

Evolving Connection Weights of Artificial Neural Networks Using Genetic Algorithm with Application to the Prediction of Stroke Disease

¹D. Shanthi, ²G. Sahoo and ¹N. Saravanan

¹Department of Computer Science and Engineering, Birla Institute of Technology,
P.O. Box 31320, Budaiya, Kingdom of Bahrain, Bahrain

²Department of Information Technology, Birla Institute of Technology, Mesra, Ranchi, India

Abstract: In Artificial Neural Network (ANN), the selection of connection weights is a key issue. The usual weights' randomization methods are used to initialize the network weights before training. The main aim of weights randomization techniques is to avoid sigmoid saturation problems that cause slow training. There are different weights randomization methods such as Manual, Automatic, optimized for uniform distribution of networks, optimized for Gaussian distribution of network input and Random seed available. In this study, we proposed a new hybrid model of Neural Networks and Genetic Algorithm (GA) to initialize and optimize the connection weights of ANN so as to improve the performance of the ANN and the same has been applied in a medical problem of predicting stroke disease for verification of the results.

Key words: Genetic algorithm, artificial neural network, neuro-genetic hybrid, weight optimization

INTRODUCTION

In the past decade, 2 areas of research, which have become very popular are the fields of Neural Networks (NN) and Genetic Algorithms (GA). Both are computational abstractions of biological information processing systems and both have captured the imaginations of researchers all over the world. Many problems in life are solved through some kind of searching process. Although, gradient descent techniques have been used effectively train feedforward neural network connection weights, researchers have experimented with evolving an optimal set of connection weights with biologically inspired genetic algorithms.

Artificial neural network: ANN is collections of mathematical models that emulate the real neural structure of the brain. In general, ANN is made up of individual interconnected simple processing elements called neurons, arranged in a layered structure to form a network that capable of performing massively parallel computation (Chen and Wah Sit, 2005). A multi-layered feed forward ANN Architecture is presented in Fig. 1.

ANN is a network of many simple processors called units, linked to certain neighbors with varying coefficients of connectivity (called weights) that represent the strength of these connections. The basic unit of ANNs,

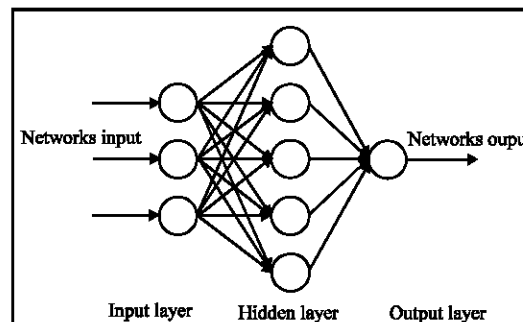


Fig. 1: Architecture of artificial neural network

called an artificial neuron, simulates the basic functions of natural neurons: it receives inputs, processes them by simple combination and threshold operations and outputs a final result. ANNs often employ supervised learning, in which training data (including both the input and the desired output) is provided.

Learning basically refers to the process of adjusting the weights to optimize the network performance. ANNs belongs to machine-learning algorithms because the changing of a network's connection weights causes it to gain knowledge in order to solve the problem at hand (Wang and John, 2006). ANN has the ability to learn complex relationship between the given set of input and output data. When presented with set of input and output

pairs, the network is able to learn the relationship between them by changing the weights of its interconnections. The process of changing the weights is called training the networks. Once, the network is trained, the weights will be frozen and that network can be used for prediction. ANN can be employed for optimization/resource allocation, pattern recognition and prediction (Kumanan *et al.*, 2006).

Genetic algorithm: Genetic Algorithm (GA) is a search algorithm based on survival of the fittest among string structures to form a search algorithm (Goldberg, 1989). For solution of optimization problems, GA has been investigated recently and shown to be effective at exploring a complex space in an adaptive way, guided by the biological evolution mechanisms of reproduction, crossover and mutation (Adeli and Hung, 1995). GA may avoid falling in to minimum when local search method like gradient descent is used in ANN. Genetic Algorithms perform the search process in four stage: initialization, selection, crossover and mutation (Davis, 1991). The following are the basic steps of genetic algorithm.

- Problem representation
- Initialize the population
- Calculate fitness
- Perform selection
- Perform crossover
- Perform mutation
- Check the convergence, if solution is not satisfied go to step 4
- End

In a typical genetic algorithm, the genetic code is a fixed-length bit string and the population is always a fixed size. The 3 most common propagation techniques are elitism, mutation and crossover. In elitism, the exact individual survives into the next generation. In mutation, a new individual is created from an old one by changing a small number of randomly selected bits in its gene. In crossover, a new individual is created from 2 old ones by randomly selecting a split point in their genes are creating a new gene with the left part from one parent and the right part from another. In any genetic algorithm, the 2 key aspects are the genetic representation and the fitness function. Together, these determine the type of problem, which is being solved and the possible solutions, which may be generated. GAs serve as an intelligent search and optimization technique and adaptation of network weights. GA was proposed as a mechanism to improve the performance of Artificial Neural Network (ANN) inferential estimator. GA is to decide the representation of connection weights in the form of binary

or real number. During the connection weights evolution, the architecture of an ANN is redefined and fixed. Genetic Algorithms are employed to optimize the neural network in variety of the forms such as feature selection, topology selection, training the network, determining the number of nodes in each layer, evolution of connection weights, evolution of learning rule etc. GA-NN combination provides powerful classification capabilities with tuning flexibility for either performance or cost-efficiency. In this study the Genetic Algorithm has been used for the evolution of connection weights in a Multi-layered feed forward ANN.

Problem definition: Stroke is a life-threatening event, in which part of the brain is not getting enough oxygen. There are different types of stroke namely Brain Attack, Embolic Stroke, Thrombotic Stroke, Ischemic Stroke, Cerebro Vascular Accident (CVA). Medical personnel treating a stroke are challenged to treat the patient as quickly as possible to avoid permanent tissue damage or death. Strokes were responsible for more deaths and nearly half of those deaths occurred outside of a hospital. Stroke is the third leading cause of death, behind heart disease and cancer. Most recovery occurs, during the first few months following a stroke. According to the National Institute of Health, the risk of stroke is greater and the recovery process is slower. Thrombo embolic strokes are caused by fatty deposits (plaques) that have built up in the arteries carrying blood to the brain. This slows blood flow and can cause clots to form on the plaques that narrow or block the flow of oxygen and nutrients to the brain. It is also, caused by a blood clot formed in another part of the body that breaks loose, travels through the bloodstream and blocks an artery carrying oxygen and nutrients to the brain. When travelling through the body the blood clot is called an embolus (Mohr, 2001). A hemorrhagic stroke is caused when an artery supplying blood bleeds into the brain. The broken blood vessel prevents needed oxygen and nutrients from reaching brain cells. One type of hemorrhagic stroke is caused when an artery that has weakened over time bulges (called an aneurysm) and suddenly bursts (http://www.americanheart.org/downloadable/heart/1200082005246HS_Stats2008.final.pdf). Thrombo-embolic Stroke can be classified as Transient Ischemic Attacks (TIA), Evolving Stroke, Completed Stroke, Residual Squeal, Classical Stroke, Inappropriate Stroke, Anterior Cerebral Territory Stroke, Posterior Cerebral Stroke, Middle Cerebral Territory Stroke. Hemorrhagic Stroke can be classified as Cerebellar stroke, Thalamic Stroke and Cortical Stroke.

The main goal of this study, is to establish either the category of a stroke disease, which is defined by some

attributes or clinical variables. Nevertheless, not all those attributes give the same quality and quantity of information when the classification is performed. Sometimes, too much information can cause deteriorating the performance of the classification. The problem of variable selection involves choosing a subgroup of variables from an overall set of them that might carry out the finest classification. The advantages of selection process are cost reduction for data acquisition, Increased efficiency of classifier system, improved understanding of classification model, Efficacy improvement. (Sugumaran and Vijayan, 2008). In this study, we have proposed a new novel hybrid neuro-genetic approach, in which GA is used for the selection of neural network architecture connection weights in the prediction of Stroke disease. The rest of the study, discusses about the related studies, the proposed model, results and discussion along with conclusion.

Related studies: Montana and Davis (1989) reported the successful application of a GA to a relatively large neural network problem. They demonstrated that GA produced results superior than backpropagation. Yao and Liu (1997) had presented a new evolutionary system, i.e., EPNet to evolve ANN architecture and connection weights simultaneously. Wieland (1991) described the idea of GA based adaptive approach to a typical control problem. Sexton *et al.* (1998a) employed GA first to search the weight vector of ANNs. They compared backpropagation with GA and resulted each GA derived solution was superior to the corresponding backpropagation solution. GA can potentially be used to optimize several factors of the process of ANNs including feature subset selection, network structure optimization, learning parameter optimization. There is great potential for further research with simultaneous optimization method using GA for other AI techniques including case-based reasoning and decision tree (Kyoung-jae *et al.*, 1999). The Combined Adaptive Resonance Theory (ART) Neural Network and Genetic Algorithm have shown the improved accuracy and objectivity of breast cancer disease diagnosis (Punitha *et al.*, 2007). Another hybrid approach, is to use both GAs and a back-propagation neural network to solve the same optimization problem. Since, GAs have a better chance of getting to the global optimum and ANNs are faster, we can combine the best of both worlds by first getting close to the global optimum using the GA and then using the ANN to improve our result (Janson and Frenzel, 1993). Sexton *et al.* (1998b) also, used tabu search to optimize the network, tabu search derived solutions were significantly superior to those of backpropagation solutions for all test data in the resulting comparison.

Sexton *et al.* (1999) again incorporated simulated annealing, one of the global search algorithms, to optimize the network. They compared with the solution derived by GA and simulated annealing and concluded solution with GA outperformed that with simulated annealing. Shanthi *et al.* (2008), in the prediction of Stroke disease using hybrid GA-ANN, found that the average prediction accuracy in GA-NN for input feature selection is 98.67%. It has been evident from the study that the optimized weights can overcome the problems of Gradient decent technique used in Back propagation algorithm.

MATERIALS AND METHODS

Most research on the application of ANN used gradient descent algorithm to minimize the error. Gradient descent algorithm is a local search algorithm and may tend to fall local minimum. Sexton *et al.* (1998b) indicated that the use of momentum restarting training at many random points to minimize the difference between actual and desired output. They also, suggested global search algorithm can be used to search weight vector instead of local search algorithms. A variety of computational models based on evolutionary process have been proposed and the most popular models are known as GAs.

A new novel hybrid neuro-genetic algorithm: For training the ANN, the backpropagation algorithm and sigmoid function are used in this model and weights are optimized by GA instead of gradient descent algorithm. Back propagation learning algorithm is a widely used algorithm but it has a drawback of converging to a set of sub-optimal weights, from which it cannot escape. The GA offers an efficient search method for complex problem space and can be used as powerful optimization tool. The new neuro-genetic hybrid algorithm is shown in Fig. 2.

The various steps of the algorithm are:

- Determine the symptoms with the help of expert and medical knowledge
- Design suitable neural network
- Initialize the populations (connection weights and thresholds)
- Assign input and output values to ANN
- Compute hidden layer values
- Compute output values
- Compute fitness using

$$\text{Error}(E) = \frac{1}{N} \sum_{t=1}^N (d(t) - y(p,t))^2$$

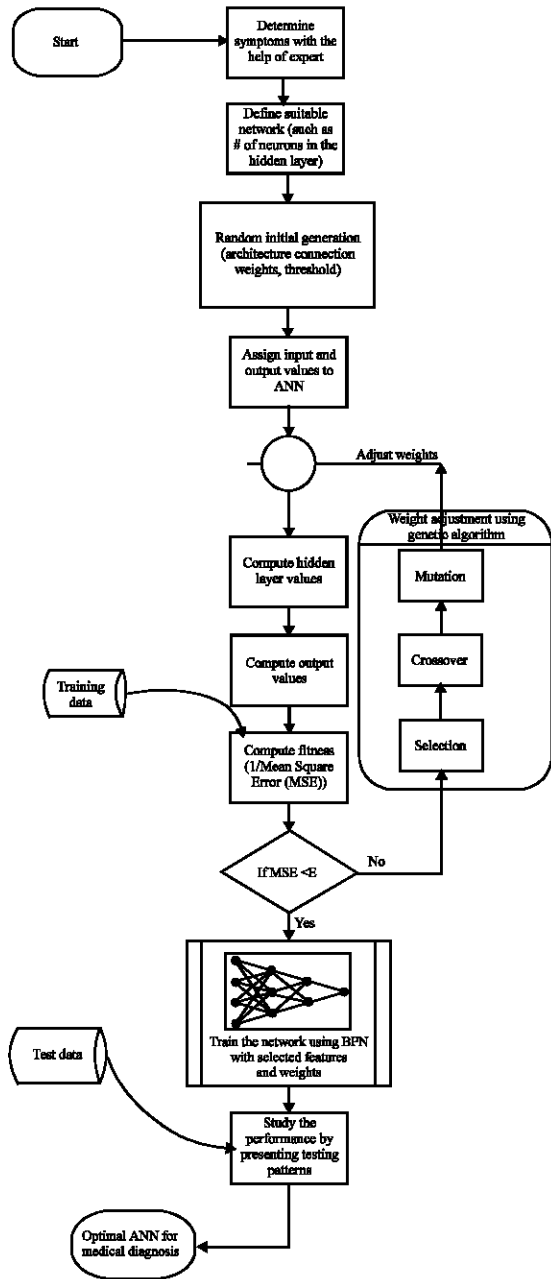


Fig. 2: Neuro-genetic hybrid algorithm

- If error is acceptable, go to step 11
- Select parents of the next generation and apply genetic operator (crossover and mutation)
- Go to step 5
- Train the neural network with selected connection weights
- Study the performance with the test data

The construction and generic operation of the genetic algorithm for feed forward neural networks is as follows:

Given a neural network that is ready to be trained, the initial population of individuals (chromosomes) is generated, each of which codifies a set of values for the weights of the connections and biases of the neural network. Then the fitness of each individual is evaluated, which entails allocating the values of the weights and biases codified by the individual to assess the network connections and calculating the mean square error. The set of training patterns are used to do this calculation. The error value is the individual's fitness. After this initialisation stage, the genetic reproduction, crossover and mutation operators are usually applied to output new generations until there is convergence toward a population of individuals that encode the set of weights that minimise the network error.

Sample geneo type for weight initialization is:

{w11, w12 ... w1n, w21, w22.....w2n, wn1, wn2.....wnh}

Weights from input to hidden layer is

{w11, w12,.....wnh}

Weights from hidden to output layer is

{w11, w12, ... whm}

The chromosome 'x' of the connection GA loop is

{w11, w12, w13.....wnh, w11, w12, w13....whm, b1, b2}

where, b 1 is bias 1 and b 2 is bias 2.

RESULTS AND DISCUSSION

The data for this study have been collected from 150 patients who have symptoms of stroke disease. The data have been standardized with the help of the clinical experts so as to be error free in nature. Table 1 shows the various input parameters for the prediction of stroke disease.

Feature selection: Data are analyzed using NeuCom tool (www.theneucom.com) and the Neuro Intelligence tool (www.alyuda.com) to define column parameters and data anomalies. Data analysis information needed for correct data preprocessing. After data analysis, the values have been identified as missing, wrong type values or outliers and which, columns were rejected as unconvertible for use with the neural network (Baxt, 1995). Feature selection methods are used to identify input columns that are not useful and do not contribute significantly to the

Table 1: Input parameters for the prediction of stroke disease

Parameters
Age
Sex
Pre-stroke mobility
Hypertension
Diabetes mellitus
Myocardial infarction
Cardiac failure
Atrial fibrillation
Smoking
High blood cholesterol
Alcohol abuse
Weakness of left arm and left leg
Weakness of right arm and right leg
Slurring of speech
Giddiness
Headache
Vomiting
Memory deficits
Swallowing difficulties
Loss of vision
Isolated vertigo
Transient double vision
Sudden difficulty in walking, dizziness or loss of balance
Hand/leg numbness
Transient loss of consciousness

Table 2: Percentage of importance of input data

Input column name	Code	Importance (%)
Age	×1	1.2285
Gender	×2	2.6922
Hypertensive	×3	1.626103
Diabetes	×4	4.229285
Myocardial	×5	0.043249
Cardiac failure	×6	0.001659
Atrial fibrillation	×7	0.034991
Smoking	×8	2.061142
Blood cholesterol	×9	8.831646
Left Arm and Leg	×10	19.209636
Right Arm and Leg	×11	2.832501
Slurring	×12	1.776497
Giddiness	×13	3.64891
Headache	×14	15.646755
Vomiting	×15	0.535259
Memory deficits	×16	1.485701
Swallowing	×17	5.224413
Vision	×18	7.366008
Double vision	×19	1.13867
Vertigo	×20	15.471004
Numbness	×21	0.136173
Dizziness	×22	8.700397

performance of neural network. In this study, Signal to Noise Ratio method is used for input feature selection. Variables that have higher values for an output class versus other classes are ranked higher as they have a higher SNR. This function, gives a quantitative measure of how much each variable in a given data set for classification discriminates one class (considered as the signa) from the other class (classes) (considered as noise). The removal of insignificant inputs will improve the generalization performance of a neural network. This method begins with all inputs and it works by removing one input at each step. At each step, the algorithm finds an input that least deteriorates the network performance and becomes the candidate for removal from the input set. Table 2 shows the finalized input parameters after applying feature selection method.

Neural-network architecture: The architecture of the neural network used in this study is the multilayered feed-forward network architecture with 22 input nodes, 11 hidden nodes and 9 output nodes. The number of input nodes are determined by the finalized data; the number of hidden nodes are determined through trial and error and the number of output nodes are represented as a range showing the disease classification. The most widely used neural-network learning method is the BP algorithm (Rumelhart *et al.*, 1986). Learning in a neural network involves modifying the weights and biases of the network in order to minimize a cost function. The cost function always includes an error term a measure of how close the network's predictions are to the class labels for the

examples in the training set. Additionally, it may include a complexity term that reacts a prior distribution over the values that the parameters can take.

Evolving weights using GA: Weight values can be binary or real coded. Some studies used a binary representation of connection weights. Binary Connection weight can eliminate irrelevant values but it may cause significance loss. ANN is initialized with random weights with each weigh being between -1.0 to +1.0 using GA. Initial population of 50 vectors chosen randomly. This study requires 2 sets of vectors of weights, first vector set consists of connection weights between input to hidden layer and 2nd vector set for connection weights between hidden layer to output layer. This study uses 25 input features and employs 22 input elements 11 hidden elements. The weights in the ANN are encoded in such a way each weight is being between -1.0 to +1.0. After that weights are assigned to each link. To calculate the fitness of each chromosome the network is trained with the stroke data and root mean square was returned. A small value, closes to zero, indicates that the network has learned well and is suited for the classification problem. In this method GA searches among several set of weight vectors simultaneously. The training is done by GA search instead of Gradient descent search method. In this application, each string or chromosome in the population represents the weight and bias values of the network. The initial population is randomly generated. By selecting suitable parameters, like selection criteria, probability of cross-over, probability of mutation, initial population, etc., to the GA, high efficiency and performance can be achieved. The objective function is minimization of the

Table 3: ANN-GA parameters used

Network parameters	Description
Neural network architecture	22-11-9
Transfer function	Sigmoid
Training data set	69.33% (104)
Validation data set	16.66% (25)
Test data set	14% (21)
Search method	Genetic algorithm
Population size	50
Fitness function	MSE
Number of generations	20
Probability of crossover	0.7
Probability of mutation	0.033
Random number seed	1

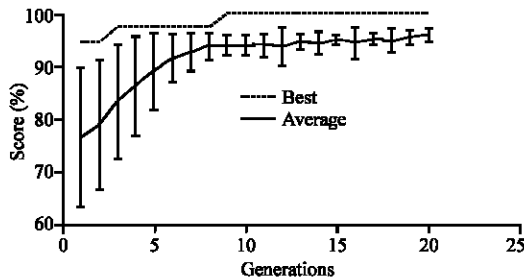


Fig. 3: Results of evaluating weights after 20 generations

Mean Squared Error (MSE). The fitness function considered is the minimum MSE and computed by recalling the network. After getting the fitness values of all chromosomes, they are ranked based on the best fitness values. For the production of offspring for next generation, half of the best ranked population is selected. This half population undergo cross over with cross-over probability. This again will be mutated to give a new offspring, with mutation probability, which is combined with selected best population to form a new population for the next generation. Table 3 shows the ANN-GA parameters used in this study.

Basically, the performance of weights evolution using GA depended on the number of populations and generations. If these parameters were set too low, the evolution may converge to immature solution. However, the larger number of populations and generations would require longer computation time for convergence. In this study, the number of individuals in the population was chosen 50 and number of generations are used to evolve the solution is 20. In this study, the Generational GA evaluates the fitness of the current population before creating an entire new population and evaluating the fitness of that population. The probability of crossover on average is chosen 0.7 and the mutation rate is chosen 0.033 and Roulette wheel selection method used for doing crossover operation.

Figure 3 represents the fitness score after each generation. A magenta line shows the best fitness that has been attained by each generation. Alongside this is

Table 4: Sample weight vectors

SWV						
Set-1	Set-2	Set-3	Set-4	Set-5	Set-6	
0.3816	-0.3033	0.1298	0.1725	-0.7764	0.4256	
-0.0591	0.6659	0.5261	0.8349	-0.9077	0.8083	
-0.0326	-0.7327	0.4702	0.7125	-0.8786	-0.063	
0.4203	-0.5317	0.5758	0.3268	-0.9627	-0.698	
-0.4009	0.0735	0.5893	-0.996	0.3196	-0.036	
-0.9786	0.0471	0.3227	-0.737	-0.5932	0.0135	
-0.4334	-0.9952	-0.925	-0.631	0.7904	-0.65	
0.1006	0.4397	0.9189	0.6676	-0.1668	0.5756	
0.7421	0.9536	0.4796	0.2138	0.7163	-0.558	
-0.3364	-0.7356	0.5941	0.3145	0.9969	0.9069	
0.3443	-0.2443	0.6302	-0.4	-0.5186	0.7417	
0.7752	-0.1872	0.3302	-0.824	-0.7449	0.2115	
-0.7964	0.9182	0.2796	-0.434	-0.4289	-0.139	
-0.3522	-0.5591	-0.435	-0.33	0.5338	-0.301	
-0.0228	-0.8697	0.8277	-0.434	-0.013	-0.406	
0.868	-0.6914	0.9187	-0.576	-0.1961	-0.614	
0.625	-0.1421	-0.632	-0.92	0.1172	0.5482	
0.0598	0.7052	0.8747	-0.012	-0.852	0.7504	

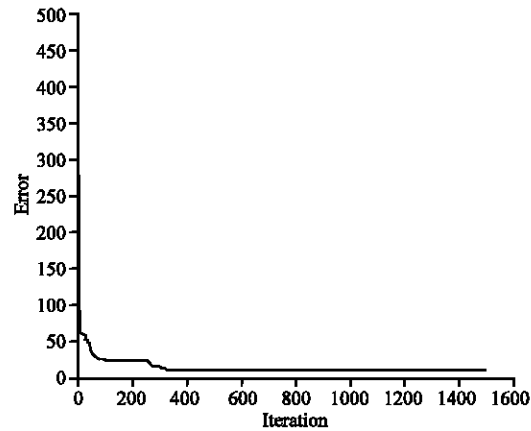


Fig. 4: Error rate during training

a blue line, which shows the average fitness of the population as well as the standard deviation of the scores in the population. This gives an impression of the degree, to which the population has converged on a single solution. Table 4 shows sample finalized weights given to the input-hidden-output connections of the neural network.

The ANN is trained with the dataset using proposed neuro-genetic algorithm. The output values precision specified for this study is 0.01. The process is terminated after 1500 iterations. Figure 4 shows the error rate after each epoch during training process. The RMSE is 0.79551 and Non Dimensional Error index 0.32353.

Table 5, in the output classification various categories of Stroke Disease.

Table 6 shows mean and standard deviation of target and actual output together with Absolute and Relative Error.

Table 5: Output classification

Class	Total	(%)
Left hemiplegia	45	30.00
TIA	25	16.70
Right hemiplegia	20	13.30
Dysphasia	19	12.70
Monoplegia	14	9.33
Aphasia	5	3.33
Right hemianesthesia	6	4.00
Quadruplegia	4	2.67
Left hemianopia	11	7.33
Dysphagia	0	0.00
Mis-classified	1	0.67
Total	150	100.00

Table 6: Statistical results

Des. Stat	Target	Output	Absolute error	Absolute relative error
Mean	3.163462	3.163693	0.005443	0.003101
SD	2.353845	2.347057	0.007417	0.004658
Min	1	1.002383	0.000079	0.000011
Max	9	8.971411	0.030639	0.019476

	45	0	0	0	0	0	0	0	0
45	45	0	0	0	0	0	0	0	0
0	0	25	0	0	0	0	0	0	0
0	0	0	20	0	0	0	0	0	0
0	0	0	0	19	0	0	0	0	0
0	0	0	0	0	14	0	0	0	0
0	0	0	0	0	0	5	0	0	0
0	0	0	0	0	0	0	6	0	0
0	0	0	0	0	0	0	0	4	0
1	1	0	0	0	0	0	0	0	11

Fig. 5: Confusion matrix

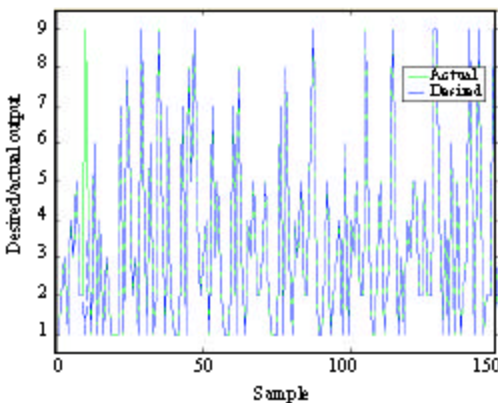


Fig. 6: Graph showing actual vs. predicted output

The following confusion matrix (Fig. 5) shows the results after training and testing the classification model and the overall accuracy is 99%.

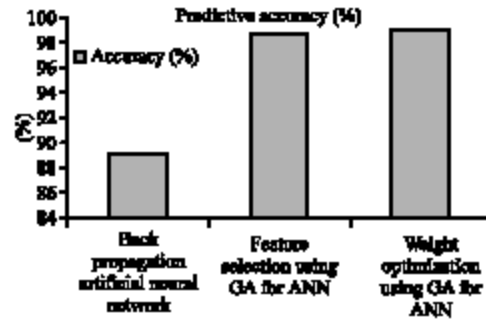


Fig. 7: Accuracy of various models

Figure 6 shown the actual output vs. the predicted output for a given set of samples.

ANN's slow convergence and stuck in the local minima.

Figure 7 shows the accuracy of various architectures and the results reveals that Weight optimization using GA for ANN has got higher accuracy than the monolithic BP-ANN approach.

The results clearly shows our new hybrid Neuro-Genetic approach provides better accuracy than traditional ANN and Feature Selection using GA and this methods prevent the ANN stuck in the local minima.

CONCLUSION

In this study, we have chosen the chromosome with the minimum error from the last generation of the GA algorithm as the initial weights of ANN to make ANN training with the stroke disease data. The real output and desired output of ANN and Hybrid ANN-GA were compared. The classification accuracy for all surfaces were improved. It is concluded that applying GA algorithm to initialize the weights of ANN can take its advantage of optimization and overcome the shortcomings of the ANN's slow convergence and stuck in the local minima.

REFERENCES

Adeli, H. and S. Hung 1995. New York, John Wiley and Sons Inc, pp: 211. ISBN: 0-471-01633-0.

Baxt, W.G., 1995. Application of artificial neural networks to clinical medicine. Lancet, 346 (8983): 1135-1138.

Chen and Wah Sit, 2005. Application of artificial neural network-genetic algorithm in inferential estimation and control of a distillation column. Masters thesis, Universiti Teknologi Malaysia.

- Davis, L., 1991. Handbook of Genetic Algorithms. New York: Van Nostrand, Reinhold.
- Goldberg, D.E., 1989. Genetic Algorithms in Search Optimization and Machine Learning, Addison-Wesley Publishing Company, Inc., pp: 372. ISBN: 0201157675.
- Janson, D.J. and J.F. Frenzel, 1993. Training product unit neural networks with genetic algorithms. *IEEE Expert*, 8 (5): 26-33.
- Kumanan, V., S.K. Nanne Saheb and C.P. Jesuthanam, 2006. Prediction of machining forces using neural networks trained by a genetic algorithm *IE(I) PR J.*, Vol. 87.
- Kyoung-jae, Han, Kim and Ingoo, 1999. Simultaneous optimization method of feature transformation and weighting for artificial neural networks using genetic algorithm: Kpplication to Korean stock market/Kim, Kyoung-jae; Han, Ingoo. *Proc. Korea Inteligent Inform. Syst. Soc. Conf.*, 11 (2): 323-335(13).
- Mohr, J.P., 2001. *Stroke Analysis*. 4th Edn. Oxford Press, New York.
- Montana, D. and L. Davis, 1989. Training feedforward neural networks using genetic algorithms. *Proc. 11th Int. Conf. Artificial Intelligence*, Morgan Kaufmann, pp: 762-767.
- Punitha, A., C.P. Sumathi and T. Santhanam, 2007. A combination of genetic algorithm and ART neural network for breast cancer diagnosis. *Asian J. Inform. Technol.*, 6 (1): 112-117.
- Rumelhart, D.E., J.L. McClelland and R.J. Williams, 1986. Learning representations by back-propagating errors. *Nature*, 323: 533-536.
- Shanthi, D., G. Sahoo and N. Saravanan, 2008. Input feature selection using hybrid neuro-genetic approach in the diagnosis of stroke disease. *Int. J. Comput. Sci. Network Security*, 8 (12): 99-107.
- Sexton, R.S., R.E. Dorsey and J.D. Johnson, 1998a. Toward global optimization of neural networks: A comparison of the genetic algorithm and backpropagation. *Decision Support Syst.*, 22 (2): 171-185. DOI: 10.1016/S0167-9236(97)00040-7.
- Sexton, R.S., B. Alidaee, R.E. Dorsey and J.D. Johnson, 1998b. Global optimization for artificial neural networks: A tabu search application. *Eur. J. Operational Res.*, 106: 570-584. DOI: 10.1016/S0377-2217(97)00292-0.
- Sexton, R.S., R.E. Dorsey and J.D. Johnson, 1999. Optimization of neural networks: A comparative analysis of the genetic algorithm and simulated annealing. *Eur. J. Operational Res.*, 114: 589-601. DOI: 10.1016/S0377-2217(98)00114-3.
- Sugumaran and Vijayan, 2008. *Several Approaches to Variable Selection by Means of Genetic Algorithms. Intelligent Information Technologies: Concepts, Methodologies, Tools and Applications*. IGI Publishing.
- Wang and John, 2006. *Encyclopedia of data warehousing and mining*. Vol. 2, I-Z, ISBN: 9781591405573.
- Wieland, 1991. Alexis Wieland. Evolving neural network controllers for unstable systems. In: *Proceedings of the International Joint Conf. Neural Networks (Seattle, WA)*, 2: 667-673.
- Yao, X. and Y. Liu, 1997. A new evolutionary system for evolving artificial neural networks. *IEEE. Trans. Neural Networks*, 8 (3): 694-713.