compare the algorithms in terms of error convergence and classification accuracy in both training and testing accuracies.

3. Training Neural Networks using BPHSA

The feed forward training process mainly involves deciding the connection weights among the neurons which minimizes the error. Although HSA can be used for both continuous and discrete optimization problems, it can also be used for NN weights

3.1. Data Representation

The NN weights are taken from the harmony vector in the harmony memory (HM) which contains several data strings to representweights of input through hidden layer processing elements, hidden across output processing elements, hidden and output biases [26].

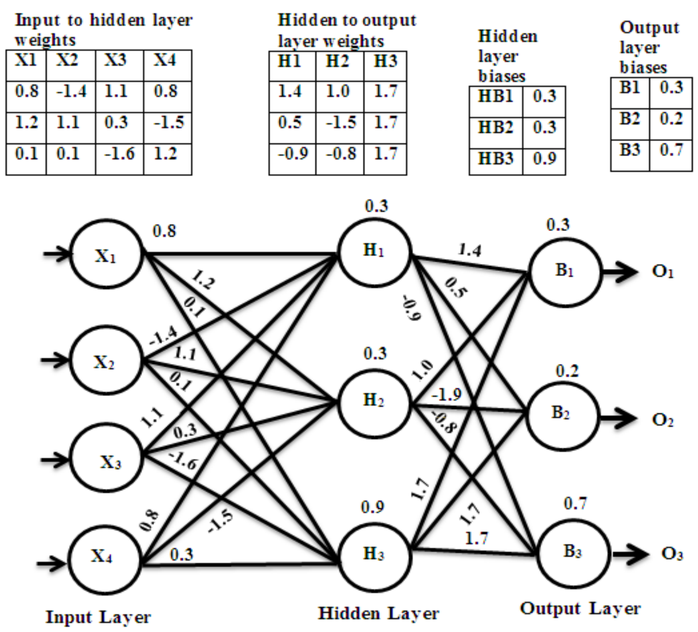

Figure 1. FFANN sample for weight vector representation

3.2. Problem Initialization

The objective (fitness) function of hybrid BPHSA used for training NN is MSE. Lower and upper boundsare taken as [-1, 1]. All solution vectors (harmonies) in the HM are randomly created using equation (1). These generated solutions are used as weights for NN:

$$\mathrm{Wi(j)} = \mathrm{LBj} + \mathrm{r(UBj - LBj)} \text{for } \mathrm{j} = 1,2,\ldots,\mathrm{n};\ \mathrm{i} = 1,2,\ldots\mathrm{HMS} \text{ where } \mathrm{r} \in (0,1) \quad (1)$$

4. Experimental setup
4.2. Datasets
The performance of the proposed model is tested using five datasets obtained from the University of California at Irvine (UCI) Machine learning Repository. These datasets are: Iris, Cancer, Glass, Wine and Thyroid datasets.

Iris dataset: one of the most popular and well known datasets used in classification problem which can be found in the pattern recognition literature. It consists of three classes (1) iris setosa, (2) iris versicolor and, (3) iris virginica. Each class has fifty types of iris plant. Its classification is based on sepal length, sepal width, petal length and petal width.

Cancer dataset: originally, this dataset was invented by Dr. William H. Wolberg as reported in clinical cases at the University of Wisconsin Hospital. It consists of 699 instances in which 485 are benign examples and 241 are malignant examples.

Wine dataset: wine came from the analysis of chemical of wines which were cultivated in the same region then derived from three different cultivars (group of plants chosen for desirable features). The dataset consists of 178 instances and 13 continuous attributes of three classes in which all classes are separable; they have been used by many others for comparing several classifiers.

Glass dataset: This dataset contains 214 instances and 9 continuous attributes. This data was used to classify glass types that were of interest in criminological investigation. The missing glass can be benefitted as evidence at the scene of the crime if it is correctly determined. Many chemical measurements are employed to determine each glass sample.

Thyroid disease dataset: this dataset is the largest dataset among the datasets used in this experiment. It consists of 7200 instances with 21 features categorical and continuous in which 15 are binary data with the remaining 6 continuous. The aim of classification is to decide whether a patient referred to the clinic is hypothyroid. It contains three types of patients; hyper-function, normal function and subnormal function. The data are divided into single training and testing sets with 3772 and 3428 instances respectively.

4.3. Datasets Partition
Each data set is divided according to ten-fold cross partition scheme. Therefore there are two subsets (10% and 90%) for training and testing purposes, respectively. This process is applied to all datasets except thyroid dataset. The training set is responsible for computing the gradient and updating the bias and the weights of the network. During training, the errors are controlled at each iteration; using BP Steady state. The test set is used after training to compare different models to examine the ability of the classifier to classify. The experimental results obtained from all three algorithms forall datasets are depicted under experiments and discussion section. The hybrid BPHSA and other algorithms are executed in five independent runs to train the algorithm for good learning capability and the applied test sets to evaluate the generalization ability of the proposed scheme. But the thyroid dataset is not applied to this procedure because it is originally partitioned into single training and testing set.

4.4. Implementation
Three layered network architecture for feed forward NNs is used for all datasets. The number of nodes in every layer highly depends on the dataset representation. The selection of hidden layer nodes is an open subject and there is no standard formula or procedure that can support the optimal hidden layer nodes. However, in this work, Equation (2) is used to determine the number of hidden layer nodes[17]. As a result, the number of processing elements in the hidden layer for Iris, Glass, Wine, Cancer and Thyroid dataset are 3, 4, 6, 4 and 8 respectively,

$$\text{Numberofhidden neurons} = \sqrt{\text{InNode} * \text{OutNode}} \tag{2}$$

where
InNode= number of input nodes
OutNode= number of output nodes

Moreover, the tanggented sigmoid transfer function which is also known as tansig is used to describe the final output of nodes. The tansig function is used as a transfer function between input layer and hidden layer and between hidden layer and output layer. It is derived from hyperbolic tangent function and has the ability to deal directly with negative numbers; the following is the equation formula of tansig transfer of function:

F(x)= 2/(1+exp (-2*n))-1 (3)

In this work,Mean Square Error (MSE) is employed to measure the error for NN. This function is continuous, monotonous, differentiable as well as it takes a single minimum function. The MSE can be defined as:

$$\mathrm{MSE} = \frac{1}{2}\sum_{K}(\mathrm{DesiredOutput} - \mathrm{NetworkOutput})^2 \quad (4)$$

Where:
Desired Outputis wanted or target value
Network Outputis actual output produced by the network.

For comparison reasons a standard BP and standard HSA is used to train feed forward NNs with same network architecture and a fitness functionto evaluate the proposed technique of BPHSA. Learning rate is set to 0.7, learning iteration is set to 5000 and error threshold is set to 0.005. The steady state parameter -which is used as exchange mechanism between the standard BP and HSA – isset to 6. The meaning of steady state is explained in section 2.3 step 5.The initial values of weights are randomly selected between -1 and 1. All algorithms are coded and executed on the same computer using Matlab. The HSA parameters are shown in Table 1.

Table 1. HAS Parameter setting

No.	Parameter	Value
1	LB	-1
2	UB	1
3	PAR	0.7
4	HMCR	0.95
5	HMS	21
6	NI	100

5. Experiments and Discussion

In this section, we present the experimental results obtained from the execution of five independent runs using three training algorithms which are: standard BP, standard HSA and the proposed method BPHSA for MLP. The average, median and standard deviation of MSE along with classification accuracy, learning iteration and training time were taken into consideration as the performance parameters of the algorithms.

In the new scheme, BPHSA-MLP, the MLP network is initially trained with standard BP. The learning process of MLP is only passed to HSA whenever we believe BP fails to converge. In other words, the BP-MLP halts the training whenever a predefined number of iterations of the steady state parameter is met. To reinitiate the learning process of MLP, the HSA continues the learning by adjusting weights and biases accordingly.

The training process can be stopped when either the minimum error of 0.005 is met or the maximum epoch of 5000 is reached. According to the BP, the learning can be stopped when either the aforementioned criteria are met or the steady state parameter equals 6. The parameter, steady state, is used either to stop the training process of MLP if BP is involved in the network training or to transfer the training process of MLP between BP and HSA algorithms to avoid the local minima.

Table 2. Results of BPHSA-MLP, HSA-MLP and BP-MLPon Iris Dataset

	MSE	Median	Std. Dev.	Accuracy	Time (s)	Epoch
BPHSA-MLP	0.00598	0.0050	0.002126617	98.5	16	1087
HSA-MLP	0.0323	0.0097	0.032619243	93.9	61.2	5000
BP-MLP	0.06662	0.0715	0.01156058	96.82	Stopped at 75.6 Sec	Not converged

As can be seen in Table 2, results show that BPHSA-MLP has converged at epoch 1087 in 16 seconds with an accuracy of 98.5% and MSE of 0.0059. These results indicate that BPHSA-MLP has the fastest convergence speed, the highest accuracy rate, the smallest error and the shortest time compared to BP-MLP and HSA-MLP.

Table 3. Results of BPHSA-MLP, HSA-MLP and BP-MLP on Glass Dataset

	MSE	Median	Std. Dev.	Accuracy	Time (s)	Epoch
BPHSA-MLP	0.01493	0.0191	0.005922795	96.06	61.2	5000
HSA-MLP	0.01634	0.0175	0.004010985	91.5	80.4	5000
BP-MLP	0.037406	0.0445	0.01156058	93.96	Stopped at 24 Sec	Not converged

The experimental results presented in Table 3 proved that both BPHSA-MLP and HSA-MLP have reached the maximum learning iteration which is 5000 with 61.2 and 8.4 seconds, respectively. However, the BPHSA-MLP has outperformed HSA-MLP in terms of MSE and correct classification rate. On the other hand, BP-MLP has stopped the training due to the steady state parameter to prevent the network being trapped at the local minima. Conversely the scheme, BPHSA-MLP, showed its ability to escape from local minima.

Table 4. Results of BPHSA-MLP, HSA-MLP and BP-MLP on Wine Dataset

	MSE	Median	Std. Dev.	Accuracy	Time (s)	Epoch
BPHSA-MLP	0.004992	0.00499	0.00000447214	98.66	7	439
HSA-MLP	0.01138	0.00723	0.011862375	96.02	40	5000
BP-MLP	0.1896	0.189	0.08726457	94.74	Stopped at 21.6 Sec	Not converged

From Table 4, it can be seen that BPHSA-MLP offers the fastest convergence rate, the smallest MSE and the highest accuracy rate compared to HSA-MLP and BP-MLP. The BPHSA-MLP has converged at iteration 439 in 7 seconds with an MSE of 0.004992 andan accuracy rate of 98.66%. However, the HSA-MLP reached the maximum learning iteration with an accuracy rate of 96.02% and MSE of 0.01138 in 40 seconds time. Unlike HSA-MLP, the BP-MLP has not converged to the minimum error and had stopped the training process in 21.6 seconds with MSE of 0.1896 and 94.74%accuracy rate. Hence, the BPHSA-MLP has capability of avoiding the local minima.

Table 5. Results of BPHSA-MLP, HSA-MLP and BP-MLP on Cancer Dataset

	MSE	Median	Std. Dev.	Accuracy	Time (s)	Epoch
BPHSA-MLP	0.01752	0.0204	0.006069349	97.28	75	5000
HSA-MLP	0.02342	0.0242	0.006335771	95.14	90	5000
BP-MLP	0.04896	0.0538	0.024012976	94.28	Stopped at 68.4 Sec	Not converged

For Cancer dataset both BPHSA-MLP and HSA-MLP reached the maximum learning iteration as shown in Table 6. However, BPHSA owns the smallest error rate and the shortest time as well as the highest accuracy rate in comparison to HSA-MLP and BP-MLP. But, the BP-MLP stopped the training at epoch 1576 in 68.4 seconds with 94.28% accuracy rate as well as 0.04896 MSE error rate.

Table 6. Results of BPHSA-MLP, HSA-MLP and BP-MLP on Thyroid Dataset

	MSE	Median	Std. Dev.	Accuracy	Time (s)	Epoch
BPHSA-MLP	0.0305	0.0307	0.001312631	94.3	306	5000
HSA-MLP	0.05004	0.05	0.000960729	92.44	313.8	5000
BP-MLP	0.03796	0.0336	0.007386677	93.54	Stopped at 536.4 Sec	Not converged

According to the results presented in Table 6 both BPHSA-MLP and HSA-MLP met the maximum iteration in 306 and 313.8 seconds, respectively. Although BP-MLP has been trapped in local minima at epoch 4777, it has the best correct classification rate and smallest error compared to HSA-MLP. Yet HSA-MLP is better than BP-MLP and BPHSA in std Dev. Also, it has the shortest time compared to BP-MLP. For correct classification rate, MSE and training time, BPHSA-MLP is the best among all.

The results showed that BPHSA is better than both standard BP-MLP and HSA-MLP in terms of convergence rate, training time and classification accuracy with clear results in all five datasets which shows the ability of technique towards the classifier's performance. However in some cases BP is better than both HSA and BPHSA such as shortest time of training. Yet, BPHSA-MLP outperforms BP-MLP and HSA-MLP in most cases, therefore the ability of the classifier has been increased and proposed model showed its ability to train the MLP type of FFANN. In addition, the proposed model contributed to generate accurate results and reduced the amount of error in most of the experiments of this hybrid form of BP and HSA.

In conclusion, the proposed technique has shown its capability and the validity to contribute to the learning enhancement of the feed forwardNN towards accurate results in classification problems.

6. Comparison between BP, HSA and BPHSA

In this section, a comparative analysis is carried out in order to compare the BP-MLP, HSA-MLP and BPHSA-MLP algorithms. This comparison is based on the correct classification percentage of each technique for all five datasets used in this experiment. Figure 2 demonstrates the correct classification percentage for all datasets.

For Iris, Glass, Cancer and Wine datasets, the BP-MLP ispoorest in both classification accuracy and convergence speed among these algorithms. However BPHSA-MLP has the fastest convergence speed and the highest correct classification rate with minimum error in reasonable time for aforementioned datasets. For Cancer and Thyroid datasets, results show that BPHSA-MLP has better results in classification accuracy compared to HSA-MLP and BP-MLP. Although BP-MLP has stopped learning at iteration 4777 in 68.4 seconds time, it has the least convergence error among these algorithms for Cancer dataset. Furthermore, HSA-MLP has the worst results in correct classification in comparison with BPHSA-MLP and BP-MLP for Cancer and Thyroid datasets but for the convergence speed HSA-MLP outperforms the BP-MLP in all aforementioned results of datasets.

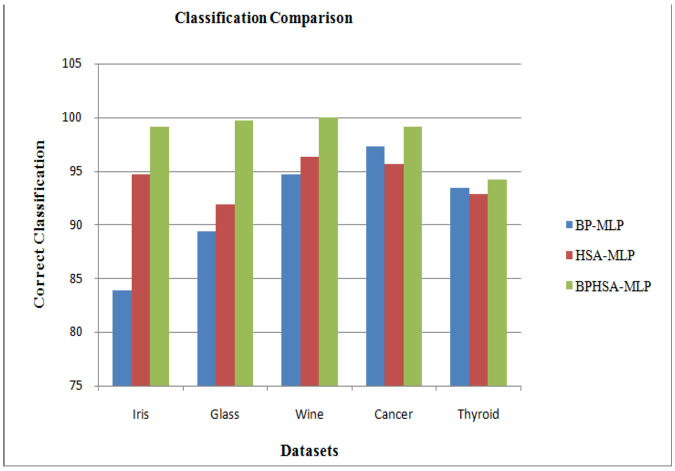

Figure 2. Correct classification percentage comparison between learning algorithms

In terms of convergence rate, BPHSA-MLP has the fastest convergence speed among these algorithms in most of these experiments with the ability of escaping local minima and reducing the errors accordingly. However, BP-MLP suffered from lack of continuity of the training process owing to the need to meet the steady state parameters in a number of specified iterations (which means errors are unchanged in this predefined iterations) which forces the BP to halt the training. HSA-MLP has been in second place in most of these experimental results. However, no local minimum has been found in HSA-MLP in all benchmark datasets. The combination of BP-MLP and HSA-MLP has solved the low performance of BP-MLP. This new scheme able to perform global search and potential to avoid local minimum problem. As a result, BPHSA-MLP has better convergence speed and the best classification accuracy.

7. Conclusion

In this paper, a hybrid Back propagation and Harmony Search Algorithm called BPHSA-MLP is employed for NNs of feed forward type to classify problems. The training performance and generalization ability of the BPHSA scheme were verified and tested using five classification benchmark datasets. The sum of squared errors, training time andaccuracy were compared with standard HSA and standard BP. The experimental results show that BPHSA-MLP scheme can successfully train feed forward type NNs with reasonable time, MSE and high accuracy. The BPHSA-MLP is better than compared algorithms in training and classification of test patterns. Therefore, BPHSA technique can be a good candidate to train NN type of feed forward for classification problems. The scheme can also be used in training both supervised and unsupervised models.

References

[1] Wei G. *Evolutionary Neural Network Based on New Ant Colony Algorithm*. Eight International Symposium on Computational Intelligence and Design. 2008; 1: 318-321.
[2] S Narayan, GA Tagliarini, EW Page. Enhancing MLP networks using a distributed data representation. *IEEE Trans. Syst., Man, Cybern. B, Cybern*. 1996; 26: 143.
[3] Castro PAD, FJ Von Zuben. *Training multilayer perceptrons with a Gaussian Artificial Immune System*. Evolutionary Computation (CEC), IEEE Congress. 2011: 1250-1257.
[4] ZW Geem, JH Kim, C Bae. Trenchless Water Pipe Condition Assessment Using Artificial Neural Network. Boston: Massachusets. 2007: 1-9.
[5] Sunila Godara, Nirmal. Intelligent and Effective Decision Support System Using Multilayer perceptron. *International journal of engineering research and applications (IJERA).* 2011; 1(3): 513-518.
[6] Katten A, Abdullahi R, Salam A. *Harmony Search based supervised training of artificial neural networks*. In proceedings of the first international conference on intelligent systems, modeling and simulation (ISMS 2010). 2010; 7(10): 105-110.
[7] Ming Z, W Lipo. *Intelligent trading using support vector regression and multilayer perceptrons optimized with genetic algorithms*. Neural Networks (IJCNN), the International Joint Conference. 2010: 1-5.
[8] Meng Joo E, L Fan. *Genetic Algorithms for MLP Neural Network parameters optimization*. Chinese Control and Decision Conference (CCDC). 2009: 3653-3658.
[9] Correa BA, AM Gonzalez. *Evolutionary Algorithms for Selecting the Architecture of a MLP Neural Network: A Credit Scoring Case*. IEEE 11th International Conference on Data Mining Workshops (ICDMW). 2011: 725-732.
[10] Liang L, L Shi-bao, et al. *A Combinatorial Search Method Based on Harmony Search Algorithm and Particle Swarm Optimization in Slope Stability Analysis*. CiSE International Conference on Computational Intelligence and Software Engineering. 2009: 1-4.
[11] Haza Nuzly Abdull Hamed SMSNS. Particle Swarm Optimization for Neural Network Learning Enhancement. *Jurnal teknologi*. 2008; 49(D): 13-26.
[12] Raul Ramos-Pollan NGdPMAGL. Optimizing the area under the ROC curve in multilayer perceptron-based classifiers. *IARIA*. 2011: 75-81.
[13] Masood Zamani, Alireza Sadeghian. A Variation of Particle Swarm Optimization for Training of Artificial Neural Networks. Computational Intelligence and Modern Heuristics, Al-Dahoud Ali (Ed.), ISBN: 978-953-7619-28-2, InTech, Available from: http://www.intechopen.com/books/computational-intelligence-and-modern-heuristics. 2010
[14] Kulluk S, L Ozbakir, et al. Training neural networks with harmony search algorithms for classification problems. *Eng. Appl. Artif.Intell*. 2011; 25(1): 11-19.
[15] Xin Sun, Qiongxin Liu, Lei Zhang. *A BP Neural Network Model Based on Genetic Algorithm for Comprehensive Evaluation. Circuits, Communications and System (PACCS).* Third Pacific-Asia Conference. 2011: 1-5.
[16] H Shayeghi, HA Shayanfar, G Azimi. A hybrid Particle Swarm Optimization Back Propagation Algorithm for Short Term Load Forecasting. *International journal on, Technical and physical problems (IJTPE)*. 2010; 4(2): 12-22.

[17] Siti Mariyam Shamsuddin, Razana Alwee, Maslina Darus. Study of Cost Functions in Three Term Backpropagation for Classification Problems. *IEEE, world congress on Nature & Biologically Inspired Computing*. 2009; 978(1): 564-570.
[18] Jing-Ru Zhang, Jun Zhang, Tat-Ming Lock, Michael R Lyu. A hybrid Particle Swarm Optimization-Back-Propagation Algorithm for Feed Forward Neural Network Training. *Elsever, Applied Mathematics and Computation*. 2007: 1026-1037.
[19] Shulin Wang, Shuang Yin, Minghui Jiang. *Hybrid Neural Network Based on GA-BP for Personal Credit Scoring. IEEE*, Fourth International Conference on Natural Computation. 2008: 209-214.
[20] Saeed Tavakoli, Ehsan Valian, Shahram Mohanna. Feedforward Neural Network Using Intelligent Global Harmony Search. Springer. 2012: 125-131.
[21] E Alba, JF Chicano. *Training neural networks with GA hybrid algorithms*. In Proc. of GECCO 2004, ser. LNCS, K. Deb et al., Ed. Springer Verlag, Berlin, Germany. 2004; 3102: 852-863.
[22] Marcio Carvalho, Teresa B Ludermir. Seventh International Conference on Hybrid Intelligent Systems, IEEE. 2007: 336-339.
[23] Masaya Yoshikawa, Kazuo Otani. *Ant Colony Optimization Routing Algorithm with Tabu Search*. Proceedings of International Multi-Conference Engineers and Computer Scientists. Hong Kong. 2010; 3(1): 1-4.
[24] Alireza Alkarzadeh, Alireza Rezazadeh. Artificial neural network training using a new efficient optimization algorithm. *Applied Soft Computing*. 2013; (13): 1206-1213.
[25] Quan-Ke Pan, PN Suganthan, M Fatih Tasgetiren, JJ Liang. A self-adaptive global best search algorithm for continuous optimization problems.Applied mathematics and computation. *Elsevier*. 2010; (216): 830-848.
[26] Abdirashid Salad Nur, Nor Haizan Mohd Radzi, Ashraf Osman Ibrahim. Artificial Neural Network Weight Optimization: A Review. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2014; 12(9): 6897-6902