Breast Cancer Detection Using Cartesian Genetic Programming evolved Artificial Neural Networks

Arbab Masood Ahmad, Gul Muhammad Khan, S.Ali Mahmud
Electrical Engineering Department
UET Peshawar, Pakistan
arbabmasood,gk502,sahibzada.mahmud@nwfpuet.edu.pk

Julian F. Miller
Intelligent System Design Group
Electronics Department
University of York, UK
julian.miller@york.ac.uk

ABSTRACT

A fast learning neuro-evolutionary technique that evolves Artificial Neural Networks using Cartesian Genetic Programming (CGPANN) is used to detect the presence of breast cancer. Features from breast mass are extracted using fine needle aspiration (FNA) and are applied to the CGPANN for diagnosis of breast cancer. FNA data is obtained from the Wisconsin Diagnostic Breast Cancer website and is used for training and testing the network. The developed system produces fast and accurate results when compared to contemporary work done in the field. The error of the model comes out to be as low as 1% for Type-I (classifying benign sample falsely as malignant) and 0.5% for Type- II (classifying malignant sample falsely as benign).

Categories and Subject Descriptors

I.2.2 [**ARTIFICIAL INTELLIGENCE**]: Automatic Programming—*Program synthesis*; I.2.6 [**ARTIFICIAL INTELLIGENCE**]: Learning—*Connectionism and neural nets*

General Terms

Algorithms, Design, Performance

Keywords

Breast Cancer; Fine Needle Aspiration FNA; Cartesian Genetic Programming; Artificial Neural Network; Neuro-evolution.

1. INTRODUCTION

Breast cancer is one of the leading causes of death in women. Detection of the disease at an earlier stage can save precious lives. Various diagnostic tests and procedures are available for detecting the presence of the disease. One of these is analysis of a biopsy taken from the breast. This is quite painful and causes discomfort to the patient but is more reliable than other diagnostic techniques. Biopsies are taken to ascertain whether a tumor is benign or malignant. Due to the discomfort of biopsies, patients often hesitate to visit a doctor until it is too late. The easiest of these tests is the mammography, in which the radiologist examines the x-rays of the breast for a possible mass. However, the accuracy of this test largely depends on the expertise of the radiologist and consequently there is a chance that a malignant lesion is diagnosed as benign or vice versa.

There are good data sets obtained via Fine Needle Aspiration (FNA) biopsy and many computational intelligence techniques have used in classification of this data [37, 10]. Such methods include multilayer perceptrons [2], radial basis function (RBF) networks [28], fuzzy classifiers [3, 25], clustering algorithms [20], evolutionary computation [8], principal component analysis [25], and different kernel-based methods [32].

The work we present in this paper is based on classification of the data available at the WDBC web site[1]. This is a processed form of the FNA data[37]. Cartesian Genetic Programming evolved Artificial Neural Networks (CGPANNs)[15] are used to classify the data. The system takes as input, the FNA data and classifies the case as either benign or malignant. Before the network can be applied for diagnosis it must be trained first. The training process consists of applying to the network, a subset of the data from Wisconsin Diagnostic Breast Cancer WDBC web site, that includes the FNA feature parameters and the corresponding classification results (either Benign or Malignant). Once trained, the network is tested on unseen FNA data set(not used in training) to decide whether a breast mass is benign or malignant.

2. PREVIOUS WORK

Different methods have been applied to the diagnosis of breast cancer. In [6] Genetic Programming (GP) was applied to extract features using the Fischer criterion. The extracted features were then given to an ANN for classification.

Fisher Linear Discriminant Analysis (FLDA) and principal component analysis (PCA) have been considered to be the best feature extraction methods [7]. In another approach, discussed in [33] preprocessed data from the Wisconsin Diagnostic Breast Cancer (WDBC) is directly fed as terminal values to a genetic programming algorithm. The output of the algorithm is compared with the required output and the fitness computed. A population of individuals is generated and the best individuals are selected, crossed

[1] Breast Cancer Wisconsin (Diagnostic) Data Set http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

GECCO'12, July 7–11, 2012, Philadelphia, Pennsylvania, USA.

over and mutated. Following a number of generations, the population converges to the solution that best represents the discrimination function.

In [4] the authors discuss the Artificial Bee Colony (ABC) algorithm for optimization of artificial Neural Network (ANN). The ABC not only avoids the problem of being trapped in the local minima as faced by other similar methods but also optimizes the network so that its performance is improved in addition to a reduced architecture. The synaptic weights, architecture and transfer function are evolved to obtain minimum classification error (CER) and minimum Mean Square Error (MSE). The algorithm is evaluated on the Breast cancer taken from UCI repository producing competitive results. In [19] the authors implemented a Particle Swarm Optimization (PSO) algorithm for evolving the synaptic weights, the topology and the transfer functions of each neuron of an ANN. They tested the technique on a number of nonlinear problems including breast cancer.

In [24] a limited database of mammograms was used for shape features of the breast masses that were classified using Genetic Programming as either benign or malignant. To refine features applied to the GP classifier, feature-selection methods including Student's t-test, Kolmogorov-Smirnov test, and Kullback-Leibler divergence were used. Land Jr. et al. [18] used a modified form of Fogel's Evolutionary programming to evolve ANNs for breast cancer detection. Using the FNA dataset [37] they were able to achieve an average success of 97.26%. Tingting Mu et al. discussed the application of support vector machines (SVMs), radial basis function (RBF) networks, and self-organizing maps (SOMs) for breast cancer detection [23]. In a SOM-RBF classifier the RBF network processes the clustering result obtained by the SOM. This provided excellent experimental results giving a detection accuracy of up to 98%.

Werner and Fogarty addressed the problem of how to obtain a mathematical discriminate function for quantifying the severity of a disease using genetic programming (GP) [33]. They managed to obtain a detection accuracy of 96.32%. They compared their results of GP with those of West and West [34] who examined many techniques: MLP, General Regression (GR), Radial Basis Function (RBF), Mixture of Experts (MOE), LDA, logistic regression, K search neighbor and Kernel for diagnosis of breast cancer. Werner and Fogarty found that the techniques examined by West and West have more false negatives than GP.

Abbass discussed an evolutionary artificial neural network (EANN) approach based on the pareto-differential evolution (PDE) algorithm with local search for breast cancer detection [1]. The approach is called Memetic Pareto Artificial Neural Network (MPANN). Abbass managed to attain an accuracy of 98.1% using MPANN. Iranpour et al. discussed the benefits of applying support vector machines (SVMs) and radial basis function (RBF) for breast cancer detection [10]. They obtained an accuracy of 98.1% which compared favourably to accuracies (in parentheses) obtained in other studies: linear SVM classifier (94%)[6], fuzzy classifiers (95.8%) [9], and edited nearest-neighbor (ENN) with pure filtering (95.6%) [17].

Janghel et al. compared various neural network models applied to the breast cancer diagnosis [11]. They implemented six models of neural networks namely Back Propagation, Radial Basis Function Networks, Learning Vector Quantization, Probabilistic Neural Networks, Recurrent Neural Network and Competitive Neural network.

Table 1 shows the results of all these previous methods along with the best results obtained through CGPANN, clearly demonstrating the competitiveness of CGPANN. We will explain the CGPANN technique and how these results were obtained in the remainder of the paper.

3. NEUROEVOLUTION

Obtaining a effective model of Artificial Neural Network (ANN) in terms of its architecture for a specific problem has always been a difficult task and was often chosen by researchers based on experience. However for over a decade now, researchers have been introducing techniques that determine this automatically. This technique evolves both topology and weight of ANN and is called a TWEANN (Topology Weight Evolved Artificial Neural Network) [38]. The general term used for artificial evolution of ANNs is known as Neuro Evolution (NE). Many aspects, such as weights, functions, inputs and ANN topology are evolved in Neuro-Evolution [38]. The genetic representation of the ANN is called genotype while the ANN itself is often referred to as the phenotype. The encoded attributes in the genotype greatly affect the solution search space [38]. Evolving only some specific system parameters for instance weights and keeping constant another attribute like the topology of the network can lead to a situation in which the solution space is restricted. In such situations evolution operates in a more constrained space and hence truly novel solutions can be missed [38]. A variety of neuro-evolutionary techniques have been explored and we give a short review here. In Symbiotic, Adaptive Neural Evolution (SANE), the neuron population is evolved alongside the network topology [22]. In Enforced Sub-Population (ESP), an extension of SANE, a subpopulation of the hidden layer neurons is evolved rather than one population [5]. Daniel Polani et al. introduced another extension of SANE called EuSANE which used a reinforced learning algorithm (Eugenic Algorithm) [15]. In conventional neuro-evolution (CNE) the genotypic representation of the entire network is made. The advantage associated with it is that it evolves the chromosomes at the network level rather than at the neural level and hence potential global solutions can be found for a predefined network's size and topology [5]. Stanley proposed a technique called NEAT which stands for Neuro Evolution of Augmenting Topologies. In this technique he proposed three innnovations: the tracking of genes with historical markings that allow easier recombination of network topologies, protecting innovation through speciation and starting from a simplified form and evolving into a more complex structure. NEAT shows better performance than many previously introduced NE algorithms [31]. Stanley et al. also introduced real time NEAT (rtNEAT) to evolve complex neural networks in real time [30]. Reisinger et al. modified NEAT and introduced modular NEAT in order to divide the problem of large search space of a complex ANN into sub parts [27]. Whiteson and Stone adapted NEAT to build value functions for temporal difference reinforcement learning. They found the technique was able to evolve individuals with better learning ability[35]. They argued that evolutionary computation can be used in reinforcement learning to improve the performance of the computation in real-time (rather than off-line). Schmidhüber et al. presented EVOLINO that is Evolution of

No.	Method	Mean (MAPE)	References
1	MLP	95.56	[6]
2	SVM	93.95	-
3	FLDA/MLP	90.92	-
4	PCA/MLP	92.02	-
5	GP/MDC	96.58	-
6	SOM-RBF	98.00	[23]
7	MLP	95.72	[33]
8	GR	96.76	-
9	RBF	97.04	-
10	MOE	96.29	-
11	LDA	96.34	-
12	Logistic	97.22	-
13	K neighbour	96.78	-
14	Kernel	95.02	-
15	GP - Test average	96.32	-
16	L2-SVM/GDVEE (RBF)	98.1	[10]
17	SVM (linear)	94.00	-
18	SVM (RBF)	97.70	-
19	Fuzzy	95.80	-
20	ENN	95.60	-
21	CGPANN	**97.00 for Type-I and 98.50 for Type-II**	Proposed Solution

Table 1: Comparison of Mean Absolute Percentage Errors (MAPE) obtained using various classification methods for FNA data.

recurrent systems with linear outputs for sequence learning [29]. EVOLINO combines neuro-evolution with analytical linear methods (e.g. linear regression or quadratic programming). The idea behind this technique is that in many cases a linear models can account for a large number of properties of a problem. Evolution is then used to deal with properties that require non-linearity and recurrence.

Pujol and Poli observed that forms of GP that use graph representations were well suited to representing and evolving artificial neural networks [26]. They showed they could obtain good results on some simple problems (XOR, binary addition and simple letter recognition).

Yao reviews a number of neuro-evolution that evolve the incidence matrix of a neural network [38]. For a N-node neural network, this is an NxN matrix, C_{ij} where the matrix entries c_{ij} represent the weight of a connection between node i and node j.

4. CARTESIAN GENETIC PROGRAMMING EVOLVED ARTIFICIAL NEURAL NETWORKS (CGPANN)

CGPANN is based on Cartesian genetic programming [21]. It represents nodes and connections in the form of a two dimensional graph rather than a tree [16]. It has a number of parameters including: number of nodes, maximum number of inputs per node (arity), number of columns, number of rows and levels-back. Arranging these parameters in various ways can create many different types of graphs. In many publications the genetic representation has been restricted to ensure that the graphs are directed and acyclic. In CGP, a genotype is made up of fixed length array of integers representing various genes (i.e. node inputs (or connections), functions and output genes). An important aspect of CGP is that when genotypes are decoded into phenotypes not all nodes are connected in the path from inputs to outputs. Nodes that are not connected are referred to as non-coding nodes. Through mutation the status of coding and non-coding genes is randomly changed, which can result in offspring that are very different from the parent. When nodes of CGP are replaced by artificial neurons with non-linear activation functions and weighted connections it is transformed to ANN and we refer to it as a CGPANN (Cartesian genetic programming ANN). Evolving CGPANNs allow topology, weights and neural functions to change. Feedforward and recurrent (FCGPANN , RCGPANN) CGP architectures have been investigated on a number of problems [14, 13, 12].

Assume that a CGPANNs genotype has a user defined number of nodes, m and the arity of a node is a. The node genes N_i are: $F, I_1, W_1, I_2, W_2, I_3, W_3 \cdots I_a, W_a$. The genotype $G(m)$ is given by

$G(m) = N_1, N_2, \cdots, N_m, O_1, O_2, \cdots, O_p$

where

F takes values of 0 or 1 indicating which activation function is used (either sigmoid or hyperbolic tangent)

I_i: represents the input to the node. For feedforward networks it can be connected to either a network input or a node in a column preceding the column containing the current node.

W_i: represents the weight associated to I_i. It is a real-valued number in the range -1 to +1

O_i: represent which neurons provide the network outputs.

It is interesting to contrast the CGP representation of an ANN with the graph incidence matrix representation. The length of CGP genotype encoding a neural network is given by $L = (2n_a + 1)m + p$, where n_a is the node arity. Whereas incidence matrices assume any node can be connected to any

other node, so the number of entries is proportional to m^2. Moreover, assuming that the entries in an incidence matrix are the weights of connections has the drawback when evolutionary algorithms are used, since it can be difficult to disconnect neurons when this can only be done by adjusting their weight to exactly zero. CGPANNs thus present a more compact representation and allow larger changes to the network to be made with few mutations.

Figure 1 shows the internal representation of a single node (neuron) with arity $a = 3$. In the network, the nodes are arranged in rows and columns in such a way that a node in a certain column can only connect to a node in a column to the left of that node or to an input. Nodes in the same column cannot be interconnected. Similarly an output can connect to the output of any node or to a network input. The genes in a chromosome are arranged in such a way that we start at the first node located in row_1 and $column_1$, move down the rows until all the nodes in the first column are considered and then to the top of the 2nd column and so on till the output genes are encountered. The groups of nodes that have their outputs ultimately connected and are part of the ultimate phenotype are called the active nodes, while those whose outputs are not used are termed as junk or inactive nodes. A typical phenotype is shown in Figure 2, with node 3,5, 7 and 9 being junk nodes.

4.1 Evolutionary Strategy

We use a 1+ λ(Number of Offspring) evolutionary strategy in which $\lambda = 9$. This, and the algorithm we use for training and testing the networks is outlined in this section.

ALGORITHM 1. ***The*** $(1 + 9)$ ***evolutionary strategy***

1: Generate a training matrix and an output column vector. The elements of each row of the former matrix correspond to the FNA features of a patient, while the elements of the later correspond to the results of the test feature evaluation by the physicians.
2: Enter the number of rows and columns of nodes, the number of inputs per node and the number of outputs for CGPANN.
3: ***for all*** *i such that* $0 \leq i < 9$ ***do***
4: Randomly generate individual i
5: ***end for***
6: Select the fittest individual, which is promoted as the parent
7: ***while*** *a solution is not found* ***or*** *the generation limit is not reached* ***do***
8: ***for all*** *i such that* $0 \leq i < 9$ ***do***
9: Mutate the parent to generate offspring i
10: ***end for***
11: Generate the fittest individual using the following rules:
12: ***if*** *an offspring genotype has a better or equal fitness than the parent* ***then***
13: Offspring genotype is chosen as fittest
14: ***else***
15: The parent chromosome remains the fittest
16: ***end if***
17: ***end while***
end

On line 10 of the procedure there is an extra condition that when offspring genotypes in the population have the same fitness as the parent and there is no offspring that is better than the parent, in that case an *offspring* is chosen as the new parent.

5. APPLICATION OF CGPANN FOR DIAGNOSIS OF BREAST CANCER

In this section we discuss aspects of the Fine Needle Aspiration dataset, the experimental setup, results, analysis and conclusions.

5.1 DIAGNOSTIC PROCEDURES AND DATA SET

The procedure for this test consists of extracting material from the mass from the suspected region of the breast cancer patient using Fine Needle Aspiration (FNA). The mass extracted is examined under microscope for the following nine features: (1) clump thickness, (2 and 3) uniformity of cell size and shape, (4) marginal adhesion, (5) single epithelial cell size, (6) bare nuclei (7) bland chromatin, (8) normal nucleoli and (9) mitoses [37]. Based on these features the physician decides whether a sample is benign (non cancerous) or malignant (cancerous), each of these features has a value between 1 and 10, however no single feature can be made a basis for decision. A dataset of patients based on these parameters was collected by physician W.O. Wolberg, University of Wisconsin Hospitals.

However, for machine learning another dataset is formed by processing the digital images of the well differentiated cells of the masses with a computer, for the following features:(1) Radius, (2) Perimeter, (3) Area, (4) Compactness, (5) Smoothness, (6) Concavity, (7) Concave points, (8) Symmetry, (9) Fractal dimention and (10) Texture [23]. The mean value, worst (mean of the three largest values) and standard error of each feature are computed for each of these parameters and this results in a total of thirty feature values [36]. A patient dataset with these feature value and their results, is available at the WDBC web site. The set consists of 569 patterns, out of which 357 are benign and 212 are malignant. These thirty features form the inputs to our system while there is only one output, that has a value of 0 for benign and 1 for malignant. To obtain this, we round to the nearest integer the real-value output of our network.

5.2 EXPERIMENTAL SETUP

An initial population of ten genotypes is randomly created. Each genotype is transformed into phenotype and the input data from the training set is applied. Then the output of the CGPANN is evaluated for fitness by comparing the output with the result decided by the physician (either benign or malignant). The fitness is the sum of the number of false negatives and false positives for the dataset. The error for each genotype is calculated and the best genotype with least error is promoted to the next generation. The parent genotype and randomly mutated copies of it make the next population. The mutation rate is set at 10% for all the experiments. In our experiments we made all the networks such that the number of rows is set to one. This makes the number of nodes equal to number of columns. The process continues from generation to generation until the error reduces to a pre-set value or the number of generations reach the pre-set value. There are two types of errors, Type-I and Type-II. In Type-I error a benign sample is falsely classified as malignant while in Type-II error a malignant sample

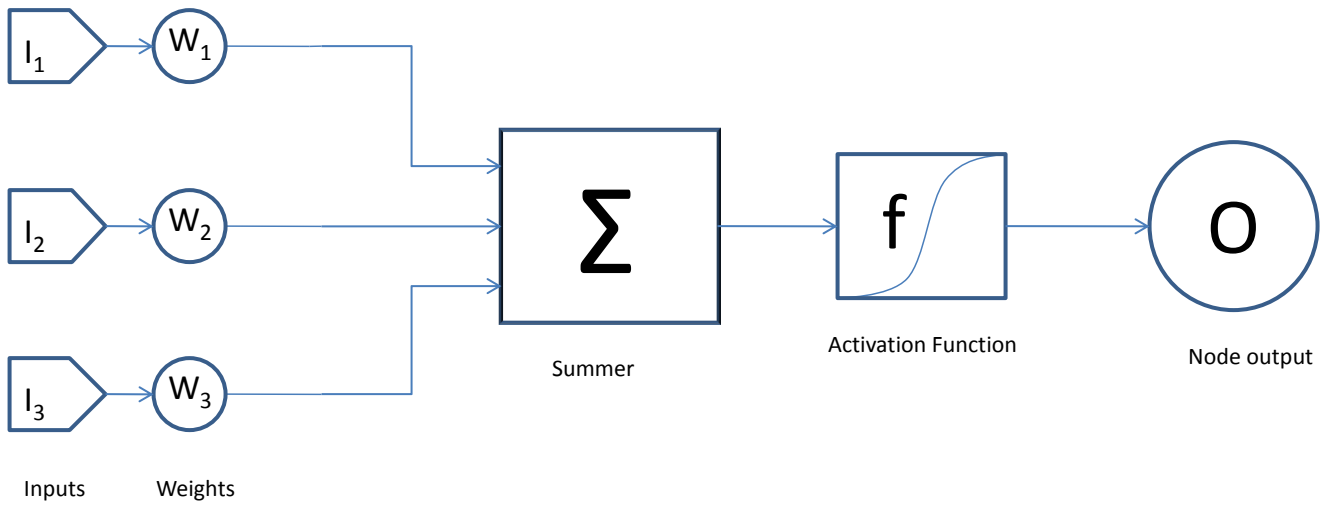

Figure 1: Internal view of a single CGPANN neuron

is falsely classified as benign. The second type is a more catastrophic mistake. The sum of errors resulting from the application of complete training data to a genotype gives the fitness of that genotype in terms of the total error. All the genotypes of the population are evaluated in this way. Once all experimentation is done, we calculate the total error, error for type-I and type-II using the following formulae:

Total Percentage Error= Fitness of the genotype (total number of false predictions)x100/N_t
Type-I Percentage Error= N_{T-I}x100/N_t
Type-II Percentage Error= N_{T-II}x100/N_t

Where,

N_t=Total number of samples diagnosed
N_{T-I}= Number of benign patients diagnosed as malignant
N_{T-II}= Number of malignant patients diagnosed as benign

5.3 RESULTS AND ANALYSIS

In order to evaluate the algorithm,initially, we ran 24 different experiments with different morphologies of the network in terms of the maximum number of nodes and number of inputs per node. The best result obtained so far has 1 (one) error while training with 200 cases i.e. 99.5% successful training. We have run all the experiments for one hundred thousand generations. Once the training is complete, we test the evolved model on another set of 200 test cases and the best results obtained are 4 errors out of 200 cases thus giving 98% accuracy as a whole. Table 2 shows the results obtained for all the cases with numbers of nodes 100, 200 and 300 and the corresponding number of inputs per node varying from 5 to 40 with a step size of 5 respectively. The data for both training and testing is selected randomly from the complete set for all the experiments.

Table 2 demonstrates a promising trend in the results with an average accuracy of 96%. It is encouraging that in these experiments most of the errors are of Type-I (99%, two errors out of two hundred samples) which falsely classifies benign sample to be malignant, while the maximum number of Type-II errors is only 1 (99.5% accurate). This means that there is very little probability that a patient having malignancy would be classified as healthy which would be a serious error in comparison to the case that a healthy patient is classified as having cancer. The later would merely indicate that more tests were required.

6. FURTHER ANALYSIS

In order to evaluate and compare CGPANN with other algorithms we have arranged the experimental setup such that it complies with previously published work. In order to do this, we have arranged the data for ten (10) fold cross validation. In ten fold cross validation each dataset is divided into 10 blocks of approximately equal size. The data is then shuffled to create ten different data sets. At the end the network is trained on the 9 blocks and tested on the tenth (10th) block. After arranging the data, we have selected the best optimal network as highlighted in table 2 with 200 nodes and 35 inputs per node as the base parameters. The nine block data set represent 512 patients data

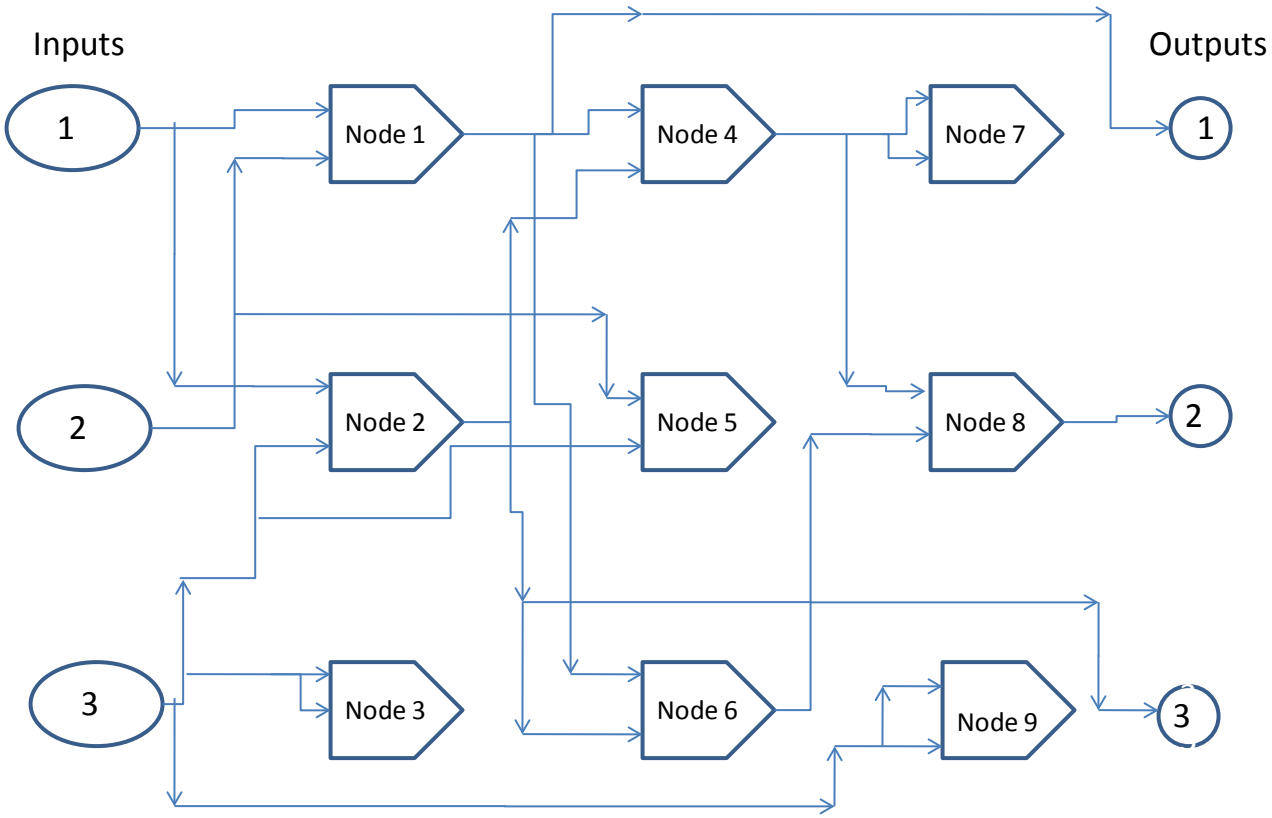

Figure 2: A typical CGPANN phenotype. Nodes 3, 5, and 9 are inactive and the remaining nodes are active.

samples with each sample having thirty feature inputs and one output. We trained the system on all ten shuffled data sets for one hundred thousand evolutionary generations in five independent evolutionary runs. We adjusted various network parameters (weights and topology)in order to obtain the best possible model. Since 512 samples are used for training the remaining 57 samples are used for testing.

Table 3 shows the training and testing results on all the ten data sets.

From the table it is evident that the network does perform well on average and attains 100 % accuracy at times for both type-I and type-II cases. The most encouraging aspect is that type-II error *average* is 98.5 % which shows the strength of algorithm in avoiding the misclassification of malignant cases as benign.

7. FUTURE DIRECTIONS AND CONCLUSION

In this work a fast learning neural network (CGPANN) is explored for producing optimal neural network model for accurate diagnosis of breast cancer. The classification accuracy of the system is better than other contemporary methods and systems developed on the same data set. Also it is encouraging that the method gives lower type II errors (classifying malignant cases as benign) than type I errors (classifying benign cases as malignant). Our approach could be further validated using additional tests by extracting more data from various sources. In future, we will investigate the use of CGPANNs for mammogram image analysis in addition to FNA data classification. This will need extra effort and some preprocessing of the mammogram images before they are applied to the CGPANN classifier. The successful application of the system to breast cancer diagnosis, will in future pave the way for its application to the diagnosis of other diseases such as cardiac arrhythmia and brain disorders.

8. REFERENCES

[1] H. A. Abbass. An evolutionary artificial neural networks approach for breast cancer diagnosis. *Artificial Intelligence in Medicine*, 25:265–281, 2002.

[2] T. C. S. S. André and R. M. Rangayyan. Classification of breast masses in mammograms using neural networks with shape, edge sharpness, and texture features. *J. Electronic Imaging*, 15(1):013019, 2006.

[3] H. D. Cheng and M. Cui. Mass lesion detection with a fuzzy neural network. *Pattern Recognition*, 37(6):1189–1200, 2004.

[4] B. A. Garro, H. Sossa, and R. A. Vázquez. Artificial neural network synthesis by means of artificial bee

Number of Nodes	Number of Inputs Per Node	Training Accuracy (%)	Testing Accuracy (%)	Testing Type-1 Accuracy (%)	Testing Type-2 Accuracy (%)
100	5	98.5	95.5	97.5	98
	10	97	95	97.5	97.5
	15	98.5	95.5	98	97.5
	20	98.5	96	98	98
	25	98.5	94.5	95.5	99
	30	98.5	96	97.5	98.5
	35	99	95	95.5	**99.5**
	40	98.5	94	96.5	97.5
200	5	98	97	98.5	98.5
	10	97.5	95.5	97.5	98
	15	97	92.5	93.5	99
	20	99	96	97.5	98.5
	25	98	95	96	99
	30	97.5	95	96.5	98.5
	35	**99.5**	96.5	97	**99.5**
	40	**99.5**	97	98	99
300	5	98	95	95.5	**99.5**
	10	98	96	97.5	98.5
	15	99	94.5	96.5	98
	20	98.5	95	97	98
	25	**99.5**	**98**	**99**	99
	30	96	95	96	99
	35	**99.5**	95.5	96.5	99
	40	99	95	96	99

Table 2: Table of results for various maximum numbers of nodes, inputs per node (arity) for a single run. The training data set used data for 200 patients and the test data set used data from 200 different patients.

colony (abc) algorithm. In *IEEE Congress on Evolutionary Computation*, pages 331–338, 2011.

[5] F. Gomez, J. Schmidhuber, and R. Miikkulainen. Accelerated neural evolution through cooperatively coevolved synapses. *J. Mach. Learn. Res.*, 9:937–965, 2008.

[6] H. Guo and A. K. Nandi. Breast cancer diagnosis using genetic programming generated feature. *Pattern Recognition*, 39(5):980–987, 2006.

[7] J. Han and N. Cercone. *ule Viz*: a model for visualizing knowledge discovery process. In *KDD*, pages 244–253, 2000.

[8] S. Ho, H. Chen, S. Ho, and T. Chen. Design of accurate classifiers with a compact fuzzy-rule base using an evolutionary scatter partition of feature space. *IEEE Trans. Syst. Man Cybern.*, 34 (2):1031–1044, 2004.

[9] S.-Y. Ho, H.-M. Chen, S.-J. Ho, and T.-K. Chen. Design of accurate classifiers with a compact fuzzy-rule base using an evolutionary scatter partition of feature space. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 34(2):1031–1044, 2004.

[10] M. Iranpour, S. Almassi, and M. Analoui. Breast cancer detection from fna using svm and rbf classifier. In *1st Joint Congress on Fuzzy and Intelligent Systems*, 2007.

[11] R. Janghel, A. Shukla, R. Tiwari, and R. Kala. Intelligent Decision Support System for Breast Cancer. In *Advances in Swarm Intelligence*, volume 6146, chapter 46, pages 351–358. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[12] G. Khan, S. Khan, and F. Ullah. Short-term daily peak load forecasting using fast learning neural network. In *11th International Conference on Intelligent Systems Design and Applications, ISDA'2011*, page in press, 2011.

[13] M. Khan, G. Khan, and J. F. Miller. Evolution of optimal anns for non-linear control problems using cartesian genetic programming. In *Proc. IEEE. ICAI'2010*, 2010.

[14] M. Khan, G. Khan, and J. Miller. Efficient representation of recurrent neural networks for markovian/non-markovian non-linear control problems. In *Proc. ISDA'2010*, pages 615–620, 2010.

[15] M. M. Khan, G. M. Khan, and J. F. Miller. Evolution of neural networks using cartesian genetic programming. In *IEEE Congress on Evolutionary Computation*, pages 1–8, 2010.

[16] J. R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Subprograms*. MIT Press, 1994.

[17] W. Lam, C.-K. Keung, and C. X. Ling. Learning good prototypes for classification using filtering and abstraction of instances. *Pattern Recognition*, 35(7):1491–1506, 2002.

[18] W. H. Land Jr, L. Albertelli, Y. Titkov, P. Kaltsatis, and G. Seburyano. Evolution of neural networks for the detection of breast cancer. In *Proc. IEEE. Int. Joint Symposia on Intelligence and Systems*, INTSYS '98, pages 34–40, 1998.

Data Set	Training Accuracy (%)	Testing Accuracy (%)	Testing Type-1 Accuracy (%)	Testing Type-2 Accuracy (%)
1	97	93	95	98
2	99	95	96.5	98
3	98	95	98	96.5
4	99	95	98	96.5
5	99	95	96.5	98
6	98	98	**100**	98
7	98	98	98	**100**
8	97	98	98	100
9	98.5	95	96.5	98
10	99	95	95	100
Average	98.5	96	97	98.5

Table 3: Cross-validation average results on various sets of patient data over five independent runs each.

[19] B. A. G. Licon, J. H. S. Azuela, and R. A. Vázquez. Design of artificial neural networks using a modified particle swarm optimization algorithm. In *IJCNN*, pages 938–945, 2009.

[20] M. K. Markey, J. Y. Lo, G. D. Tourassi, and C. E. Floyd. Self-organizing map for cluster analysis of a breast cancer database. *Artificial Intelligence in Medicine*, 27:113–127, 2003.

[21] J. F. Miller and P. Thomson. Cartesian genetic programming. In *Proc. EuroGP'2000*, volume 1802, pages 121–132, 2000.

[22] D. Moriarty. *Symbiotic Evolution of Neural Networks in Sequential Decision Tasks*. PhD thesis, University of Texas at Austin, 1997.

[23] T. Mu and A. K. Nandi. Breast cancer detection from FNA using SVM with different parameter tuning systems and SOM-RBF classifier. *Journal of the Franklin Institute*, 344:285–311, 2007.

[24] R. Nandi, A. Nandi, R. Rangayyan, and D. Scutt. Classification of breast masses in mammograms using genetic programming and feature selection. *Medical Biological Engineering & Computing*, 44(8):683–94, 2006.

[25] G. Palmer, C. Zhu, T. Breslin, F. Xu, K. Gilchrist, and N. Ramanujam. Comparison of multi excitation fluorescence and diffuse reflectance spectroscopy for the diagnosis of breast cancer. *IEEE Trans Biomed Eng.*, 50(11):1233–42, 2003.

[26] J. Pujol and R. Poli. Evolving the topology and the weights of neural networks using a dual representation. *Applied Intelligence Journal*, 8:73–84, 1998.

[27] J. Reisinger, K. O. Stanley, and R. Miikkulainen. Evolving reusable neural modules. In *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 3103 of *Lecture Notes in Computer Science*, pages 69–81. Springer, 2004.

[28] R. Santo, R. Lopes, and R. Rangayyan. Classification of mammographic masses using radial basis functions and simulated annealing with shape, acutance, and texture features. In *Proceedings of the IASTED Conference on Biomedical Engineering*, pages 458–175, 2005.

[29] J. Schmidhuber, D. Wierstra, and F. Gomez. Evolino: Hybrid neuroevolution / optimal linear search for sequence prediction. In *In Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI*, pages 853–858. Professional Book Center, 2005.

[30] K. O. Stanley, B. D. Bryant, and R. Miikkulainen. Real-time neuroevolution in the nero video game. *IEEE Transactions on Evolutionary Computation*, 9:653–668, 2005.

[31] K. O. Stanley and R. Miikkulainen. Evolving neural network through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[32] L. Wei, Y. Yang, R. M. Nishikawa, and Y. Jiang. A study on several machine-learning methods for classification of malignant and benign clustered microcalcifications. *IEEE Transactions on Medical Imaging*, 24:371–380, 2005.

[33] J. C. Werner and T. C. Fogarty. Genetic programming applied to severe diseases diagnosis. In *Proceedings Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP-2001)*, 2001.

[34] D. West and V. West. Model selection for a medical diagnostic decision support system: a breast cancer detection case. *Artificial Intelligence in Medicine*, 20(3):183–204, 2000.

[35] S. Whiteson and P. Stone. Evolutionary function approximation for reinforcement learning. *Journal of Machine Learning Research*, 7:877–917, 2006.

[36] W. Wolberg, W. Street, and O. Mangasarian. Image analysis and machine learning applied to breast cancer diagnosis and prognosis. *Pubmid:Anal Quant Cytol Histol.*, 17(2):77–87, 1995.

[37] W. H. Wolberg and O. L. Mangasarian. Multisurface Method of Pattern Separation for Medical Diagnosis Applied to Breast Cytology. *Proceedings of the National Academy of Sciences,U.S.A.*, 87:9193–9196, 1990.

[38] X. Yao. Evolving artificial neural networks. In *Proceedings of the IEEE*, volume 87(9), pages 1423–1447, 1999.