AN APPROACH TO IMPROVE FUNCTIONAL LINK NEURAL NETWORK TRAINING USING MODIFIED ARTIFICIAL BEE COLONY FOR CLASSIFICATION TASK

YANA MAZWIN MOHMAD HASSIM
ROZAIDA GHAZALI

ABSTRACT

Classification is one of the most frequent studies in the area of Artificial Neural Network (ANNs). One of the best known types of ANNs is the Multilayer Perceptron (MLP). However, MLP usually requires a rather large amount of available measures in order to achieve good classification accuracy. To overcome this, a Functional Link Neural Networks (FLNN), which has single layer of trainable connection weight is used. The standard method for tuning the weight in FLNN is using a Backpropagation (BP) learning algorithm. Still, BP-learning algorithm has difficulties such as trapping in local optima and slow convergence especially for solving non-linearly separable classification problems. In this paper, a modified Artificial Bee Colony (mABC) is used to recover the BP drawbacks. With modifications on the employed bee's exploitation phase, the implementation of the mABC as a learning scheme for FLNN has given a better accuracy result for the classification tasks.

Keywords: Classification, Functional Link Neural Network, Artificial Bee Colony Algorithm.

INTRODUCTION

Artificial Neural Networks (ANNs) have been successfully applied to a variety of real world classification task especially in industry, business and science (Widrow, Rumelhart and Lehr, 1994; Zhang, 2000). One of the best known types of ANNs is the Multilayer Perceptron (MLP). The MLP is a model that maps sets of input data onto a set of appropriate output. MLP network is usually trained by adjusting the weight of connection between neurons. However, MLP usually requires a rather large amount of available measures in order to achieve good classification accuracy. To overcome this, a Functional Link Neural Networks (FLNN) which has single layer of trainable connection weights is used. The FLNN is a flat network without an existence of hidden layer which makes the network architecture less complicated (Misra and Dehuri, 2007).

The standard method for tuning weights in FLNN is using a Backpropagation (BP) learning algorithm. The BP-learning developed by Widrow et al. (1994) is well-known and widely used for training a Neural Networks. It is a type of supervised learning method where the neural network model will "learn" through a sample set of data called training set. The training set provides the network with examples of inputs and desired outputs to be computed and the error (difference between actual and expected results) is propagated backward to calculate the gradient error in order to modify the weights. The idea of the BP-learning algorithm is to reduced error, until the networks learned the training data. The training began with random weights, and the goal is to make adjustment until the minimal error is achieved. However, one of the crucial problems with the standard BP-learning algorithm is that the

gradient search techniques tend to easily get trapped in local minimum especially for non-linear separable classification problems (Dehuri and Cho, 2010).

To recover the drawback of BP-learning, the Artificial Bee Colony (ABC) optimization algorithm is used to optimize the FLNN weights. The ABC algorithm was originally proposed by Karaboga (2005) for solving numerical optimization problem by simulating the intelligent foraging behavior of a honey bee. In this study, a modified ABC is used for training the FLNN. The modification is implemented on the employed bees' foraging behavior to improve the FLNN network ability on searching the optimal weights set for better accuracy results on classifying the out-of-sample or unseen data.

FUNCTIONAL LINK NEURAL NETWORK

Functional Link Neural Network (FLNN) is a class of Higher Order Neural Networks (HONNs) that utilize higher combination of its inputs (Pao, 1989; Patra and Bornand, 2010). It was created by Klassen and Pao (1989) and has been successfully used in many applications such as system identification (Patra and Bornand, 2010; Patra and Kot, 2002; Abbas, 2009); channel equalization (Patra and Pal, 1995), classification (Raghu, Poongodi and Regnanarayana, 1995; Abu-Mahfouz, 2005; Liuet al., 1994; Dehuri and Cho, 2010), pattern recognition (Klaseen and Pao, 1990; Park and Pao, 2000) and prediction (Majhi, Panda and Sahoo, 2009; Ghazali, Hussain and Liatsis, 2011). FLNN is more modest than MLP as it has a single-layer of trainable weights compared to the MLP whilst able to handle a non-linear separable classification tasks. The flat architecture of FLNN also make the learning algorithm in the network less complicated (Misra and Dehuri, 2007). In order to capture non-linear input-output mapping for a classification task, the input vector of FLNN is extended with a suitable enhanced representation of the input nodes which artificially increase the dimension of input space (Pao, 1989; Pao and Takefuji, 1992).

The focus of this paper is on FLNN with generic basis architecture. This architecture uses a tensor representation. Figure 1 depicts the functional link neural network structure up to the second order with 3 inputs. The first order of the network consist of the 3 inputs x_1, x_2 and x_3, while the second order of the network is the extended input based on the product unit x_1x_2, x_1x_3, and x_2x_3. The learning part of this architecture on the other hand, consists of a standard BP-learning as the training algorithm.
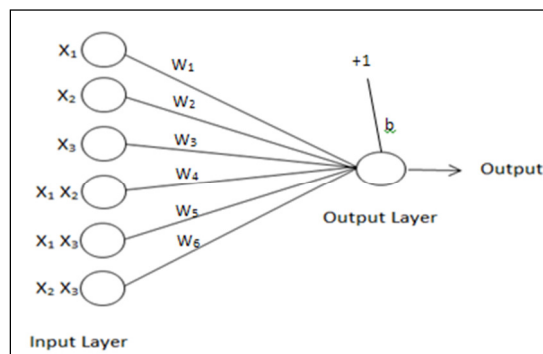

FIGURE 1. The 2nd order FLNN structure with 3 inputs

FLNN LEARNING SCHEME

Most previous learning algorithm used in the training of FLNN, is the BP-learning algorithm (Misra and Dehuri, 2007; Dehuri and Cho, 2010; Abu-Mahfouz, 2005;Ghazali, Hussain and Liatsis, 2011; Haring and Kok, 1995; 1997;Sierra, Macias and Corbacho, 2001;Dehuri, Mishra and Cho, 2008). As shown in Figure 1, the weight values between enhanced input

nodes and output node are randomly initialized. The output node, $\hat{y}$ of FLNN would correspond to the input pattern x and the number of input patterns, n. For tensor representation with single output node, the enhanced input can be noted as n+n(n-1)/2. Let the enhanced input node of tensor x be represented as $x_t = \langle x_1, x_2, \ldots x_n, x_1 x_2, x_1 x_3, \ldots x_{n-1} x_n \rangle$ and let f denotes the output node's activation function (logistic sigmoid) as per in this work:

$$f(s) = \frac{1}{1+\exp(-s)} \quad , where \quad s = wx_t + b \tag{1}$$

The output value of the FLNN is obtained by:

$$\hat{y} = f(s) \tag{2}$$

where $\hat{y}$ is the output while f denotes the output node activation function and b is the bias. In Eq. (1), wx_t is the aggregate value which is the inner product of w and x_t. The square error E, between the target output and the actual output will be minimized as:

$$E = \frac{1}{2}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3}$$

Where y_i is the target output and $\hat{y}_i$ is the actual output of the ith input training pattern, while n is the number of training pattern. During the training phase, the BP-learning algorithm will continue to update w and b until the maximum epoch or the convergent condition is reached.

Although BP-learning is the mostly used algorithm for FLNN training, the algorithm however has several limitations; it tends to easily gets trapped in local minimum especially when dealing with non-linear separable classification problems. The convergence speed of the BP learning also can gets too slow even if the learning goal and a given termination error achieved. Besides, the convergence behavior of the BP-learning algorithm is very dependable on the choices of initial values of the network connection weights as well as the parameters in the algorithm such as the learning rate and momentum (Dehuri and Cho, 2010).

STANDARD ARTIFICIAL BEE COLONY OPTIMIZATION

The Artificial Bees Colony (ABC) algorithm is an optimization tool, which simulates the intelligent foraging behavior of a honey bee swarm for solving multidimensional and multimodal optimization problem (Karaboga, 2005). In this model, three groups of bees which are employed, onlooker and scout bees determined the objects of problems by sharing information to one another. The employed bee uses random multidirectional search space in the Food Source area (FS). They carry the profitability information (nectar quantity) of the FS and share this information with the onlookers. Onlooker bees evaluate the nectar quantity obtained by the employed and bees and choose FS depending on the probability value base on the fitness. If the nectar amount of FS is higher than that of the previous one in their memory, they memorize the new position and forget the previous one (Karaboga, 2005). The employed bee whose food source has been abandoned becomes a scout and starts to search for finding a new food source randomly. The following is the standard ABC pseudo code:

```
1) Cycle = 0
2) Initialization population of scout bee with random solution x_i,j i=1,2…FS
3) Evaluate fitness of the population
4) Cycle = 1:MCN
5) form new population v_i,j for the employed bees using:
```

$$v_{i,j} = x_{i,j} + \Phi_{i,j}(x_{i,j} - x_{k,j}) \tag{4}$$

Where k is a random selected solution in the neighbourhood of i, Φ is a random
number in the range [-1,1] while j is a random selected dimension vector in i
and evaluate them

6) Apply greedy selection between $v_{i,j}$ and $x_{i,j}$
7) Calculate the probability values p_i for the solutions x_i using:

$$p_i = \frac{fit_i}{\sum_{i=1}^{FS} fit_i} \; where \; fit_i = \begin{cases} \frac{1}{1+f_i}, & if \, f_i \geq 0 \\ 1 + abs(f_i), & if \, f_i < 0 \end{cases} \quad (5)$$

8) Produce the new solutions v_i for the onlookers from the solutions x_i selected
depending on p_i and evaluate them
9) Apply the greedy selection process for onlookers
10) Determine the abandoned solution for the scout, if exists, and replace it
with a new randomly produced solution x_i using

$$x_i^j = x_{min}^j + rand(0,1)(x_{max}^j - x_{min}^j) \quad (6)$$

11) Memorize the best solution
12) cycle=cycle+1
13) Stop when cycle = Maximum cycle number (MCN).

FLNN WITH MODIFIED ABC LEARNING SCHEME

In this study, a modified ABC (mABC) is used as a learning scheme for training the FLNN. The modification is done on the part of employed bees' foraging phase so that they would exploit all weights and biases in the FLNN architecture in order to improve the network ability on searching the optimal weights set. In standard ABC algorithm, the position of a food source (FS) represents a possible solution to the optimization problem, and the nectar amount of a food source corresponds to the profitability (fitness) of the associated solution. In the case of training the FLNN with ABC, the weight, w and bias, b of the network are treated as optimization parameters to the optimization problem (finding minimum Error, E) presented in Eq. (3). The FLNN optimization parameters is treated as D-dimensional vector for the solution $x_{i,j}$ where $(i = 1, 2, ..., FS)$ and $(j = 1, 2, ..., D)$ and each vector i is exploited by only one employed bee. In order to produce a candidate food source $v_{i,j}$ from the old one $x_{i,j}$ in memory, the ABC uses Eq. (4) where $k \in \{1,2, ..., FS\}$ and both k and j are a randomly chosen indexes. The food source of $x_{i,j}$ can be represented in a form of $X = FS \times D$ matrix.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,D} \\ \vdots & \ddots & \vdots \\ x_{FS,1} & \cdots & x_{FS,D} \end{bmatrix} \quad (7)$$

As can be seen from Eq. (4) and matrix representation from Eq. (7), for each row of FS only one element from D will be chosen randomly and exploited by the employed bee by using:

$$j = fix(rand * D) + 1; \quad (8)$$

However in the case of FLNN mainly for classification tasks which are always deal with large number of optimization parameters (weights + bias), exploiting one element in each solution vector x_i will cause longer foraging cycle in finding the optimal solution (Mohmad Hassim and Ghazali, 2013). Random selection of elements in each vector x_i during employed bee phase also leads to a poor ability for FLNN network in finding the optimal weights set which result to a low classification accuracy on unseen data (Mohmad Hassim and Ghazali, 2012).

To overcome this, we eliminate the random employed bee behavior in selecting the elements in vector dimension as in Eq. (8). In the other hand, we direct the employed bee to visit all elements in D to exploit them before evaluating the vector x_i. The modified ABC is performed as shown in pseudo code below, where the box indicates the improvement made to the standard ABC:

1) Cycle = 0
2) Initialize FLNN optimization parameters, D
3) Initialize population of scout bee with random solution $x_i, i=1,2...FS$
4) Evaluate fitness of the population

66

```
 5) Cycle = 1:MCN
 6) Form new population (v_i)  for employed bees
     i.   select solution, k in the neighbourhood of i, randomly
    ii.   For j = 1:D
   iii.   Direct employed bee to exploit nectar value of j in population (v_i,j)
          using Eq. (4) where j = 1,2,...,D is a dimension vector in i
    iv.   j= j+1;
     v.   exit  loop when j = D;
 7) evaluate the new population (v_i)
 8) Apply greedy selection between v_i and x_i
 9) Calculate the probability values p_i for the solutions x_i using Eq. (4)
10) Produce the new solutions v_i for the onlookers from the solutions x_i selected
    depending on p_i and evaluate them
11) Apply the greedy selection process for onlookers
12) Determine the abandoned solution for the scout, if exists, and replace it with
    a new randomly produced solution x_i using Eq. (5)
13) Memorize the best solution
14) cycle=cycle+1
15) Stop when cycle = Maximum cycle number (MCN).
```

SIMULATION RESULTS

In order to evaluate the performance of the FLNN model trained with modified ABC (FLNN-mABC) for classification problems, simulation experiments were carried out on a 2.30 GHz Core i5-2410M Intel CPU with 8.0 GB RAM in a 64-bit Operating System. The comparison of mABC algorithm with standard BP training and standard ABC algorithms is discussed based on the simulation results implemented in Matlab 2010b. In this work we considered 3 benchmark of multiclass classification problems obtained from the UCI Machine Learning Repository (Frank and Asuncion, 2010); IRIS dataset, GLASS Identification and THYROID Disease dataset.

During the experiment, simulations were performed on the training of the second order FLNN architecture with Backpropagation algorithm (FLNN-BP), second order FLNN architecture with ABC algorithm (FLNN-ABC) and second order FLNN architecture with modified ABC algorithm (FLNN-mABC). The best training accuracy for every benchmark problems were noted from these simulations. The Learning rate and momentum used for the FLNN-BP were 0.3 and 0.7 with the maximum of 1000 epoch and the minimum error=0.001 as for the stopping criteria. Parameters setup for the both FLNN-ABC and FLNN-mABC however, only involved the setting up of stopping criteria of maximum 1000 cycles and minimum error=0.001. The activation function used for the network output for both MLP and FLNN is Logistic sigmoid function. Table 1 below summarized the parameters considered in this simulation.

TABLE 1. Parameters considered for FLNN-BP, FLNN-ABC and FLNN-mABC simulation

parameters	FLNN-BP	FLNN-ABC	FLNN-mABC
Learning rate	0.3	-	-
Momentum	0.7	-	-
Maximum epoch/cycle	1000	1000	1000
Minimum error	0.001	0.001	0.001

Ten trials were performed on each simulation of the FLNN-BP, FLNN-ABC and FLNN-mABC with the best accuracy result is noted from these 10 trials. In order to generate the training and test sets, each datasets were randomly divided into two equal sets (1st-Fold

and 2nd-Fold). Each of these two sets was alternately used either as training set or as a test set. The average accuracy values of each datasets result were then used for comparison.

IRIS CLASSIFICATION PROBLEM

This is a classical classification database made famous by Fisher (Fisher, 1936), who used it to illustrate principle of discrimination analysis. The dataset consists of 4 features and 3 classes of 50 instances each, where each class refers to a type of iris plant; Iris Setosa, Iris Versicolour and Iris Virginica. The architecture of FLNN up to second order for IRIS classification Problem is 10-3. The total of trainable weights involved in the training of FLNN is 33 (weights and biases). The classification result on unseen data in term of percentage accuracy is presented as figure 2 below.

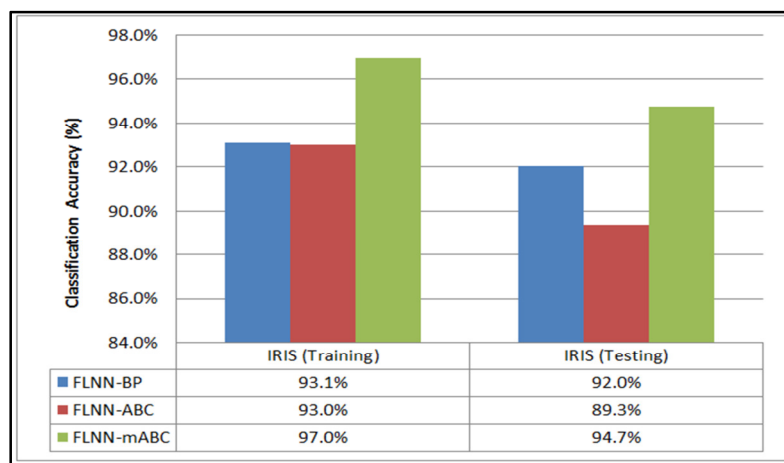|	IRIS (Training)	IRIS (Testing)
■ FLNN-BP	93.1%	92.0%
■ FLNN-ABC	93.0%	89.3%
■ FLNN-mABC	97.0%	94.7%

FIGURE 2. Iris classification accuracy by FLNN-BP, FLNN-ABC and FLNN-mABC

From figure 2, it can be seen that FLNN trained with modified ABC (FLNN-mABC) gives the best accuracy result of 94.7% on the unseen data which significantly outperformed both FLNN-BP and FLNN-ABC with the percentage difference of 2.7% and 5.4% respectively.

GLASS IDENTIFICATION PROBLEM

The glass identification database was donated by VinaSpiehler with a total of 214 instances. The dataset consists of 9 features and 6 classes, where each class refers to a type of glass: building windows float, building windows non-float, vehicle windows float, containers, tableware and headlamps. The architecture of FLNN up to second order for Glass Identification Problem is 45-6. The classification result on the unseen data in term of percentage accuracy for Glass Identification Problem is presented as figure 3 below.
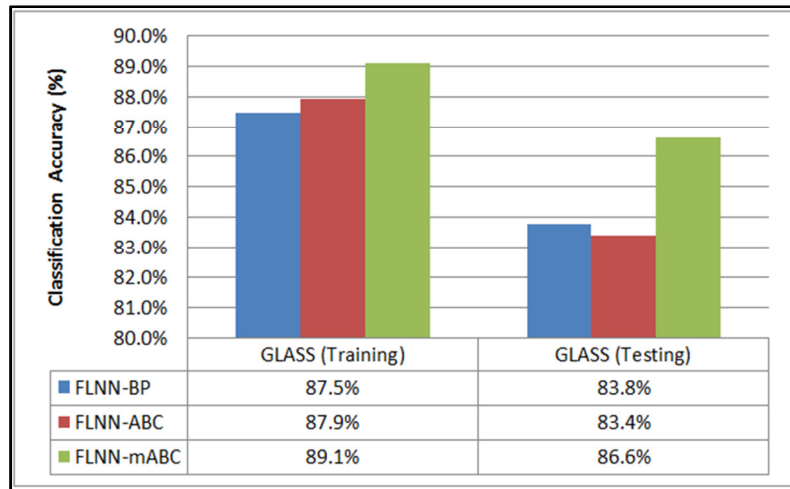
FIGURE 3. Glass classification accuracy by FLNN-BP, FLNN-ABC and FLNN-mABC

From figure 3, it is also worth noticing that FLNN trained with modified ABC (FLNN-mABC) gives 2.8% higher accuracy than FLNN-BP and 3.2% higher accuracy than FLNN-ABC in classifying an unseen data.

THYROID CLASSIFICATION PROBLEM

This dataset was created based on Thyroid Disease problem dataset donated by Stefan Aberhard from the UCI repository of machine learning database. The dataset deals with diagnosing a patient thyroid function. The dataset has 215 instances which consist of 5 features and 3 classes of thyroid functions; normal, hyper and hypo functioning. The architecture of FLNN up to second order for Glass Identification Problem is 45-6. The classification result in term of percentage of accuracy on the unseen data for Glass Identification Problem is presented as figure 4 below.
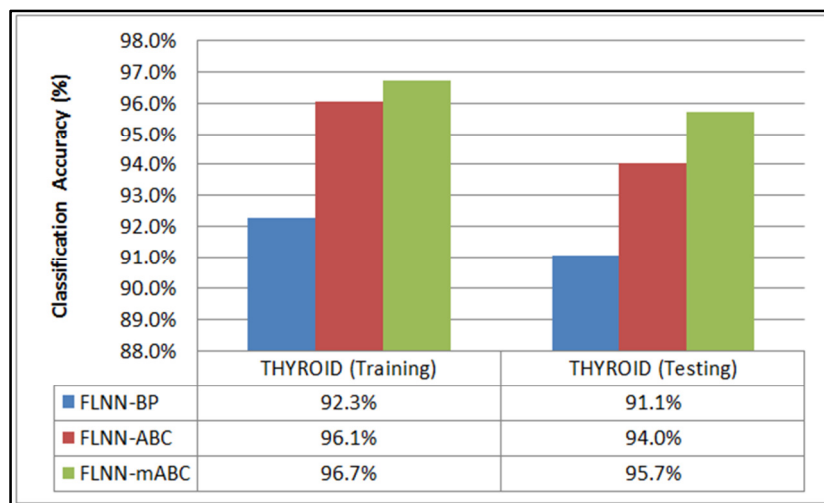

FIGURE 4. Thyroid classification accuracy by FLNN-BP, FLNN-ABC and FLNN-mABC

Figure 4, reveal that FLNN-ABC gives 94.0% accuracy on unseen data while FLNN-BP only gives 91.1% accuracy on the unseen data. Training the FLNN with standard ABC has provide the network with ability of exploitation and exploration on searching optimal weight set which give better accuracy result compared to the gradient search employed by BP-learning.

However the FLNN trained by modified ABC (FLNN-mABC) gives significantly better result with 95.7% accuracy.

According to the previous results in general, the modified ABC algorithm (mABC) offers better learning scheme for training the FLNN than the standard ABC and standard BP-learning. The experimental results from 3 classification problems have demonstrated that the proposed algorithm (mABC) performed highest accuracy instead of the standard ABC and BP-learning and may be used as alternative learning scheme for training the FLNN for classification tasks.

CONCLUSION

In this work, these authors evaluated the FLNN-mABC model for the task of pattern classification for multiclass classification problems. The experiment has demonstrated that FLNN-mABC performs the classification task quite well. For the case of IRIS, GLASS and THYROID, the simulation result shows that the proposed modified ABC algorithm can successfully train the FLNN for solving classification problems with better accuracy percentage on unseen data. This research work is hopefully would provide a better training scheme for the FLNN in order to give better classification accuracy on unseen data.

ACKNOWLEDGEMENT

REFERENCES

Abbas, H.M. 2009. System Identification Using Optimally Designed Functional Link Networks via a Fast Orthogonal Search Technique. *Journal of Computers,* 4(2): 195-202.

Abu-Mahfouz, I.-A. 2005. A comparative study of three artificial neural networks for the detection and classification of gear faults *International Journal of General System,s* 34(3):61-277.

Dehuri, S. and Cho, S.-B. 2010. A comprehensive survey on functional link neural networks and an adaptive PSO–BP learning for CFLNN. *Neural Computing & Applications,*19(2):187-205.

Dehuri, S. and Cho, S.B. 2010. Evolutionarily optimized features in functional link neural network for classification. *Expert Systems with Applications,* 37(6): 4379-4391.

Dehuri, S., Mishra, B.B. and Cho, S.B.2008. Genetic Feature Selection for Optimal Functional Link Artificial Neural Network in Classification. *Proceedings of the 9th International Conference on Intelligent Data Engineering and Automated Learning.* Daejeon, South Korea: Springer-Verlag, 156-163.

Fisher, R.A. 1936. The use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics,* 7(2):179-188.

Frank, A. and Asuncion, A. 2010. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml [12 March 2010].

Ghazali, R., Hussain, A.J. and Liatsis, P. 2011. Dynamic Ridge Polynomial Neural Network: Forecasting the univariate non-stationary and stationary trading signals. *Expert Systems with Applications,* 38(4): 3765-3776.

Haring, S. and Kok, J. 1995. Finding functional links for neural networks by evolutionary computation. *Proceedings of the fifth Belgian–Dutch conference on machine learning,* Brussels, Belgium:T. Van de Merckt, et al. (Eds.) BENELEARN'95,71–78.

Haring, S., Kok, J. and Van Wesel, M. 1997. Feature selection for neural networks through functional links found by evolutionary computation. *In: ILiu X et al (eds) Advances in intelligent data analysis (IDA-97). LNCS 1280*: 199–210.

Karaboga, D. 2005. An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report-TR06,Erciyes University, Engineering Faculty, Computer Science Department, Kayseri/Turkiye.

Karaboga, D., Akay, B. and Ozturk, C.2007. Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks. *Proceedings of the 4th international conference on Modeling Decisions for Artificial Intelligence*, Kitakyushu, Japan: Springer-Verlag, 318-329.

Klaseen, M. and Pao, Y.H. 1990. The functional link net in structural pattern recognition. *TENCON 90. 1990 IEEE Region 10 Conference on Computer and Communication Systems.* Hong Kong, China: IEEE, 567-571.

Liu, L.M., Manry, M.T., Amar, F., Dawson, M.S. and Fung, A.K. 1994. Image classification in remote sensing using functional link neural networks. *Proceedings of the IEEE Southwest Symposium onImage Analysis and Interpretation.*Dallas, Texas: IEEE, 54-58.

Majhi, R., Panda, G. and Sahoo, G. 2009. Development and performance evaluation of FLANN based model for forecasting of stock markets. *Expert Systems with Applications,* 36(3, Part 2): 6800-6808.

Misra, B.B. and Dehuri, S. 2007. Functional Link Artificial Neural Network for Classification Task in Data Mining. *Journal of Computer Science,* 3(12): 948-955.

Mohmad Hassim, Y.M. and Ghazali, R. 2012. Training a Functional Link Neural Network Using an Artificial Bee Colony for Solving a Classification Problems. *Journal of Computing Press, NY, USA,* 4(9): 110-115.

Mohmad Hassim, Y.M. and Ghazali, R. 2013. Using Artificial Bee Colony to Improve Functional Link Neural Network Training. *Applied Mechanics and Materials,* 263: 2102-2108.

Namatame, A. and Veda, N. 1992. Pattern classification with Chebyshev neural network. *International Jounal of Neural Network,* 3: 23–31.

Pao, Y.H. 1989. *Adaptive pattern recognition and neural networks*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Pao, Y.H. and Takefuji, Y. 1992. Functional-link net computing: theory, system architecture, and functionalities. *Computer,* 25(5): 76-79.

Park, G.H. and Pao, Y.H. 2000. Unconstrained word-based approach for off-line script recognition using density-based random-vector functional-linknet. *Neurocomputing,* 31(1–4): 45-65.

Patra, J.C. and Bornand, C. 2010. Nonlinear dynamic system identification using Legendre neural network. *The 2010 International Joint Conference onNeural Networks (IJCNN)*, Barcelona, Spain: IEEE, 1-7.

Patra, J.C. and Kot, A.C. 2002. Nonlinear dynamic system identification using Chebyshev functional link artificial neural networks. *IEEE Transactions onSystems, Man, and Cybernetics, Part B:Cybernetics,* 32(4): 505-511.

Patra, J.C. and Pal, R.N. 1995. A functional link artificial neural network for adaptive channel equalization. *Signal Processing,* 43(2): 181-195.

Raghu, P.P., Poongodi, R. and Yegnanarayana, B. 1995. A combined neural network approach for texture classification. *Neural Networks,* 8(6): 975-987.

Sierra, A., Macias, J.A. and Corbacho, F. 2001. Evolution of functional link networks. *IEEE Transactions onEvolutionary Computation,* 5(1): 54-65.

Widrow, B., Rumelhart, D.E. and Lehr, M.A. 1994. Neural networks: applications in industry, business and science. *Commun. ACM,* 37(3): 93-105.

Zhang, G.P. 2000. Neural networks for classification: a survey. *IEEE Transactions onSystems, Man, and Cybernetics, Part C: Applications and Reviews,* 30(4): 451-462.

Yana Mazwin Mohmad Hassim,
Rozaida Ghazali.
Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia,
Johor, Malaysia.
yana@uthm.edu.my, rozaida@uthm.edu.my