Journal of Computer Science 10 (1): 106-114, 2014
ISSN: 1549-3636

doi:10.3844/jcssp.2014.106.114 Published Online 10 (1) 2014 (http://www.thescipub.com/jcs.toc)

A COMPARATIVE ANALYSIS OF OPTIMIZATION TECHNIQUES FOR ARTIFICIAL NEURAL NETWORK IN BIO MEDICAL APPLICATIONS

[1]V. Saishanmuga Raja and [2]S.P. Rajagopalan

[1]Faculty of Computer Science and Engineering,
[2]Faculty of Computer Applications,
MGR Educational and Research Institute, Chennai, India

Received 2013-09-06; Revised 2013-09-22; Accepted 2013-11-12

ABSTRACT

In this study we compare the performance of three evolutionary algorithms such as Genetic Algorithm (GA) Particle Swarm Optimization (PSO) and Ant-Colony Optimization (ACO) which are used to optimize the Artificial Neural Network (ANN). Optimization of Neural Networks improves speed of recall and may also improve the efficiency of training. Here we have used the Ant colony optimization, Particle Swarm Optimization and Genetic Algorithm to optimize the artificial neural networks for applications in medical image processing (extraction and compression). The aim of developing such algorithms is to arrive at near-optimum solutions to large-scale optimization problems, for which traditional mathematical techniques may fail. This study compares the efficiency and results of the three evolutionary algorithms. We have compared these algorithms based on processing time, accuracy and time taken to train Neural Networks. The results show that the Genetic Algorithm outperformed the other two algorithms. This study helps researchers to get an idea of selecting an optimization algorithm for configuring a neural network.

Keywords: Genetic Algorithm, Ant-Colony Optimization, Particle Swarm Optimization, Neural Network and Image Segmentation

1. INTRODUCTION

Artificial neural networks are capable of performing many classification, learning and function approximation tasks, yet in practice sometimes they deliver only marginal performance. Inappropriate topology selection and weight training are frequently blamed. Increasing the number of hidden layer neurons helps improving network performance, yet many problems could be solved with very few neurons if only the network took its optimal configuration. Unfortunately, the inherent nonlinearity of ANN results in the existence of many sub-optimal networks and the great majority of training algorithms converge to these sub-optimal configurations. To address these problems we must use an optimal algorithm to optimize the Artificial Neural Network. Here we use three evolutionary algorithms to optimize the neural network and compare their performance.

Evolutionary algorithms are stochastic optimization methods which are population-based inspired by natural selection able to find several solutions in a single run, thus making them a good alternative to standard methods. There involves a large amount of difficulties in using mathematical optimization problems for engineering applications which contributed to have alternative solutions. Linear programming and dynamic programming techniques often fail in solving large problems with large number of variables and non linear optical solutions. To overcome these problems, researchers have proposed evolutionary-based algorithms for searching near-optimum solutions to problems.

Corresponding Author: V. Saishanmuga Raja, Faculty of Computer Science and Engineering, MGR Educational and Research Institute, Chennai, India

Artificial Neural Networks (ANNs) play an essential role in the medical imaging field, including medical image analysis and computer-aided diagnosis, because objects such as lesions and organs in a medical image may not be represented into an accurate equation easily. One of the main uses of Artificial Neural Network in Medical Image analysis is to classify lesions into some classes such as normal or abnormal, malignant or benign and lesions or non-lesions. Genetic Algorithm and Ant-colony algorithm which are population based search methods are inspired from nature, are effective in optimization with a large number of design variables and low cost function evaluation. In case of Genetic Algorithm its performance can be improved using various schemes such as fast full wave methods, micro-Genetic Algorithm and Parallel Genetic Algorithm using parallel computation. Ant colony optimization is inspired by the social behavior of ants. Ants find a shortest route to the food particles from their nest.

Particle swarm optimization algorithm was inspired by the social behavior of animals, such as bird flocking or fish schooling (Rossana *et al*., 2011). In PSO, each solution is a 'bird' in the flock and is referred to as a 'particle'. As a chromosome in Genetic Algorithms, a particle is in POS. Unlike Genetic Algorithms, in the process of evolution the PSO does not create new child from Parents, instead the particle in the population evolve to its social behavior and there by finds a path towards the destination (Jiang *et al*., 2007).

In this study, the three Evolution Algorithms are presented and are reviewed. Performance analysis is done among the three algorithms based on ease-of-use, accuracy and time taken to train the Neural Networks. We also present Guidelines for determining the appropriate parameters to be used with these algorithms.

In the section 2 we give a brief description about neural network and different variable selection process. Next we discuss about medical image segmentation. In section 4 we analyze the three evolutionary algorithms and in section 5 we present the experimental results of comparing these algorithms.

2. ARTIFICIAL NEURAL NETWORK

Artificial Neural Network is the most sought technology in the last two decades that is used in various engineering applications. The ANN is a mathematical model which inspired from the structure and functions of the neurons in the human brain. A Neural Network consists of number of neurons which are connected through weights. The ANN can learn about the environment (application or task) by adjusting the values of the weights. An ANN can be classified in to two sub categories such as Supervised Learning and Un-Supervised Learning. In supervised learning an ANN learns with a help of a "Teacher" or using an ideal output to achieve goal. In unsupervised learning an ANN does not require a teacher; instead it learns using the cost function. A desired goal in an artificial neural network is achieved by learning.

2.1. Feed-Forward Artificial Neural Networks

A neural network is called as a Feed-Forward neural network when the information flows in only direction from input to output without any loops. We take the feed forward neural network for the use in medical image segmentation. The most important factor that is to be considered in building an artificial neural network is the proper selection of the input variables.

2.2. Input Variable Selection

The performance of the Artificial Neural Network models vary based on the large variety of inputs such as un-informative inputs, or more inputs than that is required. To constitute an optimal set of input variables which may have an impact on the performance of the ANN, the following factors may be considered.

Relevance: In most cases a very few input variables are selected or the selected variables are un- in-formative. The output of the model may be very poor in this case since the input variables are not relevant to the expected output. It is advised that before selecting the input variables it is necessary to have a prior knowledge of the system and survey of the available data.

Computational effort: The number of input variables has an immediate effect in the size of the ANN which increases the computational complexity. The-se effects have a significant impact on the training speed of the neural network. When we use a Multi Layer Perceptron (MLP) ANN, the number of connection weights in the input layer increases.

Dimensionality: The number of samples required to map a given function with sufficient confidence in-creases when the dimensionality of a model increases linearly. The ANNs like MLPs fall into the curse of the dimensionality due to the increasing incoming weights as input variables. Dimensionality reduction is possible in ANN only by avoiding redundancy and irrelevant input variables.

Training difficulty: Training of an ANN becomes difficult due to the irrelevant and redundant input

variables. The effect of redundancy in input variables increases the error function. The irrelevant input variables add noise to the model which reduces the speed of learning process. More iteration may be required to determine the error function which in turn increases the computational burden. The working principle of an Artificial Neural Network is shown in the **Fig. 1**.

3. MEDICAL IMAGE SEGMENTATION USING ANN

3.1. Multi Layer Perceptron Neural Network

Multi Layer Perceptron Artificial Neural Network is used in various applications such as feature extraction, optimization, classification and compression (Hancock *et al*., 2010). The MLP Artificial Neural Network is suitable for medical image segmentation for the following reasons. The first reason is the output of the MLP ANN with a hidden layer is a non linear function with the combination of the outputs in the hidden layer. An objective function estimates the parameters of the network. The second reason is that the number of neurons in the hidden layer is lesser than the input layer. This means the smaller dimension in the hidden layer. The third reason is that the MLP ANN easily deals with the irrelevant input variables by adding zeroes to them.

Medical image segmentation is a process that involves in division of a given image into important regions with similar properties. Image segmentation is the process of identifying the boundaries of organs and tumors during clinical analysis. Image segmentation and edge detection are done after image registration. A. Dufour *et al*. (2013) pro-posed an automated method to segment the blood vessels from 3D Time of Flight (TOF) MRA volume data. The method consists of three steps: (1) Background removal, (2) volume quantization and (3) classification of primitives. First, the feed forward neural network is initialized and trained with back-propagation algorithm. The net-work is simulated after training. The features that are extracted from the medical images are assigned as input variables to the ANN.

All training is done using back propagation with adaptive learning rate and momentum with trainbpx function. During training, to set the number of epochs an optional parameter is used. Then the network is trained and simulated. The multi layer feed forward network is shown in **Fig. 2**. Wavelets are used for feature extraction. Then we compute the difference between the output and expected result. In the experiment the ANN is trained using 50 datsets obtained from MRA dataset. New MRA datasets are given as input to the trained network for testing. The segmentation performance is measured by the value accuracy as shown in the Equation (1):

$$\text{accuracy} = \frac{\text{Number of correctly classified primitives}}{\text{Total number of primitives}} \quad (1)$$

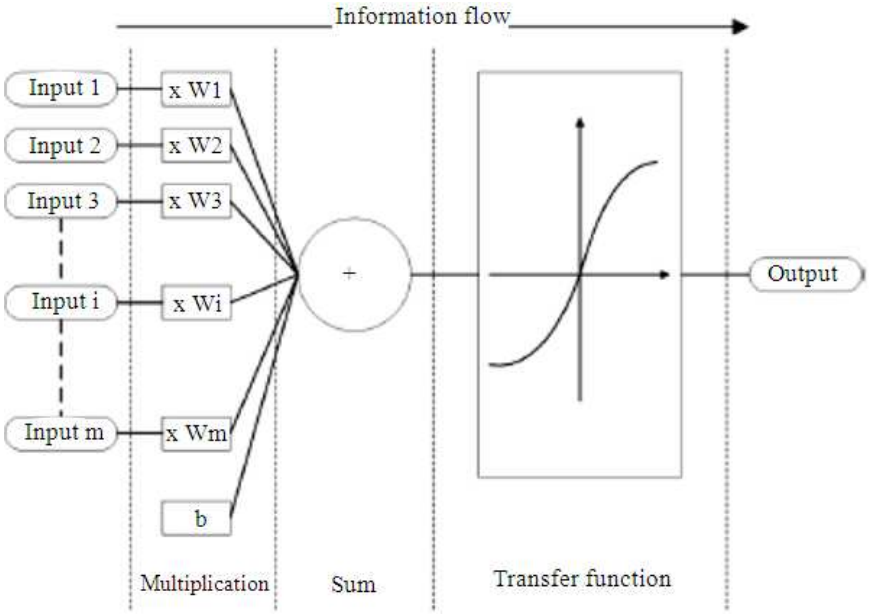

Fig. 1. Working principle of an artificial neural network

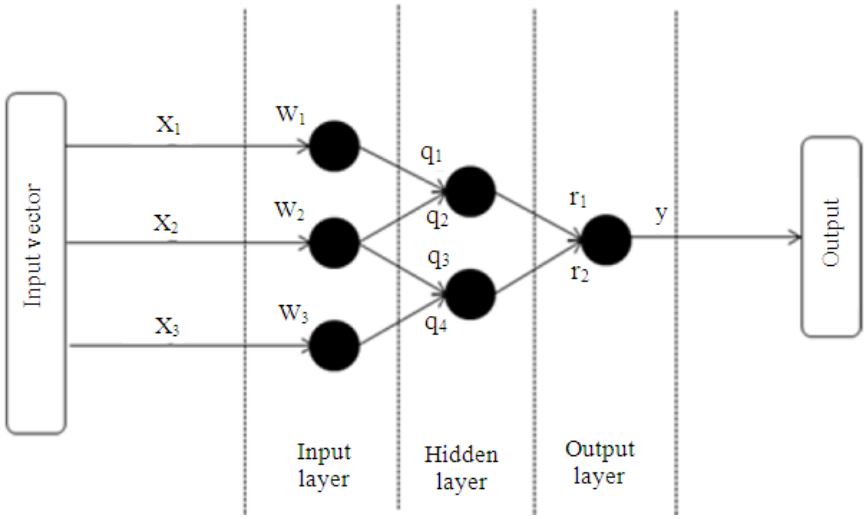

Fig. 2. Feed-Forward neural network

Ma *et al*. (2010) designed a two-layer Hopfield neural network called the Competitive Hopfield Edge-Finding Neural Network (CHEFNN) to detect the edges of CT and MRI images.

4. ANALYSIS OF THE THREE EVOLUTIONARY ALGORITHMS

The evolutionary algorithms in general have a common approach towards a given application. The given problem requires a representation for each method. A brief review is presented about the three algorithms in the sections 4.1,4.2 and 4.3.

4.1. Genetic Algorithms

Genetic algorithm is an evolutionary computing technique that can be used to solve problems with a vast solution space (Cao and Zhang, 2010). A solution to a given problem is represented in the form of a string, called 'chromosome', consisting of a set of elements, called 'genes', that hold a set of values for the optimization variables. As a preparation to start the optimization process, a Genetic Algorithm, requires a group of initial solutions as the first generation. The first generation is usually a group of randomly produced solutions created by a random number generator. The population, which is the number of individuals in a generation, should be big enough so that there could be a reasonable amount of genetic diversity in the population. Also, it should be small enough for each generation to be computed in a reasonable period of time using the computer resources available. Typically, a population includes individuals between 20 and 100. **Figure 3** shows the flowchart of a Genetic algorithm used for optimization.

The fitness function is evaluated to measure how close that the individuals fit the desired result. A fitness function could be either complex or simple depending on the optimization problem addressed. In a case of minimization problem, the most fitted individuals will have the lowest numerical value of the associated fitness function.

Individuals are selected according to a fitness-based process. The operator of selection is made up of ranking and selection progress, by which more copies of the individuals that fit the optimization problem better will be produced in the next generation. In GAs, there are mainly two ways to select a new population: Roulette Wheel Selection (RWS) and Stochastic Universal Sampling (SUS). The individuals will be recombined (crossover) after the selection. This operation is to produce two new individuals from two existing individuals selected by the operator of selection by cutting them at one or more position and exchanging the parts following the cut. The new individuals therefore can inherit some parts of both parents' genetic material. There are usually four ways of doing this: One point crossover, two-point crossover, cycle crossover and uniform crossover (Saishanmuga and Rajagopalan, 2012). **Figure 4(a)** shows an example of the two-point crossover progress. Mutation is another operator to produce new individuals. The difference is that the new individual is produced from a single old one.

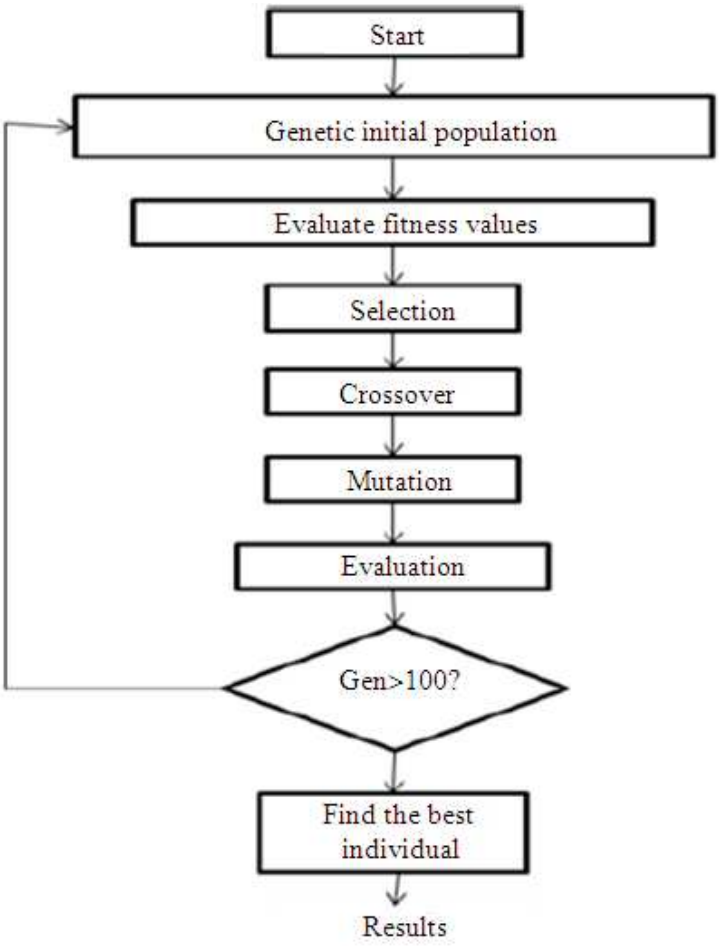

Fig. 3. Flowchart of a simple genetic algorithm

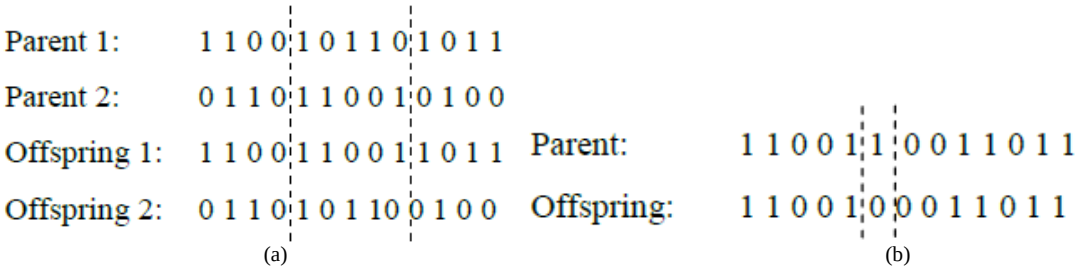

Fig. 4. (a) Crossover operation (b) Mutation operation

In this operation, the bit values of each individual are randomly re-versed according to a specified property. A mutation can also helps the GA to avoid local optimums and find the global best solution. **Figure 4(b)** represents how the mutation operator works.

4.2. Particle Swarm Optimization

PSO was developed by (Hansen *et al*., 2008). PSO is inspired by the group of birds flying together to an unknown destination. In PSO, each solution is a 'bird' in

the group and is referred to as a 'particle'. As a chromosome in Genetic Algorithms, a particle is in POS.

PSO actually imitates a group of birds that communicate with each other when flying together to an unknown destination. Initially each bird flies in a specific direction, but changes its direction when communicates with the other birds. All other birds will follow a particular bird which they think has found out the best direction to the destination. At this point all the birds fly towards that particular bird by changing their current velocity. Each bird then explores its new local position (Local Search). This process of choosing one bird in the group which is well acquainted with the current location is continued till the birds reach the desired destination. It has to be noted that the birds learn from their own intelligence and from the experience of the other birds (Global Search).

The process is started with an 'N' number of random particles. The position of the i_{th} particle is represented by a point in 'S' Dimensional space where S is number of variables. Throughout the process 'i' monitors tree values: The current position (X_i), the best position it reached in previous cycle (P_i); and the velocity (V_i). In each cycle, the position of each particle is calculated as the best fitness of all particles. Accordingly each particle updates its current velocity V_i to join the best particle (Dehuri and Cho, 2010):

$$\begin{aligned}\text{New } V_i &= \omega X \text{current } V_i + c_1 \text{ x rand}() X(P_i - X_i) \\ &+ c_2 \text{ x rand}() X(P_g - X_i)\end{aligned} \quad (2)$$

The first part of the Equation (2) represents the current position of the particle. The second part of the equation represents the new location of the particle and the third part of the equation represents the communication of the particles to compare its local position with the best particle.

4.3. Ant Colony Optimization

ACO was developed by (Geetha and Srikanth, 2012) based on the fact that ants are able to find the shortest route between their nest and a source of food. Ants use pheromone trails to communicate with each other. An ant roaming in various directions leave this pheromone on the ground making a path it followed by this trail. An isolated ant when encounters the previously laid trail decides to follow the trail with a high probability of finding a food particle. When it follows the previously laid trail it enforces its trail over it making the trail more intensive. The ant which found a food particle will return to its nest with a shortest route laying the pheromone trail. The remaining ants will follow this shortest route to the food and also they leave their pheromone tail. Ants therefore can find optimal solutions using th e local state knowledge and about the effects of actions that can be performed in the local state.

ACO can be implemented by representing a variable S for each ant and variable i to store n_i options with their values l_{ij}. Their pheromone concentration can be represented by T_{ij}. So an ant consists of S variables that will describe the path chosen by the ant. The process can be started by making m random ants. As shown in the Equation (3), Pheromone concentration associated with each possible route (variable value) is changed in a way to reinforce good solutions, as follows (Dehuri and Cho, 2010):

$$T_{ij}(t) = pT_{ij}(t-1) + \Delta T_{ij}; t = 1,2,3...,T \quad (3)$$

where, T is the number of iterations; $t_{ij}(t)$ is the revised concentration with option l_{ij} at iteration t, $t_{ij}(t\text{-}1)$ is the concentration of pheromone at the previous iteration (t-1); Δ_{tij} = change in pheromone concentration; and r = pheromone evaporation rate (0-1). The reason for allowing pheromone evaporation is to avoid too strong influence of the old pheromone to avoid premature solution stagnation (Shen *et al*., 2011).

5. EXPERIMENTAL RESULTS

The performance of the three algorithms were measured using the following criteria; (1) the percentage of success (the number of trials required for the function to reach the target value); (2) The average value of the solution obtained in all the trails; (3) The time taken by the network to learn. Twenty trail run was made for each algorithm. Two well known functions F8 and F10 are used to test the optimization algorithms. F8 function (Griewank's function) is a scalable, non linear and non seperable function which takes any number of variables (X_iS) (Ibric *et al*., 2012).

The F8 function scales to any number of variables N. The values of each variable can be put in the range of (-512 to 511). The global optimum (minimum) solution for this function is known to be zero when all N variables equal zero. F10 function is non linear and non separable which uses two variables x and y as show in Equation (4):

$$f(x_{i|i=1,N}) = 1 + \sum_{i=1}^{N} \frac{x_i^2}{4000} - \prod_{i=1}^{N} (\cos(x_i / \sqrt{i})) \quad (4)$$

The **Table 1** clearly shows that the PSO algorithm outperforms in all the criteria when compared to other algorithms. GA's performance was poor in terms of the success rate to find a target value. But GA has performed well in terms of training the network in minimum time compared to other two algorithms. ACO has not performed well in any of the test.

Table 2 compares the training and testing time of neural network optimized by the three algorithms. The results show the training performed with neural network optimized by Genetic algorithm, PSO and ACO with 15, 30, 60 and 120 samples, 10 runs. Showing the average in each generation and standard deviation for each generation run, better error found by genetic algorithm, best training method and execution time. The PSO better testing time compared to ACO. The GA has outperformed the remaining two algorithms in both the testing time and training time of the neural network.

The **Fig. 5** clearly shows the performance comparison of the three algorithms based on the time taken to train the neural network. GA takes minimum time to train the ANN and PSO takes some more times when compared to GA. ACO takes the maximum time to train the network. The Mean square error is considered while evaluating the training time.

The performance evaluation of the three algorithms based on their accuracy in image segmentation is shown in the **Table 3**. The result shows that the accuracy in image segmentation is higher when the neural network is optimized with Genetic algorithm. It is evident that GA and PSO are very closer in their results. The ACO is poor in its performance when compared to GA and PSO.

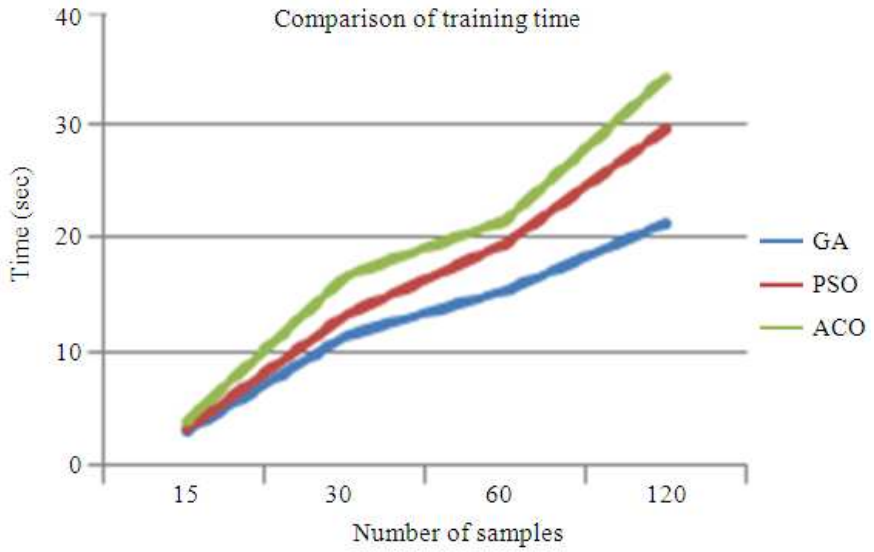

Fig. 5. Comparison of GA, PSO and ACO on training time

Table 1. Results of the optimization problem

		Number of variables						
		F8				EF10		
Comparison	Algorithms	10	20	50	100	10	20	50
% success	GA	70.000	50.000	30.000	0.000	30.000	0.000	0.00
	PSO	80.000	90.000	100.000	100.000	100.000	80.000	60.00
	ACO	60.000	50.000	30.000	0.000	80.000	60.000	50.00
Mean	GA	0.050	0.089	0.187	0.550	0.487	1.210	5.82
Solution	PSO	0.098	0.087	0.014	0.011	0.012	0.075	2.84
	ACO	0.014	0.016	0.011	0.009	0.014	0.068	0.564

Table 2. Results of training and testing

		Number of samples			
Comparison	Algorithms	15	30	60	120
Testing	GA	0.08	1.02	2.11	3.18
Time in secs	PSO	0.80	1.85	2.78	4.05
	ACO	1.20	2.12	2.99	4.15
Training	GA	2.85	11.45	15.28	21.28
Time in secs	PSO	3.25	13.25	19.65	29.65
	ACO	3.75	16.68	21.75	34.25

Table 3. Results of accuracy and average time

		No. of Samples			
Comparison	Algorithms	15	30	60	120
Accuracy	GA	100.00	98.00	96.00	92.00
in %	PSO	100.00	98.60	95.00	89.00
	ACO	97.00	94.00	90.00	86.00
Average time	GA	10.80	13.45	15.28	21.28
in seconds	PSO	11.25	15.25	19.65	24.45
	ACO	18.75	21.68	26.57	32.85

6. CONCLUSION

In the current work, we have reviewed the optimization algorithms for neural networks based on their accuracy, training time and testing time. We found that amongst the three optimization algorithms used, GA has performed well in all the evaluations. It is also evident that the Genetic algorithm is most suitable for training the neural network with minimum time and minimum mean square error. We recommend Genetic algorithm as most suitable algorithm for optimization of neural network. The limitation observed while evaluating the algorithms was that the Neural Network started mugging up the instead of learning when huge data sets were given as inputs. Future works can be addressed to compare other classifiers and others evolutionary algorithms. Others comparison criteria can be used such the needed speed and the robustness of the algorithm. A wrapper approach can be included in the proposed process in order to avoid irrelevant features over the optimization process.

7. REFERENCES

Cao, Z.F. and Z.H. Zhang, 2010. Parameter settings of genetic algorithm based on multi-factor analysis of variance. Proceedings of the IEEE 4th International Conference on Genetic and Evolutionary Computing, Dec. 13-15, IEEE Xplore Press, Shenzhen, pp: 305-307. DOI: 10.1109/ICGEC.2010.82

Dehuri, S. and S.B. Cho, 2010. Evolutionarily optimized features in functional link neural network for classification. Expert Syst. Applic., 37: 4379-4391. DOI: 10.1016/j.eswa.2009.11.090

Dufour, A., O. Tankyevych, B. Naegel, H. Talbot and C. Ronse *et al*., 2013. Filtering and segmentation of 3D angiographic data: Advances based on mathematical morphology. Med. Image Anal., 17: 147-164. DOI: 10.1016/j.media.2012.08.004

Geetha, R. and U.G. Srikanth, 2012. Ant colony optimization in different engineering applications: An overview. Int. J. Comput. Applic., 49: 19-25. DOI: 10.5120/7720-1091

Hancock, E.R., R.C. Wilson, T. Windeatt, I. Ulusoy and Escolano, 2010. Structural, Syntactic and Statistical Pattern Recognition. 1st Edn., Springer, Berlin, ISBN-10: 3540372369, pp: 939.

Hansen, N., R. Ros, N. Mauny, M. Schoenauer and A. Auger, 2008. PSO facing non-separable and ill-conditioned problems. Universite Paris.

Ibric, S., J. Djuris, J. Parojcic and Z. Djuric, 2012. Artificial neural networks in evaluation and optimization of modified release solid dosage forms. Pharmaceutics, 4: 531-550. DOI: 10.3390/pharmaceutics4040531

Jiang, Y., T. Hu, C. Huang and X. Wu, 2007. An improved particle swarm optimization algorithm. Applied Math. Comput., 193: 231-239. DOI: 10.1016/j.amc.2007.03.047

Ma, Z., J.M. R.S. Tavares, R.N. Jorge and T. Mascarenhas, 2010. A review of algorithms for medical image segmentation and their applications to the female pelvic cavity. Comput. Meth. Biomechan. Biomed. Eng., 13: 235-246. DOI: 10.1080/10255840903131878

Rossana, M.S., H. Cruz, M. Peixoto and R.M. Magalhães, 2011. Artificial neural networks and efficient optimization techniques for applications in engineering. Federal Institute of Education.

Saishanmuga, V.R. and S.P. Rajagopalan, 2012. A neuro-genetic system for face recognition. Int. J. Comput. Sci., 9: 264-267.

Shen, L.L., Y.X. Wang and L. Duan, 2011. Application of artificial neural networks on the Prediction of surface ozone concentrations. Huan. Jing Ke Xue, 32: 2231-5PMID: 22619942