

# Artificial Neural Network Training using Fireworks Algorithm in Medical Data Mining

Ram Kinkar Dutta, Nabin Kanti Karmakar and Tapas Si  
Department of Computer Science & Engineering  
Bankura Unnayani Institute of Engineering  
Bankura, West Bengal, India

## ABSTRACT

This paper proposes a novel application of Fireworks Algorithm in Artificial Neural Network training. Fireworks Algorithm is a recently developed Swarm Intelligence algorithm for function optimization. Fireworks Algorithm mimics the explosion process of fireworks. In this paper, Fireworks Algorithm is applied in training of Multi-Layer Perceptron for classification task in medical data mining. The classification task is carried out on 5 well-known medical data sets from UCI machine learning repository. A comparative study has been made with classical optimization algorithm Levenberg-Marquardt method and another Swarm Intelligence algorithm Particle Swarm Optimizer. The experimental results show that the proposed method performs better than other algorithms in classification.

## General Terms

Medical Data Mining, Machine Learning

## Keywords

Fireworks algorithm, Artificial neural network, Medical data mining, Classification

## 1. INTRODUCTION

Artificial Neural Network (ANN) [4] is modeled on human brain and it is an important machine learning tool. In medical data mining, ANN is widely and successfully used for disease diagnosis of the patients [1]. In computational model, diagnosis is the classification of disease status in terms of *abnormal* or *positive*, *normal* or *negative* or *intermediate stages*. Multi-Layer Perceptron (MLP) [4] is generally trained with well-known Back-Propagation (BP) [4] algorithm. The main drawbacks of BP algorithm is that it has slow convergence speed and it gets stuck in local optima of the error function. Recently, Levenberg-Marquardt (LM) algorithm is used instead of BP algorithm to optimize the synaptic weights of MLP. LM algorithm has stable and fast convergence for small- and medium-sized networks and patterns. But, for large-sized networks and training patterns, the training time is very high. Evolutionary algorithms are used as an alternative of BP algorithm in ANN training and evolutionary algorithm performs better than BP algorithm [9, 10]. Particle Swarm Optimization (PSO) algorithm [2] and its different variants are successfully applied in ANN training.

Y.S. Lee et al. [11] used bounded PSO-Vmax function in Feed-Forward Neural Network (FFNN) training. Though PSO algorithm has quick convergence speed, it often gets stuck in local optima due to lack in diversity. H. Shah and R. Ghazali [7] proposed an MLP training method using an improved Artificial Bee Colony (ABC) algorithm for prediction of earthquake magnitude. T. Si et al. [6] used Differential Evolution (DE) and its one variant in MLP training for real world data classification. Medical data classification is an important tasks in medical data mining for predicting the disease status. In this paper, the Fireworks Algorithm is used in MLP training for medical data classification. The classification tasks are performed over five well-known medical data sets collected from UCI machine learning repository [15]. The MLP Neural Network is trained and tested with K-fold cross-validation of data set. A comparative study of performances has been made with classical LM algorithm [5] and PSO-W algorithm which is a modified PSO algorithm [3]. Finally, the experimental results demonstrate that the proposed method performs better in classification than other algorithms used in this paper.

## 2. FIREWORKS ALGORITHM

Y. Tan and Y. Zhu [12] proposed a new Swarm Intelligence (SI) based optimization technique Fireworks Algorithm (FA/FWA) in the year 2010. Fireworks Algorithm is inspired by fireworks explosion. There are two types of sparks in FA and they are 'Explosion' spark and 'Gaussian' spark. Each fireworks and sparks represent solutions of the optimization problem. A well trade-off is employed between exploration and exploitation in the search process. Exploration is created using the bad fireworks having lower fitness and exploitation is created using good fireworks having higher fitness. Each firework generates multiple numbers of sparks with different magnitudes of explosion. Fireworks with higher fitness generate higher numbers of sparks with lower magnitude by which exploitation is achieved. On the other hand, fireworks with lower fitness generate small numbers of sparks with higher magnitude by which exploration is achieved. The number of generated sparks is controlled by the maximum number of sparks (M) and amplitude of explosion is controlled by the maximum amplitude parameters (A). Gaussian mutation is used for keeping diversity in the search space. Finally, a set of best locations is selected from all fireworks and sparks for the next iteration. In this work, an improve FWA with best selection strategy, called as IFWABS [13] is applied in MLP training. In IFWABS algorithm, a transfer function is used to

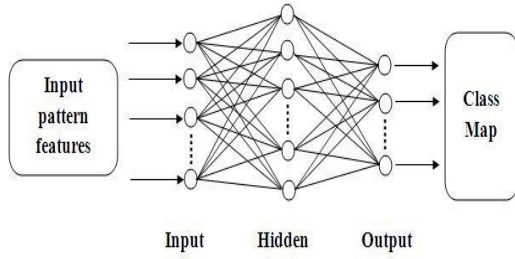


Fig. 1. An MLP neural network

generate the number of explosion sparks and their amplitude. The transfer function is as follows:

$$f(i) = \frac{1}{1 + e^{(i-1)/a}} \quad (1)$$

Where  $i$  is the index of the fireworks after sorting in ascending order according to their objective values. The control parameter  $a$  is used to change the shape of the transfer function  $f(i)$ . The control parameter  $a$  is decreased from  $a_{max}$  to  $a_{min}$ . The following linearly decreasing equation [8] is used:

$$a(t) = a_{max} - (a_{max} - a_{min}) \times \left( \frac{fitcount}{FEs} \right) \quad (2)$$

where  $fitcount$  is the number of function evaluations so far in the beginning of the current iteration  $t$  and  $FEs$  is the maximum number of function evaluations. By decreasing  $a$  with increasing iteration, the number of explosion sparks and their amplitudes are dynamically changed during the search. The number of explosion sparks is generated by the following equation:

$$S_i = M \times \frac{f(i)}{\sum_{i=1}^N f(i)} \quad (3)$$

The amplitude of explosion sparks is generated by the following equation:

$$A_i = A \times \frac{f(N - i + 1)}{\sum_{i=1}^N f(i)} \quad (4)$$

Where  $A = (X_{max} - X_{min})$  and  $[X_{max}, X_{min}]$  is the search space range. To create the explosion sparks from a firework  $X_i$ , the number of positions in location is randomly selected first and the amount of perturbation is then calculated as follows:

$$\delta x = A_i \times rand(-1, 1) \quad (5)$$

In order to maintain the diversity in search space, random mutation operator is used in fireworks as follows:

$$x_{ij} = X_{min} + (X_{max} - X_{min}) \times rand(0, 1) \quad (6)$$

After creation of all kinds of sparks, the best selection scheme is used to select the best  $N$  solutions from all fireworks and sparks. The IFWABS algorithm is given in Table 1.

### 3. ANN TRAINING USING IFWABS ALGORITHM

Artificial Neural Network is a mathematical model of biological neural network and simulates the behaviour of the human brain.

Table 1. IFWABS Algorithm

1.	Generate a set of $N$ initial locations $\mathbf{X}$ for fireworks with dimension $D$ .
2.	Calculate objective function $F(X_i)$ for each firework where $i = 1, 2, 3, \dots, N$ .
3.	<b>while</b> ( termination criteria )
4.	Sort the fireworks in ascending order based on their objective function values
5.	Calculate the transfer function values of all fireworks using Eq.( 1).
6.	Calculate the number of explosion sparks $S_i$ and amplitude $A_i$ for each $i^{th}$ firework using Eq.(3) & Eq.(4) respectively. // Generate Explosion Sparks
7.	<b>for</b> $i = 1$ to $N$
8.	Initialize the location of spark: $\hat{x}_e = X_i$
9.	Select the number of positions $l$ randomly.
10.	Calculate the perturbation $\delta x$ using Eq.(5).
11.	<b>for</b> $j = 1$ to $l$
12.	Select the index $k \in [1, D]$ randomly.
13.	$\hat{x}_{ek} = \hat{x}_{ek} + \delta x$
14.	<b>End for</b>
15.	<b>End for</b> // Perform random mutation
16.	<b>for</b> $i = 1$ to $N'$
17.	Select the firework $X_r$ , $r \in [1, N]$ randomly
18.	Initialize the location of muted firework: $\hat{x}_m = X_r$ select the number of positions $l$ randomly.
19.	<b>for</b> $j = 1$ to $l$
20.	Select the index $k \in [1, D]$ randomly.
21.	Perform mutation using Eq.(6).
22.	<b>End for</b>
23.	<b>End for</b>
24.	<b>End for</b>
25.	Select the best $N$ locations from all fireworks, sparks and muted fireworks.
26.	<b>End while</b>

All the attributes in the data set are used as input to MLP Neural Network (see Figure 1). The basic architecture of MLP consists of three types of neuron layers: input, hidden and output layers. In feed-forward networks, the signal flows from input to output units, strictly in a feed-forward direction. The basic processing elements of neural networks are called artificial neurons or nodes. Each processing node, except the input layer nodes, calculates a weighted sum of the inputs from the preceding layer to which it is connected. This weighted sum passes through the transfer ( or activation) function to derive its output which is fed as input to the nodes in the next layer. Thus, the net input to node  $j$  is obtained as

$$net_j = \sum_i w_{ij} o_i + bias_j \quad (7)$$

and output as

$$o_j = F_a(net_j) \quad (8)$$

where  $w_{ij}$  is the synaptic weight for the connection from node  $i$  to  $j$ ,  $o_j$  is the output of node  $j$ , and  $F_a(\cdot)$  is the activation function. Here, sigmoid function[4] is used as an activation function and it is defined as

$$F_a(net_j) = \frac{1}{1 + e^{(-net_j)}} \quad (9)$$

The number of input nodes in the input-layer is equal to the number of attributes in the data set. The number of nodes in the output-layer is equal to the number of classes in the data set.

In this work, MLP is trained using IFWABS algorithm to search the synaptic weight coefficients of a feed forward neural network as well as to minimize the mean-square-error in the error surface. An MLP having  $n$  input nodes in input-layer,  $(2n+1)$  hidden nodes in the hidden-layer and  $m$  output nodes in output-layer is used in this work. Mean Square Error ( $\xi$ ) is calculated by the Eq.(10) and it is used as a fitness function for fireworks in IFWABS algorithm.

$$\xi = \frac{1}{n.m} \sum_{i=1}^n \sum_{j=1}^m (t_{ij} - o_{ij})^2 \quad (10)$$

where  $i$  is a training pattern and  $j$  is the output node.  $o_{ij}$  denotes the predicted output of node  $j$  when the training pattern  $i$  is applied to the network,  $t_{ij}$  is the corresponding target output,  $n$ =number of training samples and  $m$ =number of outputs. For the outputs, a binary 1-of- $m$  encoding is used in which each bit represents one of the  $m$ -possible output classes of the problem definition. Only the correct output class carries a  $(1 - \epsilon)$ , whereas all others carry  $\epsilon$ (= 0.1) and *winner-takes-all* policy is adopted.

Total number of weight coefficients including bias in the MLP is the dimension of the firework's location. These weight coefficients are initialized in the interval  $[-1, 1]$  randomly with uniform distribution and treated as firework's location in fireworks algorithm. Each and every firework represents a neural network which is trained with the complete training set. After termination of the fireworks algorithm, the best neural network is used to classify the unknown test data.

## 4. EXPERIMENTAL SETUP

### 4.1 Data Set

In this work, five different data sets such as Cleveland heart, Wisconsin Diagnostic Breast Cancer (WDBC), PIMA Indians Diabetes, Lung Cancer and Chronic Kidney Disease data are used for classification task and they are collected from UCI machine learning repository[15]. The missing values are replaced by *attribute mean value* [14] and the data are normalized between  $[0, 1]$ . The detail description of data sets can be obtained from Ref. [15].

### 4.2 K-Fold Cross Validation

Cross-validation [14] is a very popular technique for estimating generalization error for any machine learning scheme and there are several versions. K-fold cross-validation is used in this work to obtain a reliable estimate of classifier accuracy. In each trial of K-fold cross validation, the training set is randomly split into  $K$  (here  $K = 10$ ) mutually exclusive subsets (folds) of approximately equal size. The MLP is trained using  $(K - 1)$  of the folds and tested on the fold left out. The mean of the test accuracies over  $K$  trials gives an estimate of the expected generalization performance.

### 4.3 Parameters Setting

The parameters of IFWABS are set as follows:

- (1) Number of fireworks ( $N$ ): 5
- (2) Maximum number of 'Explosion Sparks' ( $M$ ) for each firework: 40
- (3) Boundary constraints ( $M_{min}, M_{max}$ ) on number of 'Explosion Sparks' for each firework:  $[2, 32]$
- (4) Magnitude of explosion ( $A$ ):  $(X_{max} - X_{min}) = 2$
- (5) Number of mutations ( $N'$ ): 5

- (6) Minimum and maximum values of control parameter in transfer function are  $a_{min} = 1$  and  $a_{max} = 21$  respectively.

The parameters of PSO-W are set as follows:

- (1) Population size ( $N$ ) = 30
- (2)  $w_{min} = 0.4, w_{max} = 0.9$
- (3)  $c_1 = c_2 = 1.44945$
- (4)  $V_{max} = (X_{max} - X_{min})$
- (5)  $GEN_{max} = 200$

In LM algorithm, the maximum number of epochs is 2000, initial  $\mu = 0.001$ ,  $\mu$  decrease and increase factors are 0.1 and 10 respectively, maximum  $\mu = 1e10$ .

### 4.4 Termination Criteria

- (1) Maximum number of function evaluations=6000 or
- (2) Target error  $\xi_t = 0.005$ .

### 4.5 PC Configuration

- (1) System (OS): Windows 2007 64 bit
- (2) CPU: Intel(R) Core(TM) i7-4770 3.40 GHz
- (3) RAM: 8 GB
- (4) Software: Matlab 2008b

## 5. RESULTS & DISCUSSION

In this work, an MLP is trained using Fireworks Algorithm. Fireworks algorithm is used to search the network's weight by minimizing the error function of MLP. The proposed method is applied in classification of heart, breast cancer, diabetes, lung cancer and kidney disease data sets. The experiment is carried out with 10-fold cross-validation of each data set. The performance is measured using Confusion Matrix [14]. The accuracy, sensitivity or true positive rate (TPR), specificity or true negative rate (TNR) and geometric-mean (GM) are calculated using confusion matrix. The geometric-mean is used to measure the trade-off between sensitivity and specificity [5, 16] and it is an important measure for class imbalance problem. The higher value of geometric-mean indicates better trade-off between sensitivity and specificity. The geometric-mean is calculated as following:

$$GM = \sqrt{\text{sensitivity} \times \text{specificity}}$$

The mean and standard deviation of training accuracies are given in Table 2. The mean and standard deviation of testing accuracies are given in Table 3. The mean of sensitivity, specificity and geometric-mean for testing are given in Table 4. The computational cost is given in Table 5 in terms of mean CPU times for training over 10-folds of each data set. From the Table 2, it is observed that LM algorithm trains the MLP better than PSO-W and IFWABS algorithm for all data sets. But, it is also observed from the Table 2 that IFWABS algorithm performs better than PSO-W and LM algorithms in testing of heart, breast cancer, diabetes and lung cancer data sets. PSO-W performs better than other two algorithms in testing of kidney data set. Though LM algorithm trains MLP better than other two algorithms (as per training accuracy), but it overfits the MLP neural network to the training data results in poor classification of test data. From the Table 4, it is also observed that IFWABS algorithm produces higher GM values than PSO-W and LM algorithm for heart and breast cancer data. IFWABS algorithm produces higher GM values than LM algorithm for diabetes

and kidney data. LM algorithm produces higher GM values than PSO-W and IFWABS algorithm for lung data only. Overall, it can be said that IFWABS algorithm provides better trade-off between sensitivity and specificity in classification. The MLP training with IFWABS algorithm takes less computational time than PSO-W for all data sets. The LM algorithm is faster than IFWABS algorithm in MLP training for heart, diabetes and kidney data sets. The above experimental results demonstrate that MLP training with IFWABS algorithm performs better in data classification. A good trade-off between sensitivity and specificity in classification is obtained using IFWABS algorithm.

## 6. CONCLUSION AND FUTURE WORKS

Artificial Neural Network training using Fireworks Algorithm is proposed in this paper. The proposed method is applied in classification of five well-known medical data sets. A comparative study is also made with PSO-W and LM algorithm. The proposed method performs better in classification task and also maintains a good trade-off between sensitivity and specificity. The proposed method is also computationally cost effective. Therefore, the proposed method can be an useful tool for classification in medical data mining. This work can be further extended by making a comparative study of other variants of Fireworks Algorithm in MLP training for medical data classification.

## 7. REFERENCES

- [1] Amato F., López A., Peña-Méndez E. M., Vařhara P., Hampl A. and Havel J. 2013. Artificial neural networks in medical diagnosis, *J. Appl. Biomed.*, Vol. 11, 47–58, DOI:10.2478/v10136-012-0031-x
- [2] Kennedy J. and Eberhart R. C. 1995. Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks*, Piscataway, NJ, 1942–1948.
- [3] Shi Y. and Eberhart R. C. 1998. A modified particle swarm optimizer, In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 1998)*, Piscataway, NJ, 69–73.
- [4] Haykin S. 2011. *Neural Networks And Learning Machines*, PHI, 3rd Eds.
- [5] Si T. and Sujauddin Sk. 2016. A Comparison of Grammatical Bee Colony and Neural Networks in Medical Data Mining, *International Journal of Computer Applications*, 134(6), 1–4.
- [6] Si T., Hazra S. , and Jana N. D. 2012. Artificial Neural Network Training Using Differential Evolutionary Algorithm for Classification, in S.C. Satapathy et al. (Eds.): *Proceedings of the InConINDIA 2012*, AISC 132, 769–778.
- [7] Shah H. and Ghazali R. 2011. Prediction of Earthquake Magnitude by an Improved ABC-MLP, In *Proceedings of the Developments in E-systems Engineering (DeSE)*, 2011, Dubai, IEEE, 312–317, DOI:10.1109/DeSE.2011.37
- [8] Si T. and Ghose R. 2015. Explosion sparks generation using adaptive transfer function in firework algorithm, In *Proceedings of the IEEE Third International Conference on Signal Processing, Communication and Networking (ICSCN)*, 1–9.
- [9] Yao X. and Liu Y. 1997. A new evolutionary system for evolving artificial neural networks, *IEEE Transactions on Neural Networks*, Vol. 8, Issue 3, 694–713.
- [10] Yao X. 1999. Evolving artificial neural networks, *Proceedings of The IEEE*, Vol. 87 , Issue 9, 1423–1447.
- [11] Lee Y. S., Shamsuddin S. M. and Hamed H. N. 2008. Bounded PSO Vmax Function in Neural Network Learning, In *Proceedings of the IEEE Eighth International Conference on Intelligent Systems Design and Applications*, 474–479.
- [12] Tan Y. and Zhu Y. 2010. Firework Algorithm for Optimization, In: Y. Tan et al. (Eds): *ICSI 2010, Part I, LNCS 6145*, 355–364, Springer-Verlag Berlin Heidelberg.
- [13] Liu J., Zheng S. and Tan Y. 2013. The improvement on controlling exploration and exploitation of Firework Algorithm, In: Y. Tan et al.(Eds): *ICSI 2013, Part I, LNCS 7928*, 11–23, Springer-Verlag Berlin Heidelberg.
- [14] Han L., Kamber M. and Pei J. 2011. *Data Mining: Concepts and Techniques*, The Morgan Kaufmann Series in Data Management Systems Morgan Kaufmann Publishers.
- [15] <http://cml.ics.uci.edu>.
- [16] Barandela R., Sanchez J.S., Garcá V. and Rangel E. 2003. Strategies for learning in class imbalance problems, *Pattern Recognition*, 36, 849–851.

Table 2. Mean and standard deviation of training accuracy (%).

Method	Heart	WDBC	Pima	Lung	Kidney
LM	100.00 ± 0.00	100.00 ± 0.00	92.20 ± 0.33	100.00 ± 0.00	100.00 ± 0.00
PSO-W	87.26 ± 0.61	96.14 ± 20.20	78.61 ± 0.31	95.57 ± 3.78	100.00 ± 0.00
IFWABS	86.87 ± 0.16	95.01 ± 0.20	78.12 ± 0.13	99.21 ± 0.47	99.56 ± 0.09

Table 3. Mean and standard deviation of testing accuracy (%).

Method	Heart	WDBC	Pima	Lung	Kidney
LM	67.44 ± 1.74	93.90 ± 0.97	72.08 ± 2.24	48.53 ± 10.36	97.88 ± 0.77
PSO-W	80.67 ± 2.35	95.45 ± 2.32	77.57 ± 2.47	58.83 ± 3.62	99.27 ± 0.69
IFWABS	84.75 ± 1.53	95.53 ± 1.45	77.14 ± 0.53	66.71 ± 4.72	98.79 ± 0.48

Table 4. Mean TPR, mean TNR and mean GM (%).

Method	Heart			WDBC			Pima			Lung			Kidney		
	TPR	TNR	GM	TPR	TNR	GM	TPR	TNR	GM	TPR	TNR	GM	TPR	TNR	GM
LM	60.57	61.38	60.94	94.76	81.34	87.78	50.57	81.27	64.04	78.11	74.67	75.95	96.65	100.00	98.31
PSO-W	58.64	87.74	71.70	98.20	90.81	94.36	98.20	90.81	94.36	93.46	12.33	18.78	98.83	100.00	99.41
IFWABS	82.27	86.80	84.50	97.78	91.75	94.72	57.42	87.68	70.94	92.78	58.33	73.02	98.06	100.00	99.02

Table 5. Mean CPU time (in second) for training.

Method	Heart	WDBC	Pima	Lung	Kidney
LM	6.80	452.02	54.02	88.04	13.41
PSO-W	214.19	501.58	506.43	389.29	343.77
IFWABS	209.37	407.40	505.00	42.59	280.70