

# Improved Nelder-Mead Optimization Method in Learning Phase of Artificial Neural Networks

Hasan Erdinc KOCER<sup>1</sup>, Mustafa Merdan\*<sup>2</sup>, Mohammed Hussein IBRAHIM<sup>3</sup>

Accepted : 14/12/2018

Published: 31/12/2018

**Abstract:** Artificial neural networks method is the most important/preferred classification algorithm in machine learning area. The weights on the nets in artificial neural directly affect the classification accuracy of artificial neural networks. Therefore, finding optimum values of these weights is a difficult optimization problem. In this study, the Nelder-Mead optimization method has been improved and used for training of artificial neural networks. The optimum weights of artificial neural networks are determined in the training stage. The performance of the proposed improved Nelder-Mead-Artificial neural networks classification algorithm has been tested on the most common datasets from the UCI machine learning repository. The classification results obtained from the proposed improved Nelder-Mead-Artificial neural networks classification algorithm are compared with the results of the standard Nelder-Mead-Artificial neural networks classification algorithm. As a result of this comparison, the proposed improved Nelder-Mead-Artificial neural networks classification algorithm has given best results in all datasets.

**Keywords:** Artificial neural networks, Nelder-Mead optimization method, Training algorithm.

## 1. Introduction

Artificial neural networks (ANNs) is a vital part of artificial intelligence. Machine learning and cognitive sciences depend on ANNs to solve various nonlinear mappings relationships [1]. ANN's are normally used a back-propagation algorithm to fix different problems on account of their approximation features. With respecting to the weights back-propagation algorithm computes explicit gradients of the error such as the mean of square error. However, ANNs trained with gradient descent based learning algorithm generally fall of slow convergence and local minima. To get rid of these issues metaheuristic algorithm that uses global search to find the solution has been used to train ANNs. For instance, the genetic algorithm utilized to training ANNs [2]; another method employed in order to train artificial neural network is reported by Slowik [3]. He used an adaptive differential evolution algorithm to train ANNs. The aim of the calculation that is achieved is to find optimal weights regarding an error rate [3]. Salman Mohaghegi et al. analyze the performance of particle swarm optimization (PSO) and they compared its convergence speed and robustness of the algorithm with the back-propagation algorithm. The learning algorithms have been utilized in order to modify the output synaptic weight array, and in each algorithm, the centers and widths in the hidden layer are inferred utilizing an offline clustering approach. According to the results, the PSO algorithm showed good performance even with a few numbers of particles. The statistical result proved PSO as a robust algorithm for learning ANNs [4]. David J. Montana and Lawrence Davis utilized an evolutionary algorithm for enhancing the performance of feed word artificial neural networks. In addition, the algorithm added more domain-specific knowledge into ANNs [5]. Malinak

and Jaksa utilized evolutionary algorithm with a local search algorithm to get the best performance [6]. Kenndy and Eberhat used swarm intelligence PSO for the first time [7]. Carvalho and Ludermir applied PSO to medical benchmark problems. They compared the statistical result with a local search operator in order to learn ANNs. The result showed a better performance than other well-studied algorithm [8]. Apart from PSO, the scientists applied other swarm intelligence such as ant colony algorithm. Blum and Socha employed a new version of ACO algorithm for training artificial neural network [9]. Liang Hu, Longhui Qin, Kai Mao, Wenyu Chen, and Xin Fu used a genetic algorithm for training ANNs to solve the real-life problem of Multipath Ultrasonic Gas Flowmeter. They used GANNs for multipath ultrasonic flowmeter (UFM) to help decrease its error rate when detecting the flowrate of the complicated flow field. The evaluation of results showed by ANNs and GANNs demonstrated better achievement for ANNs trained by genetic algorithm compared with once trained with gradient descent based algorithm [10]. In addition to genetic algorithm, to find the optimal weight for a neural network, the genetic algorithm has been utilized to enhance the whole structure for artificial neural networks. For example, K. G. Kapanova, Dimov and M. Sellier utilized GA for optimizing the architecture of ANNs. This approach allowed rearranging hidden layers, neurons in every hidden layer, the connections between neurons and the activation function [11].

In this study, we improved the Nelder-Mead optimization method and used to determine the optimum weight values of ANNs. In the proposed improved Nelder-Mead-ANNs classification algorithm, ANN classification algorithm performs well in the learning stage. The paper is organized as follows: The ANN classification algorithm is explained in the second section. The standard Nelder-Mead optimization method is explained in the third section. The improved Nelder-Mead optimization method is explained in the fourth section and the experimental results are discussed in the fifth section. Finally, the conclusion is explained in the last section.

<sup>1</sup> Electronics and Computer Edu., Selcuk University, Konya, Turkey

<sup>2</sup> Stud. In computer Engg., Selcuk University, Konya, Turkey

<sup>3</sup> Computer Engg., Necmettin Erbakan University, Konya, Turkey

\* Corresponding Author: Email: mustafaaahmet00@yahoo.com

## 2. Artificial Neural Networks

ANNs consist of neurons and these neurons that connect together. These neurons can bind to each other very complexly as in the real neurons system. Each neuron has a different weight input and one output. For this purpose, the sum of the entries with different weights is expressed by the following Equation 1 [12].

$$net = \sum_{i=1}^m I_i W_{ij} + b \quad (1)$$

Where,  $m$  is the number of neurons in the layer,  $w$  is the weight between  $i$  and  $j$  neuron and  $b$  is a bias neuron. The neuron output value is obtained after the output of net function passes through the activation function. Usually, in ANNs systems, the sigmoid activation function is used. The sigmoid activation function is shown in Equation 2 [13].

$$F(net) = \frac{1}{1 + e^{-net}} \quad (2)$$

The error rate between the output from the ANNs system and the actual output is calculated. The mean square error is given in Equation 3 below.

$$MSE = \frac{1}{n} \sum_{x=1}^n (O_x - T_x)^2 \quad (3)$$

Where,  $n$  represents a number of instances in datasets,  $O_x$  is the output generated from the  $x^{th}$  inputs and  $T_x$  is the target output of the  $x^{th}$  inputs. After training data is completed, the trained network can estimate the result of any given dataset according to the last state of the weight values. This process is called network training. There are different training algorithms in network learning. In this study, the Nelder-Mead optimization method is used in the network training process.

## 3. Nelder-Mead optimization method

Nelder-Mead is a simple optimization method developed by Nelder&Mead [17]. It is also called as Amoeba method in many kinds of literature. It is a method widely used in multi-dimensional unconstrained optimization problems. It is very simple to understand the basic Nelder-Mead method and its usage is very easy. For this reason, it is a very popular method especially in many fields of chemistry and medicine and monolith. It is widely used for solving parameter estimation and statistical problems. It can also be used in the field of experimental mathematics [14].

The Nelder-mead optimization method is a simple method used to find the local minimum of several variable functions generated by Nelder and Mead. Simplex is an  $n$ -dimensional geometric shape. It includes  $(N + 1)$  points in  $N$  dimensions. The simplex in two dimensions consists of a triangle. It becomes a triangular prism in three dimensions (4 surfaces). It is an example of a research method that compares the function values at three corner points of a triangle. By analyzing the function at all given points, the program moves the highest and lowest values to the reflection point relative to the surface above the highest and lowest elements. In two dimensions, this method is a model search method that compares the function values at the corners of a triangle. When we consider the minimization state of a bivariate  $z=f(x, y)$  function, the value of the  $z$  function at the worst corner of the triangle (w

worst vertex) is greatest.

For this reason, this worst corner is replaced by a new point. So we get a new triangle. The search is now resumed in this new triangle, so that the process becomes a triangle array where the function value becomes smaller and smaller at the corner point value [15]. The steps of the Nelder-Mead optimization method are given below.

*Step 1: Initialize with random solution*

*Step 2: Order solutions according to*

*the fitness values  $(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1})$*

*Step 3: Calculate  $x_0$  the centroid of all points except  $x_{(n+1)}$*

*Step 4: Compute reflection point  $x_r = x_0 + \alpha(x_0 - x_n)$*

*Step 5: If  $(x_0) \leq f(x_r) < f(x_n)$  then accept  $x_r$  and terminate the iteration*

*Step 6: Expansion, If  $(x_r) < f(x_1)$  then compute the expanded point  $x_e = x_r + \gamma(x_r - x_0)$ . If  $(x_e) < f(x_r)$  accept  $x_e$  and terminate the iteration,*

*otherwise accept  $x_r$  and terminate the iteration.*

*Step 7: Contraction, If  $(x_r) \geq f(x_n)$  perform*

*a contraction between  $x_0$  and  $x_n$ .  $x_c = x_0 + \rho(x_0 - x_n)$ .*

*If  $(x_c) < f(x_n)$  accept  $x_c$  and terminate iteration.*

*Step 8: Shrink, Replace all points except the best  $x_1$  with  $x_i = x_0 + \sigma(x_i - x_1)$  and go to step 2.*

## 4. Improved Nelder-Mead optimization method

Since the Nelder-Mead optimization method is simple, it is also known as a simple optimization method. In this study, the Nelder-Mead optimization method is combined with PSO algorithm to find optimum value of ANNs weights. In the proposed classification algorithm, firstly, the weights of the ANNs classification algorithm are generated randomly and the MSE is calculated, then the best result is selected. The weights are updated according to the best solution obtained from random solutions according to the Nelder-Mead optimization method. Then the best MSE is selected. In the next iteration, if the difference error of the previous iteration and of the current error is less than 0.5 update the weights according to the velocity equation of the PSO algorithm is given in equations 4 and 5 below.

$$v_i^{t+1} = w * v_i^t + c_1 r_1 (Pbest_i - x_i^t) + c_2 r_2 (Gbest - x_i^t) \quad (4)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (5)$$

Where,  $t$  is the number of iteration,  $w$  is inertia weight,  $d$  is dimension,  $x_{id}$  current position of particle,  $v^t$  the previous velocity of particle,  $Pbest_i$  is a better fitness value,  $Gbest$  is a better fitness value of all particles,  $c_1$  and  $c_2$  are two positive constants represents accelerate factor and  $r_1$  and  $r_2$  are two random functions in the range [0,1]. Otherwise, the weight updating according to the Nelder-Mead optimization method functions. Optimization algorithms are used in many engineering areas and aim to find the best end result. In this study, the improved Nelder-Mead optimization method is used to find the optimum weights in the ANNs classification model. The flowchart of the proposed ANNs classification model is given in Fig. 1 below.

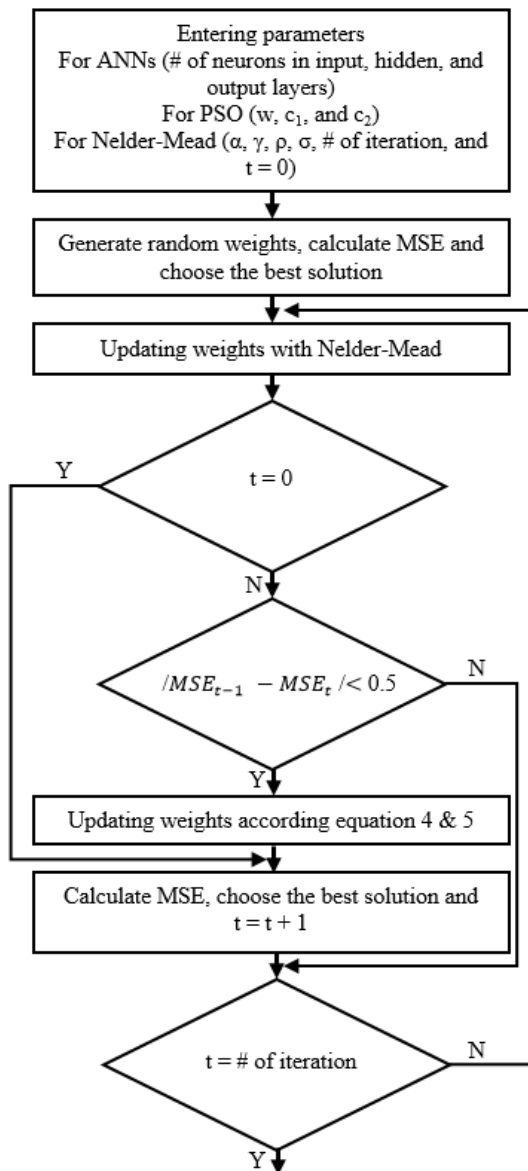


Fig. 1. Flowchart of the proposed ANNs model

## 5. The experimental results

In the experimental results section, the performance of the two methods is compared by applying the developed Nelder-Mead method and the standard Nelder-Mead method in the training of the ANNs classification algorithm. The codes of standard Nelder-Mead and developed Nelder-Mead methods were written in the visual studio. The performance of the classification algorithms was applied to nine UCI datasets [16]. The characteristic of datasets is given in Table 1.

Table 1. The characteristic of datasets

Datasets name	No. of Attributes	No. of sample	No. of classes
Balance	4	625	3
Shuttle-Landing	6	253	2
lymphography	18	148	4
Breast cancer	9	699	2
Iris	4	150	3
Wine	13	178	3
Thyroid	5	215	3
E.coli	7	336	8
Glass	9	214	6

An appropriate ANNs was created for each dataset and the number of nodes in the hidden layer was set. Bias node was used in the hidden and output layers and sigmoid was used as the activation function. In the Nelder-Mead optimization method, the values  $\alpha$ ,  $\gamma$ ,  $\rho$ ,  $\alpha$ , and  $\sigma$  were set to 2, 0.5, 1, and 0.5, respectively. Initial values of weights were between 10 and -10 and the algorithm was run at 1000 iterations. MSE was used as an activation function in both the Nelder-Mead optimization method. The classification performance of the algorithm is calculated based on the final value of this fitness function. 10-Fold cross-validation was used for classification accuracy (CA).

$$CA = \frac{\text{No. of correct predicted samples}}{\text{No. of total samples}} \quad (6)$$

The classification results of the proposed improved Nelder-Mead-ANNs classification algorithm and the standard Nelder-Mead-ANNs classification algorithm is given in table 2 below. In this table, the CA and MSE obtained in the training and testing stages of both classification algorithms are given. Also, H is the optimal number of neurons in a hidden layer.

When we examine the experimental results of Table 2, the proposed Nelder-Mead-ANNs classification algorithm showed better classification accuracy and MSE in alldatasets than the standard Nelder-Mead-ANNs classification algorithm.

## 6. Conclusion

In our study, we performed the training of the ANNs classification algorithm with standard Nelder-Mead and improved Nelder-Mead optimization methods. The improved Nelder-Mead optimization method was developed because the standard Nelder-Mead optimization method did not fulfill this task properly. With the help of the PSO optimization method, the developed Nelder-mead optimization method gave better results in the ANNs training. The improved Nelder-Mead-ANNs classification algorithm was applied to 9 datasets of UCI machine learning repository and performed better on all datasets. So the developed classification

Table 2. The classification results of the Standard Nelder-Mead-ANNs and Improved Nelder-Mead-ANNsclassification algorithms

Datasets name	Standard Nelder-Mead-ANNs				Improved Nelder-Mead-ANNs			
	Training	Testing	MSE	H	Training	Testing	MSE	H
Balance	0,789	0,792	0,330	9	<b>0,908</b>	<b>0,902</b>	<b>0,138</b>	3
Shuttle-Landing	0,889	0,892	0,181	7	<b>0,990</b>	<b>0,955</b>	<b>0,015</b>	7
lymphography	0,615	0,648	0,515	12	<b>0,902</b>	<b>0,710</b>	<b>0,169</b>	22
Breast cancer	0,970	0,970	0,055	10	<b>0,983</b>	<b>0,990</b>	<b>0,031</b>	5
Iris	0,845	0,892	0,291	9	<b>0,988</b>	<b>0,953</b>	<b>0,024</b>	8
Wine	0,992	0,928	0,42	12	<b>0,984</b>	<b>0,931</b>	<b>0,030</b>	17
Thyroid	0,816	0,802	0,325	3	<b>0,983</b>	<b>0,956</b>	<b>0,029</b>	10
E.coli	0,592	0,577	0,602	4	<b>0,840</b>	<b>0,810</b>	<b>0,276</b>	10
Glass	0,375	0,386	0,748	4	<b>0,682</b>	<b>0,619</b>	<b>0,507</b>	11

algorithm can be applied to many fields such as medical, engineering and recognition systems.

## Acknowledgement

This study presents the results of Master thesis of Mustafa Merdan in Selcuk University Natural and Applied Science Institute Computer Engineering Programme.

## References

- [1] APA Kecman, V. (2001). Learning and soft computing: support vector machines, neural networks, and fuzzy logic models. MIT press.
- [2] Alhamdoosh, M., & Wang, D. (2014). Fast decorrelated neural network ensembles with random weights. *Information Sciences*, 264, 104-117.
- [3] Slowik, A. (2011). Application of an adaptive differential evolution algorithm with multiple trial vectors to artificial neural network training. *IEEE Transactions on Industrial Electronics*, 58(8), 3160-3167.
- [4] Mohaghegi, S., del Valle, Y., Venayagamoorthy, G. K., & Harley, R. G. (2005, June). A comparison of PSO and backpropagation for training RBF neural networks for identification of a power system with STATCOM. In *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE* (pp. 381-384).
- [5] Montana, D. J., & Davis, L. (1989, August). Training Feedforward Neural Networks Using Genetic Algorithms. In *IJCAI (Vol. 89, pp. 762-767)*.
- [6] Malinak, P., & Jaksa, R. (2007, September). Simultaneous gradient and evolutionary neural network weights adaptation methods. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on* (pp. 2665-2671). IEEE.
- [7] Kennedy, J. (2011). Particle swarm optimization. In *Encyclopedia of machine learning* (pp. 760-766). Springer US.
- [8] Carvalho, M., & Ludermit, T. B. (2007, September). Particle swarm optimization of neural network architectures and weights. In *Hybrid Intelligent Systems, 2007. HIS 2007. 7th International Conference on* (pp. 336-339). IEEE.
- [9] Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, 1(4), 28-39.
- [10] Hu, Liang, et al. "Optimization of neural network by genetic algorithm for flowrate determination in multipath ultrasonic gas flowmeter." *IEEE Sensors Journal* 16.5 (2016): 1158-1167.
- [11] Kapanova, K. G., Dimov, I., & Sellier, J. M. A genetic approach to automatic neural network architecture optimization. *Neural Computing and Applications*, 1-12.
- [12] Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.
- [13] Schalkoff, R. J. (1997). *Artificial neural networks (Vol. 1)*. New York: McGraw-Hill.
- [14] Kelley, C. T. (1999). Detection and remediation of stagnation in the nelder-mead algorithm using a sufficient decrease condition. *SIAM journal on optimization*, 10(1), 43-55.
- [15] Lagarias J. C., J. A. Reeds, M. H. Wright, and P. E. Wright, 1998. Convergence properties of the Nelder-Mead simplex method in low dimensions. *SIAM Journal on Optimization*, 9(1), p. 112- 147.
- [16] Asuncion, A., & Newman, D. (2007). UCI machine learning repository.
- [17] Nelder, John A., R. Mead (1965). A simplex method for function minimization. *Computer Journal*. 7: 308–313.