Australian Journal of Basic and Applied Sciences, 9(11) May 2015, Pages: 39-46

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Selecting a Best Architecture of an Artificial Neural Network Using Genetic Algorithm for Lung Segmentation

[1]Saishanmuga raja V. and [2]Rajagopalan SP

[1]Assistant Professor, Dr. M.G.R Educational and Research Institute University, Department of Computer Science and Engineering, Faculty of Engineering and Technology, Chennai, India
[2]Professor Emeritus, Dr. M.G.R Educational and Research Institute University, Department of Computer Applications, Faculty of Engineering and Technology, Chennai, India

ARTICLE INFO

Article history:
Received 12 March 2015
Accepted 28 April 2015
Available online 5 May 2015

Keywords:
Multi layer perceptron neural network, Genetic Algorithm, Medical Image Segmentation, Lung segmentation.

ABSTRACT

Background: This paper introduces a hybrid approach in selecting a best architecture of artificial neural network using genetic algorithm. Neural networks are powerful tools for classification and regression, but it is time consuming and costly to determine the best architecture. Artificial Neural Networks are widely used for segmentation of tissues and structures from medical image. **Objective:** Through this study we prove the effectiveness of the Neuro-Genetic approach in selecting a best architecture of MLP neural network which is used in medical image segmentation, treatment plans, and evaluation of disease progression. **Results:** The existing methods do not consider the effect of architecture selection of neural networks which plays a major role in image segmentation. The results show that our approach as a technique to optimize the process of choosing the architecture of a MLP neural network that minimizes the error between target and actual output, training time, complexity of the ANN. **Conclusion:** The proposed Neuro-Genetic approach minimizes the training time and have increased the accuracy when compared to existing methods.

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: Saishanmuga raja V. and Rajagopalan SP, Selecting a Best Architecture of an Artificial Neural Network Using Genetic Algorithm for Lung Segmentation. ***Aust. J. Basic & Appl. Sci.,*** *9(11): 39-46, 2015*

INTRODUCTION

An artificial neural network (ANN) is a computational model for storing and retrieving acquired knowledge. ANNs consist of dense interconnected computing units that are simple models for complex neurons in biological systems (Ihme, 2008). The knowledge is acquired during a learning process and is stored in the synaptic weights of the inter-nodal connections. The main advantage of neural networks is their ability to represent complex input/output relationships. They are well suited for use in data classification, function approximation, and signal processing, among others. Segmentation of tissues and structures from medical images is the first step in many image analysis applications developed for medical diagnosis. Manual segmentation is usually accurate but cannot be used practically for large data sets because it is time consuming and complex. Automatic segmentation can be used in clinical applications if it they have 1) the ability to segment like an expert, 2) excellent performance for various datasets, 3) reasonable processing speed. Artificial Neural Networks (ANNs) have been developed for a wide range of applications such as function approximation, feature extraction, optimization, and classification. In particular, they have been developed for image enhancement, segmentation, registration, feature extraction, and object recognition. Among these, image segmentation is more important as it is a critical step for high-level processing such as object recognition. Multi-Layer Perceptron (MLP), Radial Basis Function (RBF), Hopfield, Cellular, and Pulse-Coupled neural networks have been used for image segmentation. These networks can be categorized into feed-forward (associative) and feedback (auto-associative) networks. MLP, Self-Organized Map (SOM), and RBF neural networks belong to the feed-forward networks while Hopfield, Cellular, and Pulse-Coupled neural networks belong to the feedback networks (Moghaddam, 2011).

Overview of neural network architecture:

The structure of a single neuron is shown in Fig. 1, where X {xi, i=1, 2, . . ., n}represent the inputs to the neuron and Y represents the output. Each input is multiplied by its weight w_i, a bias b is associated with each neuron and their sum goes through a transfer function f. As a result, the relationship

Corresponding Author: Saishanmuga raja V, Assistant Professor, Dr. M.G.R Educational and Research Institute University, Department of Computer Science and Engineering, Faculty of Engineering and Technology, Chennai, India.

between input and output can be described as follows.

$$Y = \sum_{i=1}^{n} W_i X_i + b \quad (1)$$

By selecting proper transfer functions and connections of neurons various neural networks can be constructed and trained to produce specified outputs. Learning of Neural networks in image processing involves supervised learning and un-supervised learning. In supervised learning a network is trained using a set of input and output values. In each training set, a set of associated input and output values are used and the goal is to minimize the overall output error of the network by adjusting the neuron connection weights and bias values.

In unsupervised learning, the training dataset does not have any target information. The user defines a function that measures the accuracy of the network. This function is called cost function and it is depends on the networks application. The unsupervised learning aims to minimize or maximize the cost for all the input vectors in the training set.

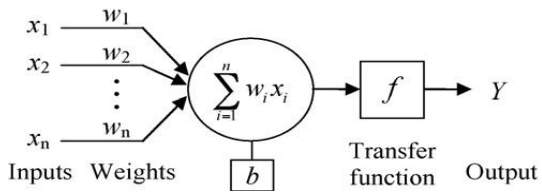

Fig. 1: Structure of a Single neuron.

The architecture or topological structure of an ANN can be characterized by the arrangement of the layers and neurons, the nodal connectivity, and the nodal transfer functions. In this work, the class of multi-layer perceptrons (MLPs) is considered which consists of an input layer with N_I input channels, N_L hidden layers, and an output layer with N_O output channels. The number of neurons in each hidden layer is denoted by N_N.

A multi-layer perceptron neural network is shown in Fig. 2(a). Information in this 2-3-2-1 network propagates uni-directionally from the input to the output channel. Each neuron includes a threshold which is indicated by a vertical arrow in Fig.2(a). In the case of a symmetric network, in which all neurons in a particular layer are connected to the same nodes, the nodal connectivity matrix can be contracted to a layer connectivity matrix L. The matrix entry L_{ij} indicates the connectivity between layer *j* and layer *i*. The layer connectivity matrix corresponding to the ANN of Fig. 2(a) is shown in Fig.2(b).

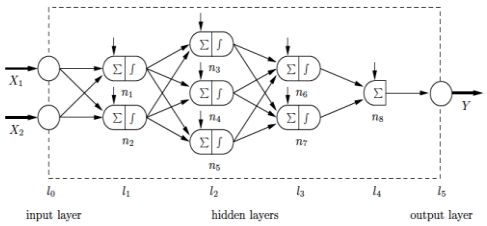

Fig. 2(a): Network Architecture

$$\underline{\underline{L}} = \begin{array}{c} \\ l_0 \\ l_1 \\ l_2 \\ l_3 \\ l_4 \\ l_5 \end{array} \begin{array}{c} \begin{array}{cccccc} l_0 & l_1 & l_2 & l_3 & l_4 & l_5 \end{array} \\ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Fig. 2(b): Layer connectivity matrix.

Feed Forward Neural Network:

Several neural network architectures are available for medical imaging segmentations out of which the most common is the feed forward network. The neurons in a feed forward network is connected unidirectional, which means the information being processed pass through the network in a single direction, from the input layer, through the hidden layers to the output layer. Feed-forward networks commonly use the back propagation (BP) supervised learning algorithm to dynamically alter the weight and bias values for each neuron in the network. The algorithm works by iteratively adjusting the connection weights of neurons based on the error in the network's actual output value when compared to the target output value. The actual modification of weights is carried out using a gradient descent algorithm, where the weights are modified after each training example is present to the network.

Radial basis function networks:

A radial basis function (RBF) network is a three-layer supervised feed-forward network that uses a nonlinear transfer function (normally Gaussian) for the hidden neurons and a linear transfer function for the output neurons. The Gaussian function is applied to the net input of each neuron to produce a radial function of the distance between each pattern vector and each hidden unit weight vector. The size and the topology of the RBF networks are flexible which makes them suitable for variety of applications. RBF networks have been successfully applied to a number of visual processing and analysis problems, including analysis of 3D structures, as well as time-series data.

Feedback Network:

In a feedback network signals or information travel in both directions introducing loops, sending signals from hidden layers and output layers backwards to earlier layers. The state of the network changes continuously till it reaches an equilibrium

point. It remains in the equilibrium point until the input changes and a new equilibrium is found.

A Hopfield network is a specific type of feedback neural network which acts as a form of associative memory in similar way as a certain part of human brain. The purpose of the associative memory is to cover a state remembered during training when only a part of the input is presented. The Hopfield network does not have any output or input neurons, all are connected directly to each other. It is simple to set up weights in a Hopfield network in order to set up a desired output patterns. They are very useful in medical image analysis where the output value must be derived from the partial or similar patterns those presented in training.

Self Organizing Map:

Self Organizing map (SOM) is a type of network that learns to map the input values for an output space. SOM maintains the topology of the input data by reducing the dimensionality making it suitable for visualizing problems. SOM are especially useful for medical imaging applications such as edge detection and segmentation as their ability to organize the neuron structure based on topographical structure of the inputs.

In SOM, each neuron is fed by the input vector (data point) $X \epsilon R^n$ through a weight vector $w \epsilon Rn$. Each time a data point is input to the network, only the neuron j whose weight vector most resembles the input vector is selected to fire, according to the following rule:

$$j= \text{argmin } (||x-w||^2),\ i=1, 2,.....m \quad (2)$$

The firing neuron j and its neighboring neurons i have their weight vectors w modified according to the following rule:

$$w_i(t+1) = w_i(t) + h_{ij} (||r_i-r_j||,t) . (x(t)-w_i(t)) \quad (3)$$

where $h_{ij}(||r_i-r_j||, t)$ is a kernel defined on the neural network space as a function of the distance $||r_i-r_j||$ between the firing neuron j and its neighboring neurons i, and the time t here defines the number of iterations. Its neighboring neurons modify their weight vectors so they also resemble the input signal, but less strongly, depending on their distance from the winner.

Neural networks for medical image segmentation:

Medical image segmentation involves in dividing a given image into important regions with similar properties. Image segmentation is an important process in outlining boundaries of organs and tumors and in the visualization of human tissues during clinical analysis. Therefore, segmentation of medical images is very important for clinical research, diagnosis, and applications, leading to requirement of robust, reliable and adaptive segmentation techniques. Image registration is done as a preprocessing step in medical imaging before image segmentation and edge detection.

Medical Image Registration:

Image registration is the process of transforming different data sets into a single coordiate system. Image registration is necessary to enable comparison, integration and fusion of images from different measurements. Medical image registration involves elastic registration to cope up with the elastic changes on the body part during imaging due to breathing or bodily changes over time. Medical image registration is a pre-procesing step in medical imaging which can help in the efficient resulting from edge detection and segmentaion. Medical image registration is an important technique for comparing and linking multiple related images from different points in time.

Image Segmentation:

(Kobashi *et al*, 2011) proposed an automated method to segment the blood vessels from3Dtime of flight (TOF) MRA volume data. The process of image segmentation consists of three-steps: (1) removal of the background, (2) volume quantization, and (3) classification of primitives by using an artificial neural network. After volume quantization by using a watershed segmentation algorithm, the primitives in the MRA image stand out. To further improve the result of segmentation, the obtained primitives have to been separated into the blood vessel class and the fat class. Three features and a feed-forward three-layered neural network are adopted for the classification. Compared with the fat, the blood vessel is like a tube—long and narrow. To this end, two features including vascularity and narrowness were introduced to measure such properties. As the histogram of blood vessels is quite different from that of the fat in shapes, a third feature, histogram consistency is added for further improvement of the segmentation.

(Jiang *et al*. 2010) applied a Hopfield neural network with penalized fuzzy c-means technique (called PFHNN) to medical image segmentation. In the algorithm, the pixels with their first and second order moments constructed from their n nearest neighbors as a training vector are mapped to a two-dimensional Hopfield neural network for the purpose of classifying the image into suitable regions. (Wang, 2011) developed a novel colony analysis system including an adjustable image acquisition subsystem and a wavelet-watershed–based image segmentation algorithm. The membership function simulated with neuron outputs is determined using a fuzzy set, and the synaptic connection weights between the neurons are predetermined and fixed to improve the efficiency of the neural network.

Edge detection:

(Jiang *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. The CHEFNN extends

the one-layer 2D Hopfield network at the original image plane to a two-layer 3D Hopfield network with edge detection to be implemented on its third dimension. With the extended 3D architecture, the network is capable of incorporating a pixel's contextual information into a pixel-labeling procedure. As a result, the effect of tiny details or noises will be effectively removed by the CHEFNN and the drawback of disconnected fractions can be overcome.

In addition, (Dourado,2008) discovered that high-level contextual information cannot be incorporated into the segmentation procedure in techniques using traditional Hopfield neural networks and thus proposed contextual constraint-based Hopfield neural cube (CCBHNC) for image segmentation. The CCBHNC uses a three dimensional architecture with pixel classification implemented on its third dimension. With the three-dimensional architecture, the network is capable of taking into account each pixel's feature and its surrounding contextual information, achieving up to 95.86% segmentation accuracy on real MRI images. Recently, still for the edge detection, (Chang, 2006) presented a special design Hopfield neural network called the contextual Hopfield neural network (CHNN). The CHNN maps the 2D Hopfield network at the original image plane. With the direct mapping, the network is capable of incorporating pixels' contextual information into an edge-detecting procedure. As a result, the CHNN can effectively remove the influence of tiny details and noise.

Genetic algorithm:

In neural network training, the most commonly used algorithms are versions of the back propagation algorithms developed by (Yin. 2011). The well known limitations of gradient search techniques applied to complex nonlinear optimization problems have often resulted in inconsistent and unpredictable performance. They typically start at a randomly chosen point (set of weights) and then adjust weights to move in the direction which will cause the errors to decrease most rapidly. These types of algorithms work well when there is a smooth transition toward the point of minimum error. But the surface of the neural network is not smooth. It is characterized by hills and valleys that cause techniques such as BPN trapped in local minimum.

Crossover Operation:

Since the individual coding of the evolutional sub-populations in this paper contains the structure and connecting weights, and different parts take different encoding methods, it is necessary to design corresponding genetic operations. For structural parts, we use binary coding and the crossover operation of the standard genetic algorithm, such as single point crossover, multi point crossover or same crossover and so on. As for the connection weights part, we use real coding, and the length of the real coding is equal to the number of 1 in the structure coding. Therefore the intersection of real-coded string should depend on the location of the intersection in structural part.

Mutation Operation:

Let us consider the structural parts and connection weights parts of evolutional individuals respectively. For structural parts, use the mutation operation of standard genetic algorithm, such as single point mutation, multi point mutation, or same mutation. For the connection weight parts, since its code is influenced by structure parts, when the code of a certain structural part changes from 0 to 1, the corresponding code of connecting weights parts should be generated randomly within a certain range.

Fitness Function Evaluation:

We set the input and output as (X_i, t_i), $i=1,2\ldots M_r$, and if input is X_i, the actual output of neural network is y_i. Now consider the P_{th}evolution sub-population. The j_{th}evolution individual is x_j^p , others are x_r^q , $q=1,2,\ldots p-1, p+1, N_p$, then the individual fitness calculation formula Eq.(4) is given below

$$F = \frac{1}{\frac{1}{Mr}\sum_{i=1}^{Mr}\left(t_i - y_i\left(x_r^1, x_r^2, \ldots x_r^{D-1}, x_r^D, x_r^{D+1}, \ldots . x_r^{N_p}\right)\right) + C} \quad (4)$$

where c is an appropriate constant.

Architecture Modelling:

In the previous section we have discussed about the importance of genetic algorithm for selecting appropriate network architecture. First we define the activation and combination function in the neural network. The objective of this study is to select an appropriate architecture of neural network based on genetic algorithm to be applied in medical image segmentation.

Implemented architectures:

We take the three important network architectures to test and compare their performance over image segmentation - *generic*, *layered*, and *architecture with variables selection*.

- Generic architectures: Simply a direct incorporation of Neural Network with no additional constraints added.
- Layered architectures: Well known type of architecture, in this case we wanted to prove whether it is represent able with Genetic Algorithm.
- Architectures with variables selection: This architecture is the example of intentionally designed architecture for special purposes.

Neural network with generic architectures are constructed without any additional constrains. The term generic means that it assigns a set of all architectures represented via different methods (Kalogirou, 2013). The generic type of architectures

can be implemented into a process of evolutionary optimization. Genetic programming is an automated method for creating a working computer program from a high-level problem statement of a problem. Genetic programming starts from a high-level statement of "what needs to be done and automatically creates a computer program to solve the problem. 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 (Raja, V. S and Rajagopalan, 2013). The technique of genetic programming is one of the techniques of the field of genetic and evolutionary computation which, in turn, includes techniques such as genetic algorithms, evolution strategies, evolutionary programming, grammatical evolution, and machine code (linear genome) genetic programming. The GP optimizes part of the parameters, usually the underlying architecture, and some other method optimizes the remaining ones, usually internal parameters of each neuron.

Fitness Function:

The fitness function is a quality measure which works as a main component of the genetic algorithm optimization. The individuals are optimized according to criteria evaluated by the fitness function during the genetic algorithm run.

The fitness is thus written as

$$f = w_{HST}\, e_{HST} + w_{ROC} e_{ROC} + w_{PO} e_{PO} + w_{MSE} e_{MSE} \qquad (5)$$

Suppose that both the factors e and weights w are non-negative, and thus also the resulting fitness f is non-negative. This will be important in later sections where we will discuss the advanced use of fitness values. In the next subsections we will describe and specify in deeper detail each of the evaluation factors. We will speak about the motivation and the particular definitions. We also list basic properties of the factors. In the last subsection we will return to the formula 4 defining the mixing of factors. We will look at the factor transformations that lead to normalized factors and consistent system of weights.

A very illustrative description of the run of the genetic algorithm is a graph of evolution of the fitness from the initial population to the last one, with generations in one axis and the particular members of each population in second. The graph is called *3D-fitness graph*. A two dimensional version of the fitness evolution shows only generations with maximum and average fitness over each population. The graph is called *2D-fitness graph*. Both of the mentioned graphs are shown in Fig.3. We can see in the 3D-fitness graph that the fitness really does not show no trend as it is randomly sampled independently on the particular phenotype. As expected, the values stretch from 0 to 20, with no significant structure.

Parameter Setting:

We already outlined all the parameters that are essential for the whole run of the genetic algorithm to proceed correctly and bring any contributive results. We did not, however, discuss on specific values of the parameters to set up so that we gain optimal behavior of all concerned parts of the genetic algorithm. The parameters are adjusted as soon as we apply the genetic algorithm to the specific data.

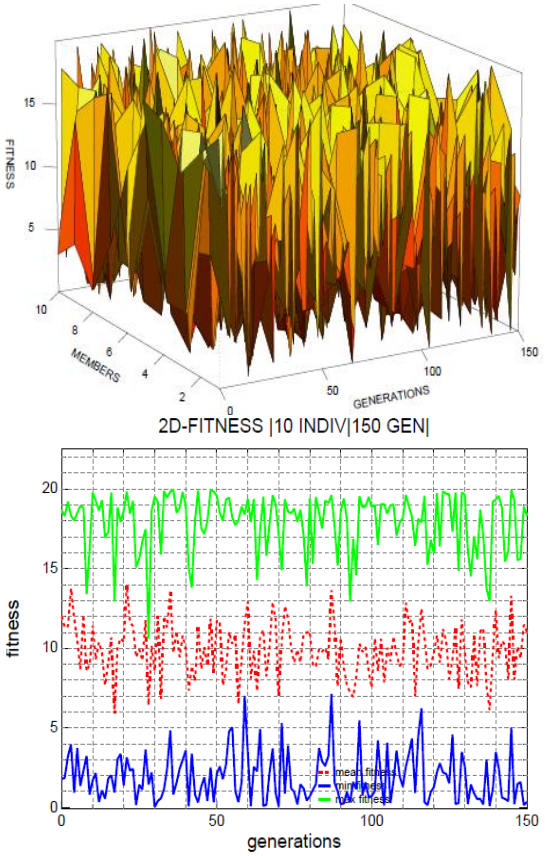

Fig.3: Two illustrations of a genetic algorithm run, a 3D-fitness (left) and 2D-fitness (right) graphs.

We have shown the table 1 only as an illustrative figure, but it is now that it becomes the real use and incorporation. Based on the 150 constructed ANN instances, learned on data, and evaluated with particular evaluation factors we calculated the means and also the empirical evaluation factors distributions estimates and used them to set up the transform functions *Ts* and *Tr*.

Table 1: Evaluation Factor

Evaluation Factor	Min	Max	Mean	Std
e_{HST}	6.9	8.22	7.30	0.1548
e_{PO}	0.59	0.77	0.64	0.0221
e_{ROC}	2.01	1.22	2.26	0.0621
e_{MSE}	1.22	1.29	1.27	0.0089

We can see that on the data we obtained sample means evaluated from 0:68 for e_{PO} to 7.70 for e_{HST} , the most volatile factor is the e_{HST} with standard deviation of 0.17. The ROC is used to evaluate the performance of the segmentation algorithm. But in our context we use e_{ROC} as an error factor.

Experimental Overview:

The first two genetic algorithm runs will proceed in 50 X 200 scenarios, i.e. with R = 50, and V = 200, evaluating 10,050 ANNs. The first was set up with five genotypes-memory (R^{new} = 45) to see a guided convergence, the second was setup without memory (R^{new} =R) the detailed set-up tables are listed in tables 2 and 3.

Table 2: Large Run Genetic Algorithm GA_1 with Memory.

GA_L01 parameter setup
MIN_{BC}= 3 MAX_{BC} = 25 MIN_{NSUC}= 1 MAX_{NSUC} = 3 MIN_{C1C}= 2 MAX_{C1C} = 10
Transform function: T_r ω_{HST} = 1.0 ω_{PO}= 0.5 ω_{ROC} = 1.0 ω_{MSE}= 0.1
P_C = 0.9 P_M = 0.1
R= 50 R^{new} = 45 *V= 200*

Table 3: Large Run Genetic Algorithm GA_2 without Memory.

GA_L02 parameter setup
MIN_{BC}= 3 MAX_{BC} = 25 MIN_{NSUC}= 1 MAX_{NSUC} = 3 MIN_{C1C}= 2 MAX_{C1C} = 10
Transform function: T_r ω_{HST} = 1.0 ω_{PO}= 0.5 ω_{ROC} = 1.0 ω_{MSE}= 0.1
P_C = 0.9 P_M = 0.1
R= 50 R^{new} = 45 *V= 200*

Table 4: Comparison 1; The Genetic Algorithm outperformed Random search (f^E_{GA} =0.995 against f^E_{RS} = 0.955)

Method	NN: nstance	f^E	f^T	ζ	η	ξ
GA	00139_00258	0.995	0.902	0.904	0.817	0.884
GA_L01	00124_00041	0.997	0.854	0.838	0.705	0.772
	00197_00035	1.000	0.785	0.784	0.654	0.702
Random	u008595	0.945	0.854	0.892	0.816	0.816
Search	u006828	0.952	0.747	0.778	0.621	0.668
RS_L01	u002728	0.961	0.752	0.789	0.584	0.685

Table 5: Comparison 2; The Genetic Algorithm outperformed Random search (f^E_{GA} =0.998 against f^E_{RS} = 0.971).

Method	NN: Instance	f^E	f^T	ζ	η	ξ
GA	00123_00044	1.000	0.913	0.914	0.823	0.879
GA_L02	00087_00014	0.999	0.837	0.838	0.703	0.771
	00049_00017	1.000	0.766	0.764	0.589	0.678
Random	u007361	0.971	0.799	0.798	0.642	0.729
Search	u009572	0.955	0.764	0.735	0.622	0.708
RS_L02	u004860	0.957	0.734	0.756	0.578	0.664

The Table 2 and Table 3 clearly show the large run of Genetic Algorithm with and without memory. The tables also show the transform function selected.

Fig.4 shows the fitness function evaluation of the genetic algorithm and the comparison of the random search algorithm and genetic algorithm are clearly shown in the Table 4 and Table 5.

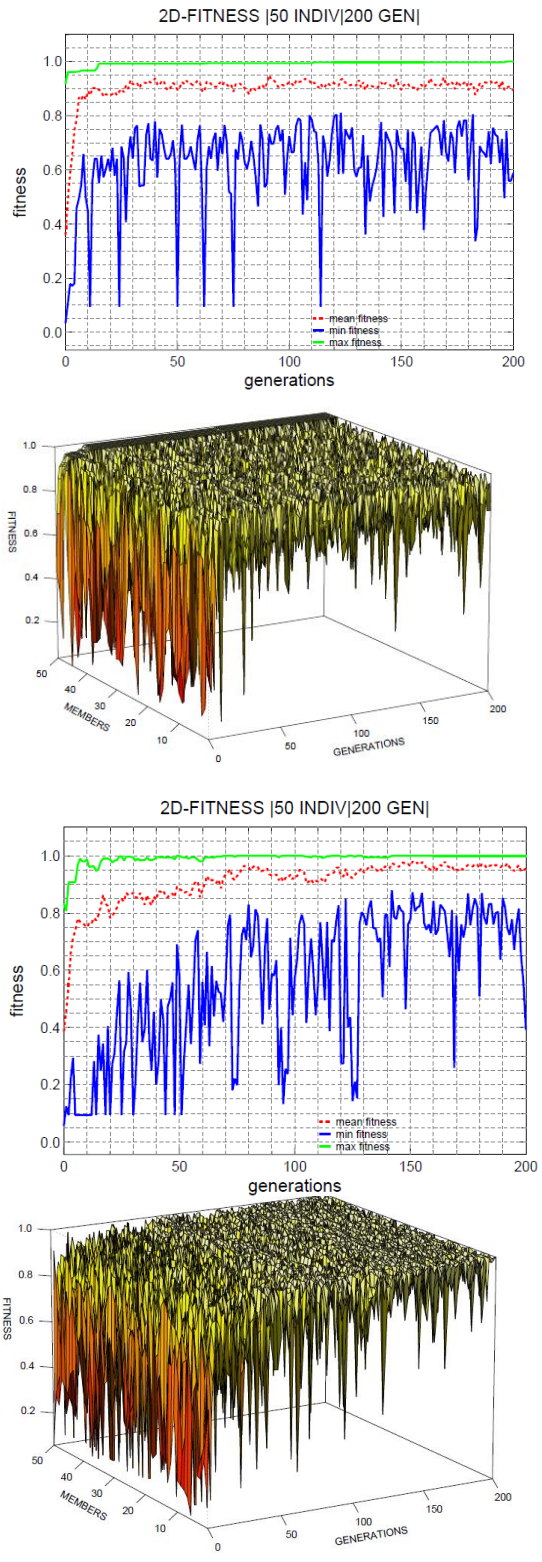

Fig. 4: Fitness Evolutions of GA_1 and GA_2 runs.

Conclusion:

The main goal of this study is to select the best architecture for an artificial neural network with the help of Genetic algorithm. This work naturally mixed the existing theory and findings with the new ideas and adjustments we needed to incorporate into the models in order to g*et al*l the parts of the ANN GA working properly and most efficiently. We defined a structured fitness function based on four evaluation factors—histogram ratio evaluation *eHST*, ROC curve evaluation *eROC*, posterior probabilities evaluation *ePO*, and a root mean square error based evaluation *eMSE*. The composed fitness then aggregates the values of individual evaluation factors into a scalar value. The two operators, mutation and cross-over follow natural extension of the operations. The third one, selection operator, was adjusted so that it would distinctively identify individuals with significantly different underlying fitness.

Using genetic algorithm we achieved to find solutions with level of required quality up to 100 % and moreover, the genetic algorithm manages to find them faster, i.e. with lower time to finding solution. This is due to the fact that the genetic algorithm searches for individuals simultaneously maximizing all factors. However, the limitation of this approach is that the optimization of neural network architecture using genetic algorithm comes with a high computational cost.

Future enhancements:

This approach is effective for segmenting deep brain structures in 3D. Its high speed makes it appropriate for real-time applications. Due to its speed and accuracy, it has a bright future for medical image segmentation applications.

ACKNOWLEDGMENT

I would like to thank my supervisor Dr. S.P. Rajagopalan for his great support in this study and Dr. V. Cyril raj, my co-supervisor for his guidance in my research work. All the results listed in this work are naturally based on results of "*NNSU tool for use of Neural Networks with Switching Units",* including the task specifications, output charts and graphs, neural networks visualizations, etc.

REFERENCES

Chang, C.Y., 2006. Contextual-based Hopfield neural network for medical image edge detection. Optical Engineering, 45(3): 037006-037006.

Dourado, A., J. Henriques, P.D. Carvalho, 2008. Neural, Fuzzy, and Neurofuzzy Systems for Medical Applications. Intelligent and adaptive systems in medicine, pp: 127-171.

Ihme, M., A.L. Marsden, H. Pitsch, 2008. Generation of optimal artificial neural networks using a pattern search algorithm: Application to approximation of chemical systems. Neural Computation, 20(2): 573-601.

Jiang, J., P. Trundle, J. Ren, 2010. Medical image analysis with artificial neural

networks. Computerized Medical Imaging and Graphics, 34(8): 617-631.

Kalogirou, S.A., 2013. Solar energy engineering: processes and systems. Academic Press.

Kobashi, S., N. Kamiura, Y. Hata, F. Miyawaki, 2001. Volume-quantization-based neural network approach to 3D MR angiography image segmentation. Image and Vision Computing, 19(4): 185-193.

Moghaddam, M.J., H. Soltanian-Zadeh, 2011. Medical Image Segmentation Using Artificial Neural Networks. INTECH Open Access Publisher.

Raja, V.S., S.P. Rajagopalan, 2013. A Comparative Analysis of Optimization Techniques for Artificial Neural Network In Bio Medical Applications. Journal of Computer Science, 10(1): 106.

Wang, W., 2011. Colony image acquisition system and segmentation algorithms. Optical Engineering, 50(12): 123001-123001.

Yin, F., H. Mao, L. Hua, 2011. A hybrid of back propagation neural network and genetic algorithm for optimization of injection molding process parameters.Materials & Design, 32(6): 3457-3464.