MUSSELS WANDERING OPTIMIZATION ALGORITHM BASED TRAINING OF ARTIFICIAL NEURAL NETWORKS FOR PATTERN CLASSIFICATION

Ahmed A. Abusnaina[1] and Rosni Abdullah[2]

[1]*School of Computer Sciences, Universiti Sains Malaysia (USM), Malaysia, abusnaina@ymail.com*
[2]*School of Computer Sciences, Universiti Sains Malaysia (USM), Malaysia, rosni@cs.usm.my*

ABSTRACT. Training an artificial neural network (ANN) is an optimization task since it is desired to find optimal neurons' weight of a neural network in an iterative training process. Traditional training algorithms have some drawbacks such as local minima and its slowness. Therefore, evolutionary algorithms are utilized to train neural networks to overcome these issues. This research tackles the ANN training by adapting Mussels Wandering Optimization (MWO) algorithm. The proposed method tested and verified by training an ANN with well-known benchmarking problems. Two criteria used to evaluate the proposed method were overall training time and classification accuracy. The obtained results indicate that MWO algorithm is on par or better in terms of classification accuracy and convergence training time.

Keywords: Artificial neural network, Mussels Wandering Optimization, supervised training, Optimization, Evolutionary algorithms

INTRODUCTION

The artificial neural network (ANN) is an interconnected group of processing units "artificial neurons" via a series of adjusted weights; these neurons use a mathematical model for information processing to accomplish a variety of tasks such as identification of objects and patterns, making decisions based on prior experiences and knowledge and prediction of future events based on past experience (Lin, 2007; Bennett, Stewart & Beal, 2013; Dhar et. al., 2010).

ANNs are considered as a simplified mathematical approximation of biological neural networks in terms of structure and function. Basically, the most challenging aspects of ANN are: the mechanism of learning (training algorithms) that adjusts the neurons' weights values to minimize the error function (a measure of the difference between the actual ANN output and the desired output), and the mechanism of information flow that depends on ANN structure (Ghosh-Dastidar, & Adeli, 2009; Suraweera, & Ranasinghe, 2008).

The training process deals with adjusting and altering the weights and/or structure of the network depending on a specific training algorithm. The training-dataset is fed to the network in order to determine its outputs during the training process; the objective of this process is to minimize an ANN's error function.

Training ANN fall into two main categories: traditional learning algorithms and Evolutionary-based training algorithms. Gradient-based technique such as Back-propagation (BP) is the most well-known algorithm for traditional learning algorithms. This type of learning algorithms suffer from its slowness because of improper learning steps (Silva, Pacifico, & Ludermir, 2011; Huang, Zhu, & Siew, 2004). Therefore, numerous iterative learning steps are necessary for the sake of obtaining better accuracy and performance. In addition, traditional learning algorithms simply fall into local minima problem (Kattan, Abdullah, 2011).

Evolutionary-based training algorithms that depend on global optimization methods overcome the disadvantages of traditional learning algorithms (Kattan, Abdullah, 2011; Karaboga, Akay, & Ozturk, 2007). The search space of the ANN weights training process is considered as continuous optimization problem because it is high-dimensional and multimodal, also it could be corrupted by noises or missing data (He, Wu, Saunders, 2009; Karaboga, Akay, & Ozturk, 2007).

Complex optimization problems have been handled by Meta-heuristics algorithms, such as Evolutionary-based training algorithms inspired biologically, such as Genetic algorithm (GA) (Gao, Lei, & He, 2005), Artificial Bee Colony (ABC) (Karaboga, Akay, & Ozturk, 2007), Group Search Optimizer (GSO) (He, Wu, Saunders, 2009a; He, Wu, Saunders, 2009b), Particle Swarm Optimization (PSO) (Su, Jhang & Hou, 2008) and the Harmony Search (HS) algorithm which is inspired from the improvisation process of musicians (Kattan, Abdullah, 2011; Kattan, Abdullah, & Salam 2010).

Nevertheless, most of these evolutionary-based training algorithms were based on using the classical XOR problem, such as the method proposed by Karaboga et. al. (2007). So, most of these algorithms were unable to generalize its superiority against others (Kattan, Abdullah, 2011), because the training dataset size of this problem is too small. Furthermore, the XOR problem does not have local minima (Hamey, 1998).

This work proposes a new method for training feed-forward ANN by adapting MWO algorithm. The proposed method aims to minimize the ANN training time while maintaining an acceptable accuracy rate. This objective was achieved with making use of well-known benchmarking problems that considered larger and more complex than classical XOR problem.

The rest of this paper is organized as follows: Section II introduces the MWO algorithm; its coefficients and equations, section III presents the proposed method and the adaptation process of MWO algorithm in ANN, section IV covers and discusses the experimental setup and results. Finally, section V presents the conclusions and future works.

MUSSELS WANDERING OPTIMIZATION ALGORITHM (MWO)

MWO is a novel meta-heuristic algorithm, ecologically inspired for global optimizations by Jing An *et. al.* (An, Kang, Wang, & Wu, 2012). MWO is inspired by mussels' movement behavior when they form bed pattern in their surroundings habitat. Stochastic decision and Le´vy walk are two evolutionary mechanisms used mathematically to formulate a landscape-level of mussels' pattern distribution.

The population of mussels consists of N individuals, these individuals are in a certain spatial region of marine ''bed'' called the habitat. The habitat is mapped to a d-dimensional space S^d of the problem to be optimized, whereas the objective function value $f(s)$ at each point $s \in S^d$ represents the nutrition provided by the habitat. Each mussel has a position $x_i:=(x_{i1}, \ldots, x_{id})$; $i \in \mathrm{N}_N = \{1,2,3, \ldots, N\}$ in S^d, which therefore, they forming a specified spatial bed pattern.

The MWO algorithm is composed of six steps as follows: (1) Initialization of mussels-population and the algorithm parameters. (2) Calculate the short-range density ζs and long-range density ζ_l for each mussel. (3) Determine the movement strategy for each mussel. (4) Update the position for all mussels. (5) Evaluate the fitness of each mussel m_i after position updating. (6) Examine the termination criteria. The MWO algorithm is shown in Figure 1, where the list of equations used by MWO is presented in APPENDIX A.

(1)Initialization of mussels-population and the algorithm parameters:

1.1 Generate N mussels and placed them uniformly in space S^d.

1.2 Set the algorithm parameters: maximum generation *G*, range-references coefficients of α and β, space scale factor δ, moving coefficients *a*, *b*, and *c*, and walk scale factor γ.

1.3 For each mussel m_i evaluate its initial fitness by computing the objective function $f(x_i)$.

1.4 Using the calculated objective function $f(x_i)$ find the best mussel and record its position as x_g.

(2) Calculate the short-range density ζ_s and long-range density ζ_l for each mussel:

2.1Using all mussels' coordinate positions, compute the distances D_{ij}; $_{i,j} \in _{NN}$ between any two mussels by using Eq. (1).

2.2Compute the short-range reference r_s and long-range reference r_l by Eq.(2).

2.3For all mussels, calculate their ζ_{si} and ζ_{li} ; $i \in N_N$ by using Eq. (3) and (4), respectively.

(3) Determine the movement strategy for each mussel:

3.1Calculate the moving probability P_i of mussel m_i according to the short-range density ζ_{si} and long-range density ζ_{li} by using Eq. (5).

3.2Determine the movement, if $P_i = 1$, calculate its step length ℓ_i by Eq. (6).

(4) Update the position for all mussels:

For all mussels compute the new position coordinate $\dot{x}_i$ of mussel m_i in S^d by using Eq. (7).

(5) Evaluate the fitness of each mussel m_i after position updating:

5.1Calculate the objective function $f(\dot{x})$ for the new positions.

5.2 Rank the solutions in order to find the global best mussel m_g, and update the best record [best position x_g and optimal fitness $f^*(x_g)$].

Figure 1. MWO Algorithm

PROPOSED METHOD

In this work, the Feed-forward ANN weights (including biases) are adjusted and tuned using MWO algorithm in order to solve a given classification problem as illustrated in Figure 2. MWO is chosen to train the ANN because of its great ability to tackle the hard optimization problems. As well as, MWO parameters (specially, shape parameter μof the Le´vy distribution) can be adjusted to fit any problem. The ANN is composed of three types of layers, namely, input, hidden and output layer. Each layer contains a number of neurons, which obtain their inputs from previous neurons-layer and forward the output to their following neurons-layer.

Vector-based scheme is adapted in this work to represent the Feed-forward ANN (He, Wu, Saunders, 2009b; Kattan, Abdullah, Salam, 2010; Fish, et. al., 2004). Accordingly, each ANN is represented by a set of vectors: Input-vector, Hidden-vector, Output-vector, Weight-IH-vector, Weight-HO-vector, b_{Hidden}-vector, and b_{Output}-vector. These vectors form the complete set of ANN structure with their corresponding weights and biases.

The dataset file that includes the input patterns is read first. Then the population of mussels is generated. Each individual mussel represents a complete feed-forward ANN. After that, the MWO is applied repeatedly until the training termination condition is met. The sum squared errors (SSE) is considered as the objective function to evaluate the mussel fitness, i.e. minimizing the SSE. The bipolar-sigmoid activation function is used because of its superiority over hard-limited or linear threshold functions (Kasabov, 1998).

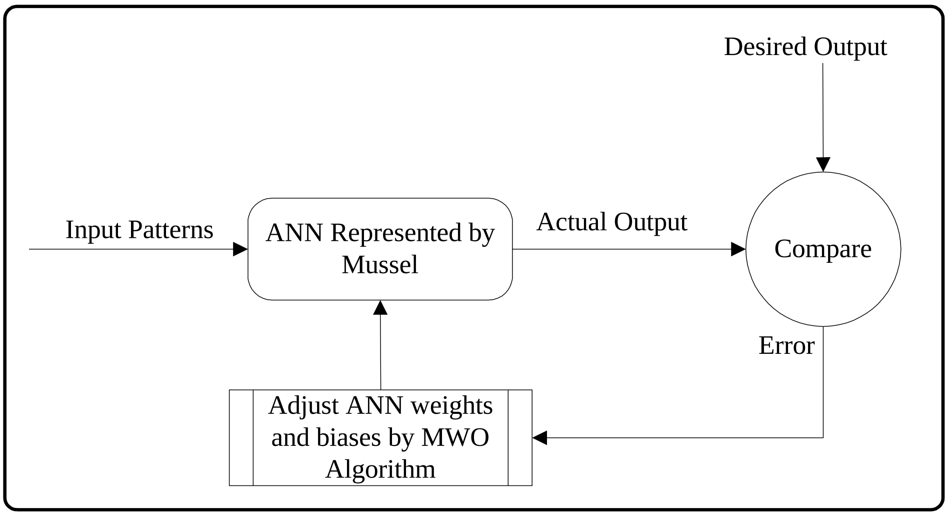

Figure 2. Schematic diagram of MWO-based training of ANN

EXPERIMENTAL SETUP AND RESULTS

In order to evaluate the MWO optimization algorithm, four widely-used benchmark classification datasets have been used. The dataset obtained from UCI Machine Learning Repository (Frank, & Asuncion 2013) namely, Ionosphere dataset, Magic dataset, Wisconsin Breast Cancer dataset and Diabetes dataset. The ANN structure was designed based on 3-layer (input-hidden-output) with varying number of neurons depending on the dataset problem as demonstrated in Table1.

During the initialization step of MWO algorithm, the collection of its coefficients and parameters values were set. These values were set as in (An, Kang, Wang, & Wu, 2012) and there were stable for all datasets and during all the training and testing sessions. The training termination condition used is the number of generation *G* is set to 100.

Ten individual sessions were conducted for each dataset, each session has two phases: training and testing phase. The best result out-of-ten approach which has been used in recent literature i.e. Kattan & Abdullah (2011) is also used in this work. The best result achieved out of ten is reported and compared against the following algorithms: Improvised Harmony Search algorithm (IHS), Back-Propagation algorithm (BP) and Genetic Adaptive Neural Network Training (GANN).

Table 1. Benchmarking datasets

Dataset	Number of Patterns	ANN Structure	Weights and Biases
Magic	9510	10-4-2	54
Ionosphere	351	33-6-2	218
Diabetes	768	8-7-2	79
Breast Cancer	699	9-8-2	98

Magic Dataset

Magic dataset is very huge, originally it includes 19,020 patterns. In order to get results in reasonable time, 50% of it has been used; i.e. number of patterns is 9510. MWO algorithm outperforms all other algorithms in terms of training time, as it needs 1.8 minute to initialize and train the whole dataset, while others suffer in this issue. On the other hand, MWO ranked second in terms of classification accuracy, as BP reports the highest accuracy with 83.9%. The result of training time and classification accuracy for Magic dataset is given in Table 2.

Ionosphere Dataset

Ionosphere dataset originally has 34 input parameters. By reviewing the input values, the second input has the same value for all patterns, for this reason the second input was omitted. MWO algorithm outperforms all other algorithms in terms of training time. The Ionosphere dataset has 351 patterns with 33 input features. In terms of classification accuracy MWO ranked last, while all other algorithms have near values. This could be as a result of Ionosphere dataset has many similar input values which made MWO algorithm not able to distinguish the different classes of this dataset clearly. In addition, the adapted MWO algorithm in this study uses the recommended coefficients and parameters in (An, Kang, Wang, & Wu, 2012); perhaps other values of these coefficients could give a better result. The result of training time and classification accuracy of ANN for Ionosphere dataset is given in Table 2.

Wisconsin Breast Cancer Dataset

The Wisconsin Breast Cancer Dataset is another well-known benchmark classification problem. The Wisconsin Breast Cancer Dataset contains 699 instances; the dataset has only 6 missing data values, whereas these patterns were removed. MWO algorithm outperforms all other algorithms in terms of training time, as it needs three minutes to initialize and train the whole dataset, other algorithms suffer in this criterion as they need more than 10 minutes. In other respects, MWO ranked second in terms of classification accuracy in line with GA, as IHS reports the best accuracy. The result of training time and classification accuracy of ANN for Cancer dataset is given in Table 2.

Pima Indians Diabetes Dataset

The classification problem of Diabetes diagnostic is whether the patient shows positive signs of diabetes according to World Health Organization criteria. The Diabetes database has 768 patterns with 8 input features. MWO algorithm scores the best in terms of training time, where it reports less than three minutes to initialize and train the Diabetes dataset. Other algorithms suffer in this criterion with varying amount of time, whereas BP reports the worst result with more than 5.5 hours. In terms of classification accuracy, MWO ranked last, but it has very close accuracy percentage to other algorithms. The result of training time and classification accuracy of ANN for Diabetes dataset is given in Table 2.

Table 2. Training time and classification accuracy for different datasets

Algorithm	Classification Accuracy				Training Time (hh:mm:ss)			
	MWO	IHS	GANN	BP	MWO	IHS	GANN	BP
Magic	78.3%	77.39%	77.87%	83.97%	0:01: 48	1:59:13	0:48:18	4:35:
Ionosphere	85.7%	94.37%	94.37%	95.77%	0:03: 49	0:03:58	0:35:57	0:24:

Breast	98.5%	100.0%	98.5%	95.7%	0:03:00	0:10:19	0:10:30	0:27:
Diabetes	75.1%	76.6%	79.8%	78.5%	0:02:48	0:11:10	0:29:28	5:30:

MWO appears as competitive in both criteria for the different datasets, however, the achieved results of MWO are reported using the recommended values of coefficients and parameters in (An, Kang, Wang, & Wu, 2012). The shape parameter (μ) of the Le´vy distribution that used in MWO is sensitive to dataset problem; each dataset problem may has its own value that can make MWO performs well. This reason makes some degradation in MWO performance regarding classification accuracy. Perhaps other values of shape parameter (μ) could improve the classification accuracy and training time more.

CONCLUSIONS

In this paper, the Mussels Wandering Optimization (MWO) algorithm which is a novel and simple global optimization algorithm has been adapted and used to train feed-forward artificial neural networks. The pattern-classification problem has been used for algorithm testing purposes.

Two criteria are considered in the performance evaluation process; over all training convergence time and classification accuracy. The results showed that the MWO algorithm has been successfully adapted and applied to train feed-forward artificial neural networks. The results also indicate that MWO algorithm is competitive in terms of convergence time and classification accuracy compared to other algorithms, whereas MWO was better or in par with other algorithms.

The application of MWO algorithm to other pattern-classification problems such as iris datasets is currently ongoing work. In addition, the future work considers the dynamic and self-adaptive techniques of adjusting the shape parameter μof the Le´vy distribution to improve the classification accuracy.

ACKNOWLEDGMENT

This research is supported by UNIVERSITI SAINS MALAYSIA and has been funded by the Research University Cluster (RUC) grant titled by “Reconstruction of the Neural Micro circuitry or Reward-Controlled Learning in the Rat Hippocampus” (1001/PSKBP/8630022).

REFERENCES

Lin, CH. SF. (2007). Toward a New Three Layer Neural Network with Dynamical Optimal Training Performance. In: Proceedings IEEE International Conference on Systems, Man and Cybernetics. pp. 3101-3106.

Bennett, C., Stewart, R., & Beal, C.D. (2013). ANN-based Residential Water End-use Demand Forecasting Model. Expert Systems with Applications, vol. 40, pp. 1014–1023.

Dhar, V.K., Tickoo, A.K., Koul, R., & Dubey, B. P. (2010). Comparative performance of some popular artificial neural network algorithms on benchmark and function approximation problems. PRAMANA journal of physics, Indian Academy of Sciences, vol. 74, No. 2. pp. 307-324.

Ghosh-Dastidar, S., & Adeli, H. (2009). A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection. Neural Networks, vol. 22, pp.1419-1431.

Suraweera, N. P., & Ranasinghe. D. N. (2008). A Natural Algorithmic Approach to the Structural Optimisation of Neural Networks. In: Proceedings of 4th International Conference on Information and Automation for Sustainability, pp. 150 –156.

Silva, D.N.G., Pacifico, L.D.S., & Ludermir, T.B. (2011). An evolutionary extreme learning machine based on group search optimization. In: IEEE Congress of Evolutionary Computation, CEC 2011, art. no. 5949670, pp. 574-580.

Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2004). Extreme learning machine: A new learning scheme of feedforward neural networks. In: Proceedings of IEEE International Conference on Neural Networks 2, pp. 985-990.

Kattan, A., & Abdullah, R. (2011). Training of Feed-Forward Neural Networks for Pattern-Classification Applications Using Music Inspired Algorithm. (IJCSIS) International Journal of Computer Science and Information Security, vol. 9, no. 11, pp. 44-57.

Karaboga, D., Akay, B., & Ozturk, C. (2007). Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks. In: 4th International Conference on Modeling Decisions for Artificial Intelligence MDAI 2007, LNAI 4617, Springer-Verlag Berlin Heidelberg, pp. 318–329.

He, S., Wu, Q.H., & Saunders, J.R. (2009a) Group Search Optimizer: An Optimization Algorithm Inspired by Animal Searching Behavior. IEEE Transactions on Evolutionary Computation, vol. 13, no. 5, pp. 973-990.

An, J., Kang, Qi., Wang, L., & Wu, Q. (2012). Mussels Wandering Optimization: An Ecologically Inspired Algorithm for Global Optimization. Cognitive Computation, vol. 5, no. 2, pp. 188-199.

Gao, Q., Lei, K.Q.Y., & He, Z. (2005). An Improved Genetic Algorithm and Its Application in Artificial Neural Network. In: Fifth International Conference on Information, Communications and Signal Processing, pp. 357–360.

He, S., Wu, Q.H., & Saunders, J.R. (2009b). Breast cancer diagnosis using an artificial neural network trained by group search optimizer. Transactions of the Institute of Measurement and Control, vol. 31, no. 6, pp. 517–531.

Kattan, A., Abdullah, R., & Salam, R. (2010). Harmony Search Based Supervised Training of Artificial Neural Networks. In: International Conference on Intelligent Systems, *Modeling and Simulation*, pp.105-110.

Fish, K., Johnson, J, Dorsey, E., & Blodgett, J. (2004). Using an Artificial Neural Network Trained with a Genetic Algorithm to Model Brand Share. Journal of Business Research, vol. 57, pp. 79-85.

Kasabov, N., (1998). Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering. The MIT Press Cambridge, England, Second printing.

Frank, A., & Asuncion, A. (online accessed on February, 2013). UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. Available: http://archive.ics.uci.edu/ml

Su T., Jhang J., & Hou C. (2008). A hybrid Artificial Neural Networks and Particle Swarm Optimization for function approximation. International Journal of Innovative Computing, Information and Control. vol. 4, No.9. pp.2363-2374.

Hamey, L. G. C. (1998). XOR Has No Local Minima: A Case Study in Neural Network Error Surface Analysis. Neural Networks, vol. 11, pp. 669-681.

APPENDIX A:

The list of equations used in MWO algorithm (An, Kang, Wang, & Wu, 2012)

Eq. No.	Formula	Parameters Description
Eq. (1)	$D_{ij} := \|x_i - x_j\| = \left[\sum_{k=1}^{d} (x_{ik} - x_{jk})^2\right]^{1/2}$ $i, j \in N_N$	D_{ij}: spatial distance between mussels m_i and m_j in S^d. N: number of mussels.
Eq. (2)	$\begin{cases} r_s(t) := \alpha \, . \, max_{i,j \in N}\{D_{ij}(t)\}/\delta \\ r_l(t) := \beta \, . \, max_{i,j \in N}\{D_{ij}(t)\}/\delta \end{cases}$	r_s: short-range reference. r_l: long-range reference. α and β are positive constant coefficients with $\alpha<\beta$. $max_{i,j \in N}\{D_{ij}(t)\}$is the maximum distance among all mussels at iteration t. δ: scale factor of space, which depends on the problem to be solved.
Eq. (3)	$\zeta_{si} := \#(D_i<r_s) / (r_s N)$	ζ_{si}: short-range density, ζ_{li}: long-range density. Where $\#(A < b)$ is used to compute the count in set A satisfying $a<b$; $a\in$A; D_i is the distance matrix from mussel m_i to other mussels in the population.
Eq. (4)	$\zeta_{li} := \#(D_i<r_l) / (r_l N)$	
Eq. (5)	$P_i := \begin{cases} 1, \text{ if } a - b\zeta_{si} + c\zeta_{li} > z \\ 0, \; otherwise \end{cases}$	a, b, and $c$are positive constant coefficients. $z$is a value randomly sampled from the uniform distribution [0,1].
Eq. (6)	$\ell_i := \gamma \, [1 - \text{rand}()]^{-1/(\mu-1)}$	ℓ_i: step length , μ : is the shape parameter, which it is known as the Le´vy exponent or scaling exponent that determines the movement strategy; $1.0 < \mu < 3.0$. γ : the walk scale factor.
Eq. (7)	$\grave{x}_i := \begin{cases} x_i + \ell_i \Delta_g, \; if \; P_i = 1 \\ x_i, \qquad if \; P_i = 0 \end{cases}$	$\grave{x}_i$: new mussel-position coordinate. x_i : current mussel-position coordinate. x_g : best mussel-position coordinate. $\Delta_g = x_i - x_g$.