Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IMPACT FACTOR: 6.017

IJCSMC, Vol. 8, Issue. 3, March 2019, pg.223 – 229

CLASSIFICATION ON BREAST CANCER USING GENETIC ALGORITHM TRAINED NEURAL NETWORK

Hiba Badri Hasan; Asst Prof. Dr. Sefer Kurnaz

Electrical and Computer Engineering

Electrical and Computer Engineering

hbbd999@yahoo.com; sefer.kurnaz@altinbas.edu.tr

Abstract— Artificial neural networks have been in the position of producing complex dynamics in control applications over the last decade, especially when they are linked to feedback. Although ANNs are strong for network design, the harder the design of the network, the more complex the desired dynamic is. Many researchers tried to automate the design process of ANN using computer programs. Search and optimization problems can be considered as the problem of finding the best parameter set for a network to solve a problem. Recently, the problem of optimizing ANN parameters to train different research datasets has been targeted by two commonly used stochastic genetic algorithms (GA). The process based on the neural network is optimized with GA to enable the robot to perform complex tasks. However, using such optimization algorithms to optimize the ANN training process cannot always be balanced or successful. These algorithms simultaneously aim to develop three main components of an ANN: synaptic weight, connections, architecture and transfer functions set for each neuron. Developed with the proposed approach, ANN is also compared with hand-designed Levenberg-Marquardt and Back Propagation algorithms.

Keywords— BREAST CANCER; GENETIC ALGORITHM; TRAINED; NEURAL NETWORK

I. INTRODUCTION

The neural system is arranged into hidden layers, input and output of the Artificial Neural Networks (ANNs). The neurons are joined together by a series of synaptic weights. An ANN is a powerful tool for identifying patterns, predictions and regressions in a variety of problems. The ANN continually changes its synaptic values during the learning process until sufficient acquired knowledge (until a certain number of iterations have been achieved or the error value of the target has been achieved). After completion of the learning or training stage, the ability of the ANN to generalize the problem with samples other than those employed during the training stage must be assessed. Finally, during training and testing, it is expected that the ANN will be able to accurately classify the patterns of a particular problem. In recent years several classic ANN algorithms have been suggested and developed. However, many of them can stay trapped in nondesirable solutions; that is, they will be far from the optimum or the best solution. Moreover, most of these algorithms cannot explore multimodal and noncontinuous surfaces.

Therefore, other kinds of techniques, such as bioinspired algorithms (BIAs), are necessary for training an ANN. As BIAs are strong optimization tools and can resolve very complicated optimization problems, they are well accepted by the Artificial Intelligence Community. For a given problem, BIAs can browse large multimodal and continuous search areas and find the optimum value for the best solution. BIAs is based on nature's behavior described as swarm intelligence. The term [1] defines this concept as being owned by unintelligent agents with limited individual capacities, but intelligent collective behavior.

There are several studies which use evolutionary and bio - inspired algorithms as a basic way of training ANN [2]. Metaheuristic methods for training neural networks are based on local search, population methods, and others such as cooperative coevolutionary models [3]. An excellent work in which the authors show a comprehensive literature review of evolving ANN algorithms [2]. The majority of the research reports however focus on the development of synaptic weight, parameters [4] or the evolution of the numbers of the neurons for hidden layers, but the designer determines the number of hidden layers previously. Moreover, the researchers do not involve the evolution of transfer functions, which are an important element of an ANN that determines the output of each neuron. In [5], for instance, the authors suggested a combining of the Ant Colony Optimization (ACO) methodology to identify ANN and PSO for weight adjustment. Additional studies such as the [6] modify the Simulated Annealing (SA) PSO for the acquisition of a set of synaptic weight and threshold ANNs. In [7], the authors use Evolutionary Programming to obtain the architecture and weight to solve problems of classification and forecasting. Another example is [8] where Genetic Programming is used to obtain graphs that represent different topologies. In [9], the Differential Evolution (DE) algorithm was applied to design an ANN to solve a weather forecasting problem. In [10], the authors use a PSO algorithm to modify the synaptic weights to model the relationship between daily rains and runoffs in Malaya. In [11], the authors compare the back-propagation method versus basic PSO to adjust only the synaptic weights of an ANN for solving classification problems. In [12], the set of weights are evolved using the Differential Evolution and basic PSO. The three principal elements of an ANN were simultaneously developed in other works such as [13]: architecture, transfers and synaptic weights. In [14] the authors solved the same problem with a differential Evolution (DE) algorithm and suggested a new model with a PSO (NMPSO) algorithm for the authors. The author has also used [15] to develop the design of an ANN with two different fitness functions by using an Artificial Bee Colony (ABC) algorithm.

Therefore in this research work, we proposed a technique that uses PSO for ANN training to improve the training and testing performance of existing ANN on the diabetes dataset.

II. GENETIC ALGORITHM

The biological genetic algorithm metaphor is the development of species by the fittest survival as described by Charles Darwin. The crossover of genetic information between two parents in an animal or plant population produces a new person. The DNA stores the genetic data for the individual's construction. The genome of human DNA comprises 46 chromosome, four strings, abbreviated A, T, G and C. One of twenty amino acids, or a' start protein building' or' stop protein building ' signal, is translated into three bases. A total of about 3 billion nucleotides are present. These can be structured into genes that contain information about the individual's construction in one or more parts. However, the vast majority of the genes, the "junk" genes, have no meaningful information and only 3% of all genes. Genetic data, the genome itself, are called the individual's genotype. The result is called a phenotype. The individual. The same genotype could lead to various phenotypes. This is clearly illustrated by the Twins. The process of natural evolution is simulated by a genetic algorithm. It aims at optimizing a number of parameters. The original idea contains genetic data in a bit string of a fixed length which is called a parameter string or an individual. Everything is called a certain value. This thesis employs a range of different coding techniques, but the basic principles apply as well. A possible solution to this problem is provided by each parameter string. It includes information about building a neural network for the GANN issue. The fitness value is the quality of the solution. Crossover, selection and mutation are the fundamental GA operators. The main structure of a genetic algorithm is shown in Figure 1. This starts with the random generation of the original population, an early group of people.

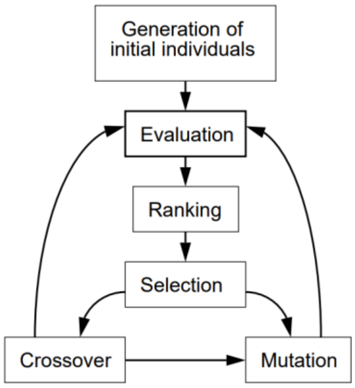

Figure 1. Structure of the principal genetic algorithm

Evaluate and classify individuals. Since the number of persons is constant in each population, an old person, generally the one with the worst fitness value, must be discarded for each new individual. Two basic operators are available to create new people: mutation and intersection. Mutation is simpler. During mutation, some bits of the parameter string are rotated randomly. Mutation may apply as an independent operator to offspring created through crossover or randomly to any person in the population.

III. TRANSFER FUNCTIONS

The threshold or transfer feature is also known as activation functions. The activating functions have been used to transform neuron activation levels into output signals. Numerous activation functions are available in the neural network. Identity function, step function, part linear function and sigmoid function are various function types.

A. Identity activation function:

The activation function of identity is also referred to as "liner activation." The Network Activation function can be shown easily to fit a line regression model of the form if the ID is used in the network $Y_i= B_0+ B_1+\cdots+B_k$ where $x_1,x_2,\ldots\ldots\ldots,x_k$ are the k network inputs, Y_i is the i_{th} network output B_1,B_2 ,……..,B_k are the coefficients in the regression equation. Consequently, a neural network with identity activation used in all its sensors is uncommon to be found.

B. Sigmoid activation function:

Nonlinearity in the model is used in the artificial neural network sigmoid functions. The result of a linear combination of its input signals is calculated by a network neuroelement using a signmoid function. The sigmoid function makes an interface between the product and itself easier and more popular in the Neural Network.

$$\varphi(v)= 1/(1+\exp(-av)) \qquad (1)$$

Sigmoid function results are generally used in learning algorithms. The Sigmoid graph is shaped as' S'. This function is defined as an expanding function which is commonly used for neural artificial network development. Sigmoid is a function that strictly increases, and shows a balance between linear and nonlinear functions.

One - polar – is the sigmoid function.

C. Step function:

This is a unipolar threshold, known as.

$$\varphi(v) = \begin{Bmatrix} 1\ if\ v\ \geq 0 \\ 0\ if\ v > 0 \end{Bmatrix} \qquad (2)$$

The neuron K output with a threshold is

$$y(k) = \begin{Bmatrix} 1\ if\ v_k\ \geq 0 \\ 0\ if\ v_k\ > 0 \end{Bmatrix} \qquad (3)$$

v_k *is th e induced local field of th e neuron*

$$v_k = \sum_{j=1}^{m} W_{kj}\ X_j + b_k \qquad (4)$$

When the neuron output is 1 when the local neuron field induced is not - negative, the neuron output is 0.

D. Piece Wise Linear Function

It can be defined as a unipolar function

$$\varphi(v) = \begin{Bmatrix} 1, & v \geq +1/2 \\ v, +\frac{1}{2} > & v > -1/2 \\ 0, & v \leq -1/2 \end{Bmatrix} \qquad (5)$$

When it is expected that the amplification factor is within the linear area

The specific circumstances of linear functions are

- If the linear operating area is maintained without saturation, a linear combiner is produced.
- If the linear region's amplification factor is infinitely large, it reduces to a threshold feature.

E. Learning Rules in neural network

There are many different types of study rules in the neural network, usually divided into 3 categories.

A. Supervised Learning
B. Unsupervised Learning

A. Supervised Learning

Training sets are available for supervised learning. This type of rule includes a set of examples with proper network behavior. The inputs are given as a training in controlled learning and the expected results are achieved.

Parameters in this type of study are adjusted step by step by error signal; parameters are adjusted step by step by step by error signal.

A number of examples (trainings set) together with correct network conduct are provided for the learning rule.

$$\{x1, d1\}, \{x2, d2\}, \ldots\ldots\ldots, \{xn, dn\} \qquad (6)$$

The network input is xn in this case and dn is the required destination input. The output is generated by input. In order to make network outputs more exact, the Study rule is employed to change network biases and weights.

We commit ourselves with supervised learning to give the system the desired answer (d) when the entry is implemented. The distance between the real response and the desired response is used to correct the network parameter externally. For example, the error can be used to change weighing in the study of input patterns or circumstances where the answer to the error is recognised. For the learning mode, the training set, several input and output patterns are needed.

B. Unsupervised learning

In unexpected learning, self-organized learning is also known. Objective output is not available in uncontrolled learning. In that case, only network input changes weights and biases. For pattern reorganization, unattended study grouping is used. The answer required is not known in unattended learning, therefore explicit error information cannot be utilized for improving network behaviour. Information of this type is not available to correct the wrong answers so that learning has to be done based on observations of marginalized or unknown responses to the data.

The algorithms in unchecked learning use redundant row data, which have no etiquette for class membership or associations. In order to identify its parameters in this way, the network needs to detect any existing patterns, properties, regulations, etc. Unattended study means learning without the teacher because it is not necessary for the teacher to participate, but the teacher must set objectives. Feedback on neural networks is important as well. Feedback is called progressive learning, which for uncontrolled learning is very important.

IV. ARTIFICIAL NEURAL NETWORK OUTPUT

Sensitivity, septicity and accuracy are preferred statistics for determining the performance of a classifier. Susceptibility is the estimation rate for patients with epileptic diseases, speciality is the estimation speed for healthy people and accuracy is true. Equality. These statistical numbers are calculated using (36), (37) and (38).

Sensitivity= TP/(TP+FN) (36)

Specificity= TN/(TN+FP) (37)

Accuracy= (TP+TN)/(TP+FP+TN+FN) (38)

In the above equations, the number of TP - diagnosed epilleptic patients, the total number of normal epileptic patients, for whose epileptic disease was diagnosed, and the total number of normal epileptic patients, for which FN was diagnosed.

V. FITNESS FUNCTION

MSE is the error of the ANN output and the pattern of desire. The minimum MSE generator (see the following equation) is the best person here:

$$F_1 = MSE = \frac{1}{p.M} \sum_{\xi=0}^{p} \sum_{i=0}^{M} (d_i^{\xi} - y_i^{\xi})$$

Where yi is the ANN's output.

VI. IMPLEMENTATION AND RESULTS

Comparative Results

Dataset Preprocessing

The dataset had some missing values which was fix using SPSS's Estimation-Maximization (EM) algorithm and then processed using z-score for artificial neural network training.

Expectation–maximization algorithm

An EM - algorithm is an iterative method in statistics in which a model is based on unconsidered latent variables to identify the most likely or maximum posterior estimate (MAP). In order to determine the distribution of variables in latent variables in step E, EM iteration changes the performance of E-step, creating a programmable expectation function evaluated by the existing parameter estimate and a M-step, calculating the parameters that maximize the expected log likelihood at Step E.

Z-Score

Simply put, z is the default number of the average dataset. However, the standard differentiation below or above population is more technically measured. Also a z-point can be placed in the normal distribution curve as the default score. Z-scoring varies from-3 default deviations to + 3 standard deviations (where the distribution curves take the right side) to+ 3 standard deviations (the far-left one). You have to know the medium μ and standard population differences if you want z-score to be used.

Z values are a way of using "normal" populations to compare test results. There are thousands of results and units to be reached in test results or surveys. However, these results often appear insignificant. For instance, it could be good information to know that someone is 150 pound weight, but to look at an extensive database, especially if some weights are recorded in kilograms, if you want to compare it with the "average." A z - score can indicate where the person's weight is compared to the average population weight.

Breast Cancer Results

First we present the individual for ANN-GA using Breast Cancer dataset. We used 20 neurons the hidden layer. Figure 1 and 2 are showing the fitness or Root Mean Squared Error (RMSE) or error outcomes by using the existing ANN-GA approach for 2000 iterations.

Figure 1 contains the error calculation process for 2000 iterations for artificial neural network training with 20 neurons in the hidden layer. Figure 2 contains the prediction and the classification of artificial neural network compared with the targets attribute in the dataset.

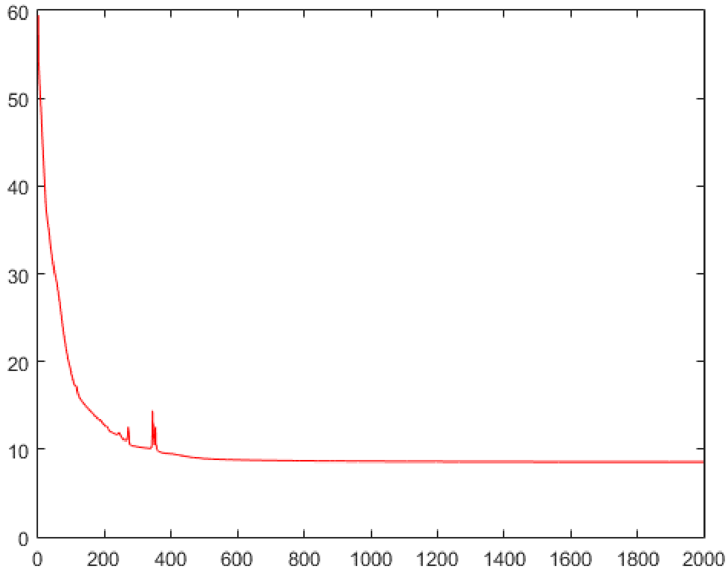

Figure 2: Error graph performance using ANN-GA for Breast Cancer Dataset

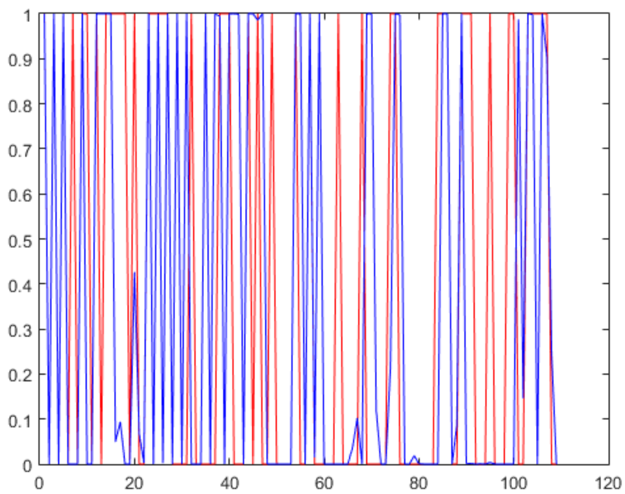

Figure 3: Error iteration graph performance using ANN-GA for Breast Cancer Dataset for original target and predicted outcomes.

Table 1. The collected results

Training Error (Fitness/RMSE)	0.0463
Training Accuracy	95.37%
Testing `Error(Fitness/RMSE)	0.06926
Testing Accuracy	93.074%
Training Sensitivity	0.980079
Training Specificity	0.939674
Testing Sensitivity	0.957832
Testing Specificity	0.910392

Table 1 contains the results for the method according to Figure 1 and Figure 2 which show the training error and accuracy, testing error and accuracy (using RMSE as a fitness function) and both the specificity and sensitivity for training and testing.

VII. CONCLUSIONS

The comparative results showing that using the advantages novel GA algorithm we can able to optimize the performance of ANN training and testing to solve the real time problems. From these experiments, we observed that the fitness functions that generated the ANN with the best weighted recognition rate were those that used the classification error. The GA was compared in terms of the accuracy, error rate, sensitivity rate, specificity rate and accuracy rate for both training and testing perspective with other researchers. The GA algorithm achieved the greate performance. The transfer functions that more often were selected for each algorithm were: the Gaussian functions for the basic BP algorithm; the sinusoidal function for GA algorithm. In general, the ANNs designed with the proposed methodology were very promising. The proposed methodology automatically designs the ANN based on determining the set connections, the number of neurons in hidden layers, the adjustment of the synaptic weights, the selection of bias, and transfer function for each neuron.

REFERENCES

[1] G. Beni and J. Wang, “Swarm intelligence in cellular robotic systems,” in Robots and Biological Systems: Towards a New Bionics? vol. 102 of NATO ASI Series, pp. 703–712, Springer, Berlin, Germany, 1993.
[2] X. Yao, “Evolving artificial neural networks,” Proceedings of the IEEE, vol. 87, no. 9, pp. 1423–1447, 1999.
[3] E. Alba and R. Martí, Metaheuristic Procedures for Training Neural Networks, Operations Research/Computer Science Interfaces Series, Springer, New York, NY, USA, 2006.
[4] J. Yu, L. Xi, and S. Wang, “An improved particle swarm optimization for evolving feedforward artificial neural networks,” Neural Processing Letters, vol. 26, no. 3, pp. 217–231, 2007.
[5] M. Conforth and Y. Meng, “Toward evolving neural networks using bio-inspired algorithms,” in IC-AI, H. R. Arabnia and Y. Mun, Eds., pp. 413–419, CSREA Press, 2008.
[6] Y. Da and G. Xiurun, “An improved PSO-based ANN with simulated annealing technique,” Neurocomputing, vol. 63, pp. 527–533, 2005.
[7] X. Yao and Y. Liu, “A new evolutionary system for evolving artificial neural networks,” IEEE Transactions on Neural Networks, vol. 8, no. 3, pp. 694–713, 1997.
[8] D. Rivero and D. Periscal, “Evolving graphs for ann development and simplification,” in Encyclopedia of Artificial Intelligence, J. R. Rabuñal, J. Dorado, and A. Pazos, Eds., pp. 618–624, IGI Global, 2009.
[9] H. M. Abdul-Kader, “Neural networks training based on differential evolution algorithm compared with other architectures for weather forecasting34,” International Journal of Computer Science and Network Security, vol. 9, no. 3, pp. 92–99, 2009.

[10] K. K. Kuok, S. Harun, and S. M. Shamsuddin, "Particle swarm optimization feedforward neural network for modeling runoff," International Journal of Environmental Science and Technology, vol. 7, no. 1, pp. 67–78, 2010.

[11] B. A. Garro, H. Sossa, and R. A. Vázquez, "Back-propagation vs particle swarm optimization algorithm: which algorithm is better to adjust the synaptic weights of a feed-forward ANN?" International Journal of Artificial Intelligence, vol. 7, no. 11, pp. 208–218, 2011.

[12] B. Garro, H. Sossa, and R. Vazquez, "Evolving neural networks: a comparison between differential evolution and particle swarm optimization," in Advances in Swarm Intelligence, Y. Tan, Y. Shi, Y. Chai, and G. Wang, Eds., vol. 6728 of Lecture Notes in Computer Science, pp. 447–454, Springer, Berlin, Germany, 2011.

[13] B. A. Garro, H. Sossa, and R. A. Vazquez, "Design of artificial neural networks using a modified particle swarm optimization algorithm," in Proceedings of the International Joint Conference on Neural Networks (IJCNN '09), pp. 938–945, IEEE, Atlanta, Ga, USA, June 2009.

[14] B. Garro, H. Sossa, and R. Vazquez, "Design of artificial neural networks using differential evolution algorithm," in Neural Information Processing. Models and Applications, K. Wong, B. Mendis, and A. Bouzerdoum, Eds., vol. 6444 of Lecture Notes in Computer Science, pp. 201–208, Springer, Berlin, Germany, 2010.

[15] B. A. Garro, H. Sossa, and R. A. Vazquez, "Artificial neural network synthesis by means of artificial bee colony (abc) algorithm," in Proceedings of the IEEE Congress on Evolutionary Computation (CEC '11), pp. 331–338, IEEE, New Orleans, La, USA, June 2011.

