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.216 – 222

DIAGNOSIS ON LUNG CANCER USING ARTIFICIAL NEURAL NETWORK

Mohammed Khalaf Abdullah; Asst Prof. Dr. Sefer Kurnaz

Electrical and Computer Engineering

Electrical and Computer Engineering

mhmd.khlf2016@gmail.com; sefer.kurnaz@altinbas.edu.tr

Abstract— Artificial neural networks in the last decade, especially when linked to feedback, have been able to produce complex dynamics in control applications. Although network designs are robust by the ANNs, the more difficult the network design is, the more complex it is. Many investigators tried to automate ANN's computer programs design process. Search and optimization problems can be taken into account as the difficulty of identifying the best network parameter to solve a problem. Two commonly used stochastic genetic algorithms (GA) have recently addressed the problem of optimizing ANN parameters to train different research datasets. The process is optimized using GA to allow the robot to perform complex tasks based on the neural network. However, it cannot always be balanced or successful to use these optimisation algorithms to optimize the ANN training process. These algorithms are designed to develop the synaptic weight, connections, Architecture, and Transfer functions of each neuron, three key components for an ANN at the same time.

Keywords— DIAGNOSIS; LUNG CANCER; ARTIFICIAL NEURAL NETWORK

I. INTRODUCTION

The neural systems are arranged to hide the Artificial Neural Networks (ANNs), the input and the output of them. A series of synaptic weights combine the neurons. An ANN is a powerful tool in a number of problems for the determination of patterns, predictions and regressions. During the learning process, the ANN constantly changes its synaptic values until sufficient knowledge is acquired (unless a number of iterations are achieved or the error value of the target is met). The ability of the ANN to generalize the problem in samples other than those used during the training stage must be evaluated following a completion of the learning or training. Finally, the ANN is expected to correctly classify the patterns of a specific problem during training and testing. Several classic ANN algorithms were proposed and developed in recent years. Many of them can, however, remain caught up in unsolicited solutions; they are far from the ideal or the best solution. In addition, most of these algorithms cannot investigate multimodal or non-continuous surfaces.

Other types of techniques are therefore required to train an ANN, such as bio-inspired algorithms (BIAs). The Artificial Intelligence Community is well accepted because BIAs are strong optimisation instruments and able to solve very complex problem optimisation problems. BIAs can scan large multimodal and continuous search areas for a certain problem and find the optimal value for the best solution. BIAs are based on the behaviour of nature called swarm intelligence. This concept is defined by [1] as owned by unintelligent agents of limited individual capacity, but intelligent collective behaviour.

Several trials use evolutionary and organically inspired algorithms as a fundamental method of ANN training [2]. In neural networks metaheuristic methods are based on local searches, population and other methods, such as cooperative models [3]. The authors present an excellent review of evolving ANN algorithms [2]. An excellent work. The majority of research reports focus, however, on the development and development of synaptic weight, parameters [4] or the evolution of neuronal numbers for hidden layers. Moreover, researchers do not involve the development of transmission functions, an important element of an ANN that determines each neuron's output. In [5], for example, the authors proposed the combination of ANN and PSO for weight-adjustment with Ant-Colony-Optimization (ACO) methodology. Further studies such as [6] amend the Simulated Annealing PSO (SA) to acquire a set of ANNs with synaptic weights and thresholds. In [7], authors use Evolutionary Programming to get the architecture and weight to solve classification problem and prediction problem. Another example is Genetic Programming [8] where graphs representing different topologies have been obtained. In [9] an ANN was designed to solve a weather forecasting problem with the differential evolution (DE) algorithm. In [10], the authors use a synaptic weight algorithm to change the relationship between daytime rainfall and runoff in Malaya. In [11] the authors only adjust synaptic weights of an ANN to solve classification issues by compared the back-propagation method to the base PSO. In [12] the weighing set is developed by the differential evolution and fundamental PSO. In other works, such as the architecture, transfers and synaptic weights, the three principal features of the ANN were also developed. With the Evolution (DE) differential algorithm [13], the authors resolved the same problem and suggested a new pattern with the authors ' NMPSO (PSO) algorithm. In addition, [14] the author has used an algorithm of the Artificial bee colony (ABC) to develop an ANN with two different fitness functions.

In this research, we therefore proposed a technique using Backpropagation for ANN education supported with genetic algorithm for parameter optimization for better training and testing on the dataset of diabetes for existing ANN.

II. GENETIC ALGORITHM

The biological genetic algorithm is the development of species by their survival, as described by Charles Darwin. The crossover of genetic information between two parents in a animal or plant population is the production of a new individual. The DNA stores the genetic data for the building of the individual. 46 chromosomes, four strings, abbreviated A, T, G, and C are part of the human DNA genome. One of twenty amino acids is translated into three bases: one' start protein building' or' stopping protein building' signal. There are approximately three billion nucleotides. These may be structured into genes containing information on the construction of the individual in one or more components. However, the vast majority of genes -the "junk" genes -are not used, and only 3% of all genes contain important data. Genetic information, the genome itself, is called the genotype of the person. This results in a phénotype. The person. Different genotypes might result in the same genotype. The Twins illustrate this clearly. A genetic algorithm simulates the process of natural development. It is intended to optimize several parameters. The original concept includes the genetic information in a bit string of a fixed length called a parameter string or an individual. Everything is referred to as a possible value. This thesis employs a range of different encoding techniques but also the basic principles. Each string of parameters provides a possible solution to this problem. It contains information on the construction of a

GANN neural network. The quality of the solution is the fitness value. The fundamental GA operators are crossover, selection and mutation. Figure 1 shows the principal structure of a genetic algorithm. It starts with the random generation, the original population of an early group of people.

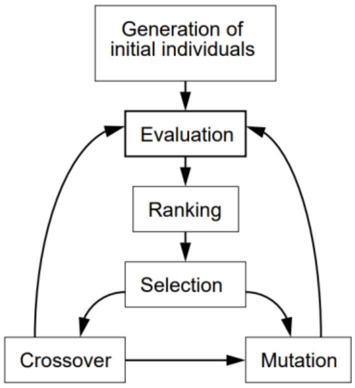

Figure 1. Structure of a genetic algorithm

Assess and classify people. As there is a constant number of people in each population, for each new individual, an old person, normally the oldest fitness entity, must be rejected. To create new people, two basic operators are available: mutation and intersection. It's simpler to mutate. Some bits of the string parameter are rotated during mutation randomly. Crossover creates offspring or any individual in the population, as an independent operator, may be affected by mutation.

III.TRANSFER FUNCTIONS

Also known as activation functions is the threshold or transfer feature. The functions used to activate the neuron are converted into output signals. There are a number of activation features in the neural network available. Different function types include identity function, step function, linear part function and sigmoid function.

A. Identity activation function:

The Network Activation function can be shown to fit a line form regression model, if the ID is used on a Yi= B0 + B1 + ADB network with a number of x1,x2,...,xk are the k network inputs, Yi is the B1,B2,.... ,Bk is the coefficient in the regression equation. The Network Activation function is also known as the liner activating function. Therefore, it is uncommon in all of its sensors to find a neural network with identity activation.

B. Sigmoid activation function:

In the neural artificial network sigmoid functions the model's nonlinearity is used. A linear combination of their input signals is calculated using a signmoid function by a network neuroelement. The Sigmoid function facilitates and enhances the interface within the Neural Network between a product and itself.

$$\varphi(v)= 1/(1+\exp(-av)) \tag{1}$$

In learning algorithms, sigmoid function results are generally used. The figure of Sigmoid is 'S' formed. This function is defined as a growing function commonly used for the development of artificial neural networks. Sigmoid is a function that increases strictly and displays a balance of linear and nonlinear functions.

One - polar – is the sigmoid function.

C. Step function:

This is a unipolar threshold, known as.

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

The neuron K output with a threshold is

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

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

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

When the neuron output is 1. The neuron output is 0 when the induced neuron field is not negative.

D. Piece Wise Linear Function

It can be defined as a unipolar function

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

If the amplification factor is expected to be within the linear zone

The particular conditions of linear functions are

- A linear combiner is produced if the linear operating area is kept without saturation.
- If the amplification factor in the linear region is infinitely wide, the threshold function is reduced.

E. Learning Rules in neural network

In the neural network, there are many different kinds of study rules, usually divided into two categories.

A. Supervised Learning
B. Unsupervised Learning

A. Supervised Learning

For supervised learning, training sets are available. This rule contains a number of examples with correct network behaviour. The inputs are provided as a controlled learning training and the expected outcomes are achieved. Parameters are set step by step by error signal in this type of study; the parameters are set step by step by error signal.

The learning rule contains a number of examples (trainings set) with the right networking behaviour.

$$\{x1, d1\}, \{x2, d2\}, \dots\dots\dots, \{xn, dn\} \quad (6)$$

In this case, the network input is xn and dn is the destination input required. The input produces the output. The study rule is used to change network biases and weights to make network outputs more accurate. We undertake supervised learning in order to provide the system with the required response (d) when the entry is implemented. To correct the network parameter externally, the distance between the actual answer and the desired answer is used. For example, in the study of input patterns or circumstances where an error response is recognised, the error can be utilized to change the weighting. The training set, multiple input and output patterns are required for the learning mode.

B. Unsupervised learning

Auto-organized education is also known in unexpected learning. In uncontrolled learning, objective output is not available. In this case, only the weight and baises of the network input change. Unattended study grouping is used for pattern reorganization. Unattended learning does not know the answer, so explicit data for errors cannot be used to enhance network behaviour. Information of this type does not exist to correct the wrong responses, so that it is necessary to learn about marginalized or unknown reactions to the data.

Unchecked learning algorithms use redundant row data that have no tag for classmates or associations. The network must detect any existing patterns, properties, regulations, etc, if its parameters are to be identified in this way. Unattended learning means learning without a teacher because the teacher does not need to take part, and the teacher needs to set targets. It is also important to have feedback on neural networks. Feedback is called gradual learning, which is very important for uncontrolled learning.

IV. ARTIFICIAL NEURAL NETWORK OUTPUT

The statistics preferred for determining classification performance are sensitivity, septicity and accuracy. Susceptibility for patients with epileptic illnesses is the rate of estimate, and accuracy is the rate of estimation for healthy individuals. Egalitarianism. The figures calculated using (36), (37) and (38) are statistical figures.

$$\text{Sensitivity} = TP/(TP+FN) \tag{36}$$

$$\text{Specificity} = TN/(TN+FP) \tag{37}$$

$$\text{Accuracy} = (TP+TN)/(TP+FP+TN+FN) \tag{38}$$

In these equations, the number of epilleptic patients diagnosed with TP - diagnosed, the total number of normal epileptic patients diagnosed with epileptic disease and the total number of normal epileptical patients diagnosed with FN.

V. FITNESS FUNCTION

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

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

Where yi is the ANN's output.

VI. IMPLEMENTATION AND RESULTS

This chapter presents the extensive simulation results for methods investigated in this project is Genetic Algorithm optimized structured Artificial Neural Network trained by Backpropagation GA (ANN-BP) by using research data source (Lung Cancer Dataset). We implemented the ANN using GA algorithm to optimize the parameters of ANN to train and test this research's dataset using BP in order to measure the different performance parameters.

Comparative Results

We used the 70 % training and 30 % testing scenario with varying number neurons of the hidden layer is 20 GA (ANN-BP).

Diabetes Dataset Results

First we present the individual for GA (ANN-BP) using Lung Cancer dataset. We used 5 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 GA (ANN-BP) approach for 100 iterations for GA and 200 for backpropagation.

Figure 1 contains the error calculation process for 100 iterations GA for parameters and 200 iterations for artificial neural network training with 5 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.

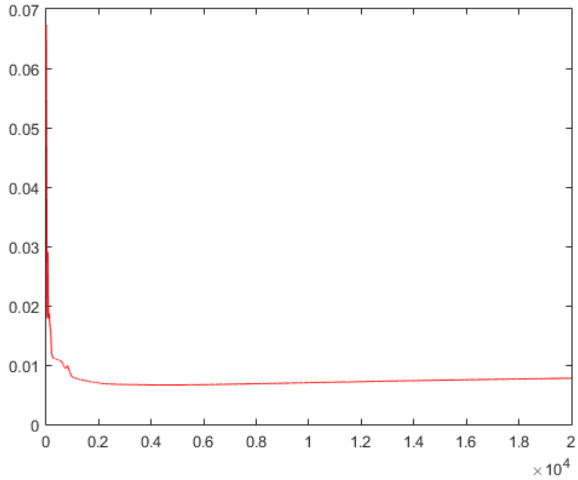

Figure 2: Error graph performance using GA (ANN-BP) for Lung Cancer Dataset

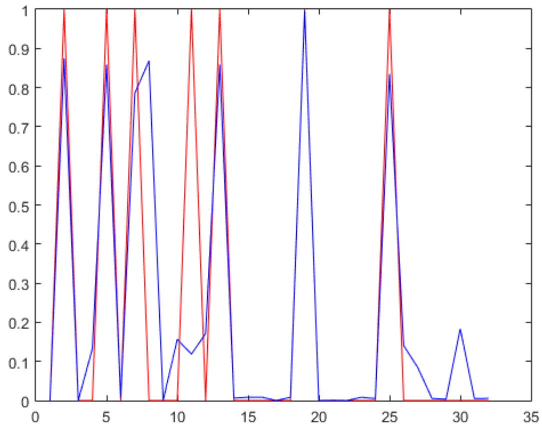

Figure 3: Error iteration graph performance using GA (ANN-BP) for Lung Cancer Dataset for original target and predicted outcomes.

Table 1. The collected results

Training Error (Fitness/RMSE)	0.1067
Training Accuracy	89.33%
Testing Error(Fitness/RMSE)	0.08726
Testing Accuracy	91.274%
Training Sensitivity	0.950079
Training Specificity	0.883084
Testing Sensitivity	0.949879
Testing Specificity	0.908452

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 BP algorithm with GA parameter modification 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 modified BP 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 modified BP 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 modified BP 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. 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.

[14] 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.

