Evolving Artificial Neural Networks for DNA Microarray Analysis

Kyung-Joong Kim
Dept. of Computer Science, Yonsei University
134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749,
Korea
uribyul@candy.yonsei.ac.kr

Sung-Bae Cho
Dept. of Computer Science, Yonsei University
134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749,
Korea
sbcho@cs.yonsei.ac.kr

***Abstract-* DNA microarray technology provides a format for the simultaneous measurement of the expression level of thousands of genes in a single hybridization assay. One exciting result of microarray technology has been the demonstration that patterns of gene expression can distinguish between tumors of different anatomical origins. Standard statistical methodologies in classification and prediction do not work well or even at all when *N* (the number of samples) < *p* (genes). Modification of existing statistical methodologies or development of new methodologies are needed for the analysis of cancer. Recently, designing artificial neural networks (ANNs) by evolutionary algorithms has emerged as a preferred alternative to the common practice of selecting the apparent best network. In this paper, we propose an evolutionary neural network that classifies gene expression profiles into normal or colon cancer cell. Colon cancer is the second only to lung cancer as a cause of cancer-related mortality in Western countries. Colon cancer is a genetic disease, propagated by the acquisition of somatic alterations that influence gene expression. Experimental results on colon microarray data with evolutionary neural network show that the proposed method can perform better than other classifiers. Contribution of this paper is applying evolutionary neural network to gene expression classification problem.**

1 Introduction

Evolutionary ANNs combine the learning of neural networks and evolution of evolutionary algorithms [1]. A great deal of works has been done on evolutionary artificial neural networks (EANNs). Evolutionary algorithm can be used for various tasks, such as connection weight training, architecture design, learning rule adaptation, input feature selection, connection weight initialization and rule extraction from ANNs [2].

Recently, experimental techniques based on oligonucleotide or cDNA arrays now allow the expression level of thousands of genes to be monitored in parallel. The critically important thing for cancer diagnois and treatment is precise prediction of tumors. One of the remarkable advances for molecular biology and for cancer research is a DNA microarray technology. DNA microarray datasets have a high dimensionality corresponding to the large number of probes used and there are often comparatively few samples. In this paper, we address the problem of prediction of cancer using a small subset of genes from broad patterns of gene expression data.

In cancer research, microarray technology allows the better understanding of the regulation of activity of cells and tumors in various states [3]. Prediction, classification, and clustering techniques are used for analysis and interpretation of the microarray data. Colon cancer is the second most common cause of cancer mortality in Western countries [4]. Gene expression data in 40 tumor and 22 normal colon tissue samples were gathered with an Affymetrix oligonucleotide array complementary to more than 6,500 human genes [5]. We chose to work only with the 2,000 genes of the greatest minimal expression over the samples [5].

We propose an evolutionary neural network for classifying (predicting) human tumor samples based on microarray gene expressions. This procedure involves dimensionality reduction using information gain and classification using EANN. The proposed methods are applied to colon cancer microarray data sets involving various human tumor samples. We have compared evolutionary neural network to the well-known classification methods [6].

2 Backgrounds

2.1 DNA Microarray Technology

DNA microarrays are used to quantify tens of thousands of DNA or RNA sequences in a single assay [7]. With the development and application of DNA microarrays, the expression of almost all human genes can now be systematically examined in human malignancies [8]. DNA sequences are initially transcribed into mRNA sequences. These mRNA sequences are translated into the amino acid squences of the proteins that perform various functions. Measuring mRNA levels can provide a detailed molecular view of the genes. Measuring gene expression levels under different conditions is important for expanding our knowledge of gene function. Gene expression data can help in better understanding of cancer.

The main goal of analyzing gene expression data is the identification of sets of genes that can serve as classification. Understanding cellular responses to drug treatment is another important goal of gene expression

0-7803-7804-0 /03/$17.00 © 2003 IEEE

profiling. The complexity of microarray data calls for data analysis tools that will effectively aid in biological information mining.

Uncovering broad patterns of genetic activity, providing new understanding of gene functions and generating unexpected insight into biological mechanism are the impact of microarray-based studies [9]. It is essential to efficiently analyze DNA microarray profiles because the amount of DNA microarray data is usually very large. The analysis of DNA microarray data is divided into four branches: clustering, classification, gene identification, and gene regulatory network modeling. Many machine learning and data mining methods have been applied to deal with them. Information theory has been applied to gene identification problems. Also, Boolean network, Bayesian network, and reverse engineering methods have been applied to gene regulatory network modeling problem [10].

Several machine learning techniques have been previously used for classifying gene expression data, including Fisher linear discriminant analysis, *k* nearest neighbor, decision tree, multi-layer perceptron, support vector machine, boosting, and self-organizing map. Also, many machine learning techniques have been used in clustering gene expression data, such as hierarchical clustering, self-organizing map, and graph theoretic approaches.

Clustering methods do not use any tissue annotation (e.g., tumor vs. normal) in the partitioning step. In contrast, classification methods attempt to predict the label of new tissues, based on their gene expression profiles after training on examples (training data) that have been classified by an external "supervision."

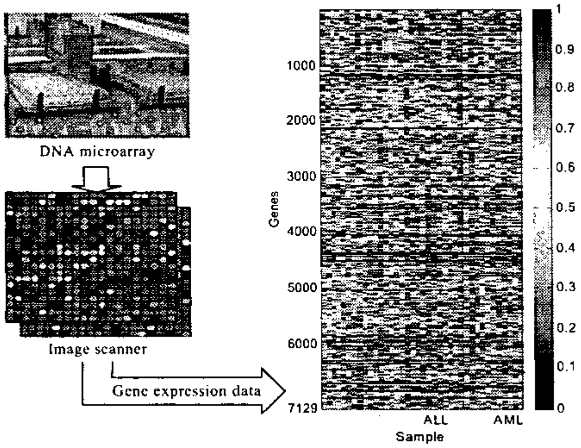


Figure 1. General process of acquiring the gene expression data from DNA microarray

DNA arrays consist of a large number of DNA molecules spotted in a systematic order on a solid substrate. Depending on the size of each DNA spot on the array, DNA arrays can be categorized as microarrays when the diameter of DNA spot is less than 250 microns, and macroarrays when the diameter is bigger than 300 microns. The arrays with the small solid substrate are also referred to as DNA chips. It is so powerful that we can investigate the gene information in short time, because at least hundreds of genes can be put on the DNA microarray to be analyzed.

DNA microarrays are composed of thousands of individual DNA sequences printed in a high density array on a glass microscope slide using a robotic arrayer as shown in Fig. 1. The relative abundance of these spotted DNA sequences in two DNA or RNA samples may be assessed by monitoring the differential hybridization of the two samples to the sequences on the array. For mRNA samples, the two samples are reverse-transcribed into cDNA, labeled using different fluorescent dyes mixed (red-fluorescent dye Cy5 and green-fluorescent dye Cy3). After the hybridization of these samples with the arrayed DNA probes, the slides are imaged using scanner that makes fluorescence measurements for each dye. The log ratio between the two intensities of each dye is used as the gene expression data .

$$gene\_expression = \log_2 \frac{\mathrm{Int(Cy5)}}{\mathrm{Int(Cy3)}}$$

where Int(Cy5) and Int(Cy3) are the intensities of red and green colors. Since at least hundreds of genes are put on the DNA microarray, we can investigate the genome-wide information in short time.

Table 1. Relevant works on colon cancer classification

Authors	Methods		Accuracy [%]
	Feature	Classifier	
Furey *et al.* [11]	Signal to noise ratio	SVM	90.3
Li *et al.* [12]	Genetic algorithm	KNN	94.1
Ben-Dor *et al.* [13]	All genes, TNoM score	Nearest Neighbor	80.6
		SVM with quadratic kernel	74.2
		AdaBoost	72.6
Nguyen *et al.* [14]	Principal component analysis	Logistic Discriminant	87.1
		Quadratic Discriminant	87.1
	Partial least square	Logistic Discriminant	93.5
		Quadratic Discriminant	91.9

2.2 Related Works

Derisi *et al.* pointed out that the expression patterns of many previously uncharacterized genes provide clues to their possible functions [15]. Eisen *et al.* noted that the clustering of gene expression data groups together efficiently genes of known similar function [16]. Shamir described some of the main algorithmic approaches to clustering gene expression data [17]. Getz *et al.* presented a two-way clustering approach to gene microarray data analysis [18]. There are many researchers to attempt to predict colon cancer using various machine learning methods and they show that prediction rate of colon cancer can be approximately 80~90% (Table 1).

There are some related works on EANNs. EANNs combine the advantages of the global search performed by evolutionary algorithms and local search of the learning algorithms (like BP) of ANN. Yao [19] proposed EANNs approach, EPNet based on Fogel's evolutionary progamming (EP) as evolutionary algorithm. EPNet emphasizes the evolution of ANN behaviors by EP and uses a number of techniques, such as partial training after each architectural mutation and node splitting, to maintain the behavioral link between parent and its offspring effectively. EPNet also encourages parsimony of evolved ANNs by attempting different mutations sequentially. That is, node or connection deletion is always attempted before addition. EPNet has shown good performance in an error rate and a size of ANN.

Cho proposed a new approach constructing multiple neural networks that uses genetic algorithms with speciation to generate a population of ANNs that are accurate and diverse [20]. Speciation in genetic algorithm creates different species, each embodying a sub-solution, which means to create diverse solutions not the best one. Experiments with the Breast cancer data from UCI benchmark datasets show that the method can produce more speciated ANNs and improve the performance by combining them. Several methods are applied to combine speciated neural networks.

Keedwell deals with development of a neural-gaenetic hybrid algorithm which is capable of extracting both the gene regularatory network and classification information from gene expression data [21].

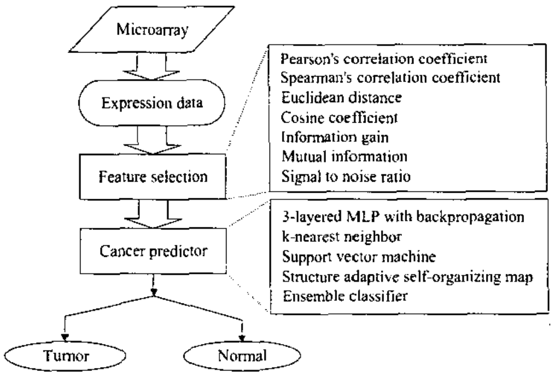


Figure 2. Cancer classification system

2.3 Machine Learning for DNA Microarray Analysis

We define machine learning for DNA microarray that selects discriminative genes related with classification from gene expression data, trains classifier and then classifies new data using the learned classifier. The system is shown in Fig. 2. After acquiring the gene expression data calculated from the DNA microarray, our prediction system has 2 stages: feature selection and pattern classification.

The feature selection can be thought of as the gene selection, which is to get the list of genes that might be informative for the prediction by statistical, information theoretical methods, etc. Since it is highly unlikely that all the 7,129 genes have the information related to the cancer and using all the genes results in too big dimensionality, it is necessary to explore the efficient way to get the best feature. We have extracted 30 genes using information gain, and the cancer predictor classifies the category only with these genes [22].

Given the gene list, a classifier makes decision as to which category the gene pattern belongs at prediction stage. We have adopted four most widely used classification methods for comparison as shown in Fig. 2.

3 Evolutionary Neural Network

Recently, designing an artificial neural networks (ANNs) by evolutionary algorithms has emerged as a preferred alternative to the common practice of selecting the apparent best network.

3.1 Representation

To evolve an ANN, it needs to be expressed in proper form. There are some methods to encode an ANN like binary representation, tree, linked list, and matrix. We have used a matrix to encode an ANN since it is straightforward to implement and easy to apply genetic operators [23]. When N is the total number of nodes in an ANN including input, hidden, and output nodes, the matrix is $N \times N$, and its entries consist of connection links and corresponding weights. In the matrix, upper right triangle (see the Fig. 3) has connection link information which describes 1 when there exists connection link and 0 when there is no connection link. Lower left triangle describes the weight value corresponding to connection link information. Fig. 3 shows an example of encoding of an ANN that has one input node, three hidden nodes, and one output node.

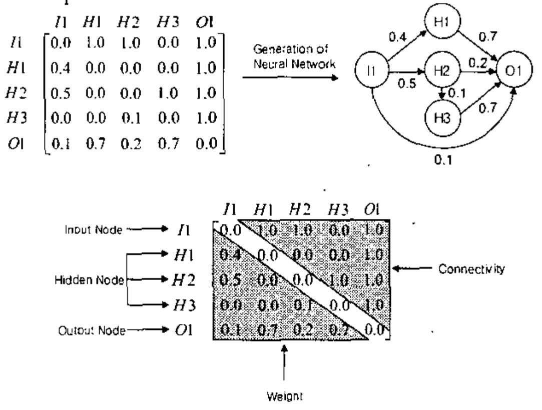


Figure 3. An example of neural network representation

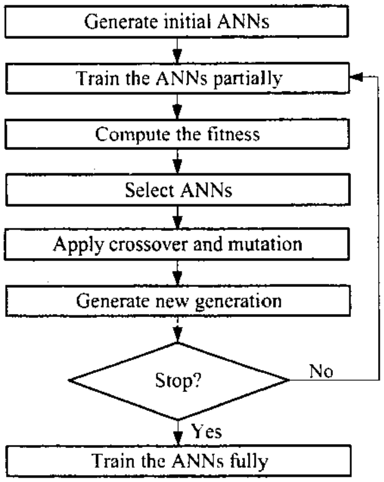


Figure 4. Genetic algorithm for evolving neural network

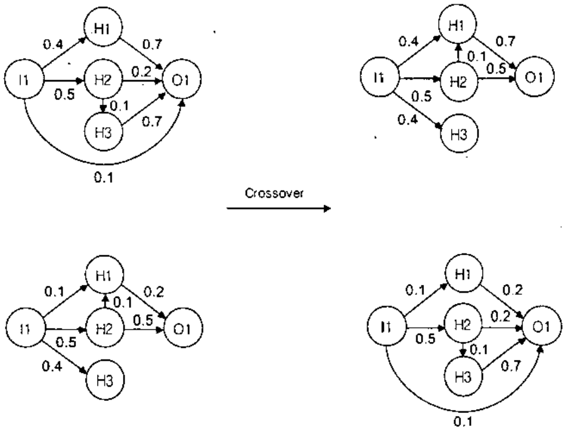


Figure 5. Crossover operation

3.2 The Evolutionary Algorithms

Fig. 4 shows the overview of evolving neural network. Each ANN is generated with random initial weights and full-connection. The fitness of ANN is a recognition rate of validation data. Then, each ANN is trained partially with training data to help the evolution search the architecture of ANN and is tested with validation data to compute the fitness. Once the fitness is caculated, selection that chooses the best 50% individuals to apply genetic operators is conducted. The genetic operators, crossover and mutation, are applied to those selected individuals. Then the next generation is created. The process is repeated until stop criterion is satisfied. Finally, the ANNs in the last generation are trained fully.

3.3 Crossover

The crossover operator exchanges the architecture of two ANNs in the population to search ANNs with various architectures [24]. In the population of ANNs, crossover operator selects two distinct ANNs randomly and chooses one hidden node from each selected ANN. These two nodes should be in the same entry of each ANN matrix encoding the ANN to exchange the architectures. Once the nodes are selected, the two ANNs exchange the connection links and correspoding weight information of the nodes and the hidden nodes after that. Fig. 5 shows an example of crossover. In this example, two ANNs have one input node, three hidden nodes, and one output node. Links related to the H2 node are exchanged.

3.4 Mutation

The mutation operator changes a connection link and a corresponding weight of a randomly selected ANN from the population. Mutation operator performs one of the two operations that are addition of a new connection and deletion of an existing connection. Mutation operator selects an ANN from the population of ANNs randomly and chooses one connection link from it. If the connection link does not exist and the connection entry of the ANN matrix is 0, the connection link is added. It adds new connection link to the ANN with random weights. Otherwise, if the connection link already exists, the connection is deleted. It deletes the connection link and weight information. Fig. 6 shows two examples of the mutation.

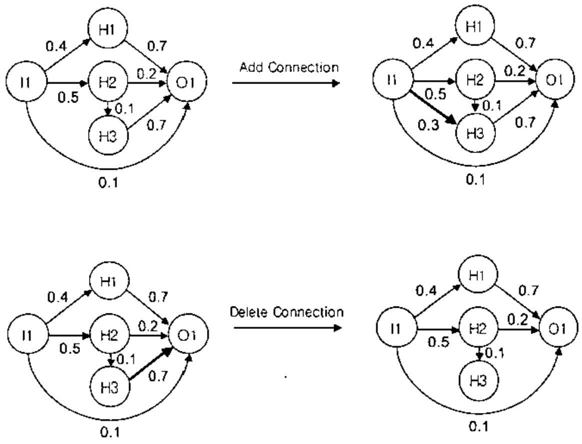


Figure 6. Mutation operation

4 Experimental Results

Colon cancer dataset consists of 62 samples of colon epithelial cells taken from colon-cancer patients. Each sample contains 2000 gene expression levels. Although original data consist of 6000 genes expression levels, 4000 out of 6000 were removed based on the confidence in the measured expression levels. 40 of 62 samples are colon cancer samples and the remaining are normal samples. Each sample was taken from tumors and normal healthy parts of the colons of the same patients and measured using high-density oligonucleotide arrays. 31 out of 62 samples were used as training data and the remaining were used as test data in this paper. (Available at http://www.sph.uth.tmc.edu:8052/hgc/default.asp)

Table 2. 15 genes selected by information gain

	Name
1	Human monocyte-derived neutrophil-activating protein (MONAP) mRNA, complete cds.
2	Human desmin gene, complete cds.
3	MYOSIN HEAVY CHAIN, NONMUSCLE (Gallus gallus)
4	Human cysteine-rich protein (CRP) gene, exons 5 and 6.
5	COLLAGEN ALPHA 2(XI) CHAIN (Homo sapiens)
6	Human gene for heterogeneous nuclear ribonucleoprotein (hnRNP) core protein A1.
7	P03001 TRANSCRIPTION FACTOR IIIA ;.
8	MYOSIN REGULATORY LIGHT CHAIN 2, SMOOTH MUSCLE ISOFORM (HUMAN);contains element TAR1 repetitive element ;.
9	MITOCHONDRIAL MATRIX PROTEIN P1 PRECURSOR (HUMAN);.
10	Human aspartyl-tRNA synthetase alpha-2 subunit mRNA, complete cds.
11	Human cysteine-rich protein (CRP) gene, exons 5 and 6.
12	Human cysteine-rich protein (CRP) gene, exons 5 and 6.
13	Human homeo box c1 protein, mRNA, complete cds.
14	MACROPHAGE MIGRATION INHIBITORY FACTOR (HUMAN);.
15	Human splicing factor SRp30c mRNA, complete cds.

4.1 Feature Selection

As mentioned before, the feature size of colon dataset is 2000. It is too large to manipulate in learning algorithm and all features are not useful for classification. Only relevant features are useful for classification to show better performance. Feature ranking method is used to classify the genes. Information gain is a representative feature ranking and selection method that is used in C4.5. If there are two classes and they are C_1 and C_2, U is the set of all instances and each instance has pre-determined class (C_1 or C_2). Each instance is represented with the vector of 2000 genes and each gene becomes one feature. Information gain of a gene is defined as follows.

$$IG(G_i,C_j) = P(G_i,C_j)\log\frac{P(G_i,C_j)}{P(C_j)P(G_i)} + P(\overline{G_i},C_j)\log\frac{P(\overline{G_i},C_j)}{P(C_j)P(\overline{G_i})}$$

where G_i is the ith gene and $P(G_i,C_j)$ means the probability G_i is true when the class is C_j. There is no answer for an optimal number of features for classification but approximately 20~40 genes are appropriate for classification. In this paper, we use 30 genes for classification that has high information gain in the feature ranking. Table 2 shows the name of 15 genes that are selected.

4.2 Classifiers

SASOM (structure adaptive self-organizing maps) has been used by 4×4 map with rectangular topology, 0.05 of initial learning rate, 1000 of initial maximum iteration, 10 of initial radius, 0.02 of final learning rate, 10000 of final maximum iteration and 3 of final radius. We have used SVM's with linear function and RBF function as kernel function. In RBF, we have changed 0.1~0.5 gamma variable. We have used 3-layered MLP with 5~15 hidden nodes, 2 output nodes, 0.01~0.50 of learning rate and 0.9 of momentum. Similarity measures used in KNN are Pearson's correlation coefficient and Euclidean distance. KNN has been used with k=1~8.

4.2.1MLP

Error backpropagation neural network is a feed-forward multilayer perceptron (MLP) that is applied in many fields due to its powerful and stable learning algorithm. The neural network learns the training examples by adjusting the synaptic weights of neurons according to the error occurred on the output layer. The power of the backpropagation algorithm lies in two main aspects: local for updating the synaptic weights and biases, and efficient for computing all the partial derivatives of the cost function with respect to these free parameters. The weight-update rule in backpropagation algorithm is defined as follows:

$$\Delta w_{ji}(n) = \eta\delta_j x_{ji} + \alpha\Delta w_{ji}(n-1)$$

where $\Delta w_{ji}(n)$ is the weight update performed during the nth iteration through the main loop of the algorithm, η is a positive constant called the learning rate, δ_j is the error term associated with j, x_{ji} is the input from node i to unit j, and $0 \leq \alpha < 1$ is a constant called the *momentum*.

4.2.2 KNN

k-nearest neighbor (KNN) is one of the most common methods among memory based induction. Given an input vector, KNN extracts k closest vectors in the reference set based on similarity measures, and makes decision for the label of input vector using the labels of the k nearest neighbors.

Pearson's coefficient correlation and Euclidean distance have been used as the similarity measure. When we have an input X and a reference set $D = \{d_1, d_2, \ldots, d_N\}$, the probability that X may belong to class c_j, $P(X, c_j)$ is defined as follows:

$$P(X,c_j) = \sum_{d_i \in kNN} \mathrm{Sim}(X,d_i)P(d_i,c_j) - b_j$$

where Sim(X, d_i) is the similarity between X and d_i and b_j is a bias term.

4.2.3 SASOM

Self-organizing map (SOM) defines a mapping from the input space onto an output layer by unsupervised learning algorithm. SOM has an output layer consisting of N nodes, each of which represents a vector that has the same dimension as the input pattern. For a given input vector X, the winner node m_c is chosen using Euclidean distance between x and its neighbors, m_i.

$$\|x - m_c\| = \min_i \|x - m_i\|$$

$$m_i(t+1) = m_i(t) + \alpha(t) \times n_{ci}(t) \times \{x(t) - m_i(t)\}$$

Even though SOM is well known for its good performance of topology preserving, it is difficult to apply it to practical classification since the topology should be fixed before training. A structure adaptive self-organizing map (SASOM) is proposed to overcome this shortcoming. SASOM starts with 4×4 map, and dynamically splits the output nodes of the map, where the data from different classes are mixed, trained with the LVQ learning algorithm.

4.2.4 SVM

Support vector machine (SVM) estimates the function classifying the data into two classes. SVM builds up a hyperplane as the decision surface in such a way to maximize the margin of separation between positive and negative examples. SVM achieves this by the structural risk minimization principle that the error rate of a learning machine on the test data is bounded by the sum of the training-error rate and a term that depends on the Vapnik-Chervonenkis (VC) dimension. Given a labeled set of M training samples (X_i, Y_i), where $X_i \in R^N$ and Y_i is the associated label, $Y_i \in \{-1, 1\}$, the discriminant hyperplane is defined by:

$$f(X) = \sum_{i=1}^{M} Y_i \alpha_i k(X, X_i) + b$$

where $k(\ .)$ is a kernel function and the sign of $f(X)$ determines the membership of X. Constructing an optimal hyperplane is equivalent to finding all the nonzero α_i (support vectors) and a bias b. We have used SVM^{light} module and SVM^{RBF} in this paper.

4.3 Results and Analysis

Parameters of genetic algorithm are as follows. In EANN, the population size is 20 and the maximum generation number is 200. Each ANN is feed-forward ANN and back-propagation is used as learning algorithm. Learning rate is 0.1 and the partial training presents the training data 200 times and full training presents the training data 1000 times. Crossover rate is 0.3 and mutation rate is 0.1. Elitism is adopted. Fitness function of EANN is defined as the recognition rate for validation data. In colon dataset, the number of data sample is very small and we use test data as validation set.

We have conducted 10 runs of experiments to get the average. Fig. 7 shows variation of 10 runs, and min, max and average of 20 individuals in the last generation for each run. Figure 8 shows the comparison of classifiers' performance and EANN performs well. The results of other classifiers are the best of many trials. Figure 9 shows 10-fold cross validation results. The neural network that shows the best performance contains 203 connections. The number of connections from input nodes to hidden nodes is 147, from input nodes to output nodes is 26, from hidden nodes to hidden nodes is 25, and from hidden nodes to output nodes is 5. The neural network contains 42 nodes: 30 input nodes, 10 hidden nodes and 2 output nodes. Figure 10 shows four different connections among nodes. Figure 11 shows the best neural network.

Table 3 summarizes confusion matrix of the best evolutionary neural network. The network produces wrong classification in 24 and 30 (sample id). Sensitivity of the classifier is 81.8% and specificity is 100.0%. This means that the classifier does not classify normal person into patient but it classifies patient into normal person with the probability of 18.2%. This means that if the person whom the classifier decides as a normal is a patient with the probability of 9%. Relationship between specificity and sensitivity is negatively correlated and the cost for misclassification for two cases is important point to decide the level of two measures. In practice, the cost of classifying normal person into cancer person is cheaper than classifying cancer person into normal person. In the first case, normal person needs only another test with doctor but the second case makes worse situation for the person.

Table 3. Confusion matrix of the best EANN

EANN		Predicted	
		0 (Normal)	1 (Cancer)
Actual	0 (Normal)	20	0
	1 (Cancer)	2	9

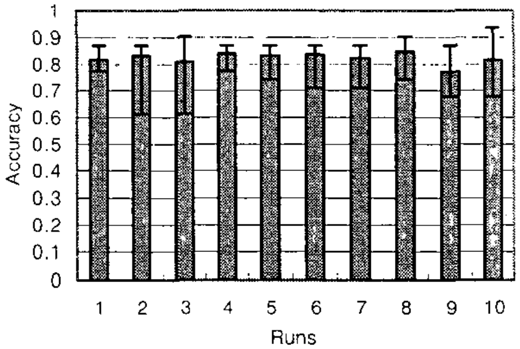
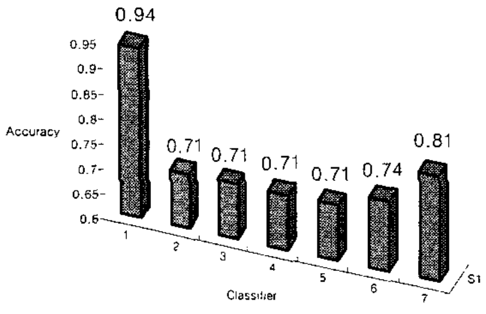


Figure 7. Max, min and average accuracy of 10 runs



1: EANN 2: MLP 3: SASOM 4: SVM (Linear) 5: SVM (RBF) 6: KNN (COSINE) 7: KNN (PEARSON)

Figure 8. Comparison of classification rate (maximum accuracy for each classifier)

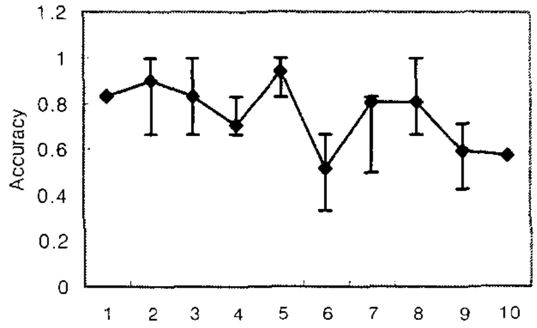


Figure 9. 10-fold cross validation results

5 Conclusions

The problem of distinguishing normal from tumor samples is an important one. We have introduced evolutionary neural network methods for the classification of tumors based on microarray gene expression data. The methodologies involve dimension reduction of the high dimensional gene expression space followed by information gain. We have illustrated the method's effectiveness in predicting normal and tumor samples in colon cancer data set. The methods are able to distinguish between normal and tumor samples with high accuracy. There are many approaches to predict cancer data using machine learning including SASOM, SVM, MLP and KNN. EANN is hybrid method of evolutionary algorithm and neural network to find solution without expert knowledge. Comparison with other classifiers shows that EANN performs very well. In comparison with other methods that are published, EANN is competitive.

Acknowledgement

This research was supported by Brain Science and Engineering Research Program sponsored by Korean Ministry of Science and Technology.

Bibliography

[1] M. Conrad, "Computation: Evolutionary, neural, molecular," *2000 IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*, pp. 1-9, 2000.

[2] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423-1447, Sep 1999.

[3] D. A. Rew, "DNA microarray technology in cancer research," *European Journal of Surgical Oncology*, vol. 27, no. 5, pp. 504-508, Aug 2001.

[4] G. A. Chung-Faye, D. J. Kerr, L. S. Young and P. F. Searle, "Gene theraphy strategies for colon cancer," *Molecular Medicine Today*, vol. 6, no. 2, pp. 82-87, Feb 2000.

[5] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proc. Natl. Acad. Sci. USA*, vol. 96, pp. 6745-6750, June 1999.

[6] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, Wiley Interscience, 2000.

[7] D. D. Shoemaker and P. S. Linsley, "Recent developments in DNA microarrays," *Current Opinion in Microbiology*, vol. 5, no. 3, pp. 334-337, 2002.

[8] G. M. Hampton and Jr. H. F. Frierson, "Classifying human cancer by analysis of gene expression," *Trends in Molecular Medicine*, vol. 9, no. 1, pp. 5-19, Jan 2003.

[9] C. A. Harrington, C. Rosenow and J. Retief, "Monitoring gene expression using DNA microarrays," *Current Opinion in Microbiology*, vol. 3, pp. 285-291, 2000.

[10] I. Shmulevich, E. R. Dougherty and W. Zhang, "From Boolean to probabilistic Boolean networks as a models of genetic regulatory networks," *Proceedings of the IEEE*, vol. 90, no. 11, pp. 1778-1792, 2002.

[11] S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906-914, 2000.

[12] L. Li, C. R. Weinberg, T. A. Darden and L. G. Pedersen, "Gene selection for sample classification based on gene expression data: Study of sensitivity to choice of parameters of the GA/KNN method," *Bioinformatics*, vol. 17, no. 12, pp. 1131-1142, 2001.

[13] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and N. Yakhini, "Tissue classification with gene expression profiles," *Journal of Computational Biology*, vol. 7, pp. 559-584, 2000.

[14] D. V. Nguyen and D. M. Rocke, "Tumor

classification by partial least squares using microarray gene expression data," *Bioinformatics*, vol. 18, no. 1, pp. 39-50, 2002.

[15] J. Derisi, V. Iyer and P. Brosn, "Exploring the metabolic and genetic control of gene expression on a genomic scale," *Science*, vol. 278, pp. 680-686, 1997.

[16] M. B. Eisen, P. T. Spellman, P. O. Brown and D. Bostein, "Cluster analysis and display of genome-wide expression patterns," *Proc. of the Natl. Acad. of Sci., USA*, vol. 95, pp. 14863-14868, 1998.

[17] R. Shamir and R. Sharan, "Algorithmic approaches to clustering gene expression data," *Current Topics in Computational Biology*, In T. Jiang, T. Smith, Y. Xu and M. Q. Zhang (eds), MIT Press, 2001.

[18] G. Getz, E. Levine and E. Domany, "Coupled two-way clustering analysis of gene microarray data," *Proc. Natl. Acad. Sci. USA*, vol. 97, no. 22, pp. 12079-12084, 2000.

[19] X. Yao and Y. Liu, "A new evolutionary system for evolving artificial neural networks," *IEEE Trans. on Neural Networks*, vol. 8, no. 3, pp. 694-713, May 1997.

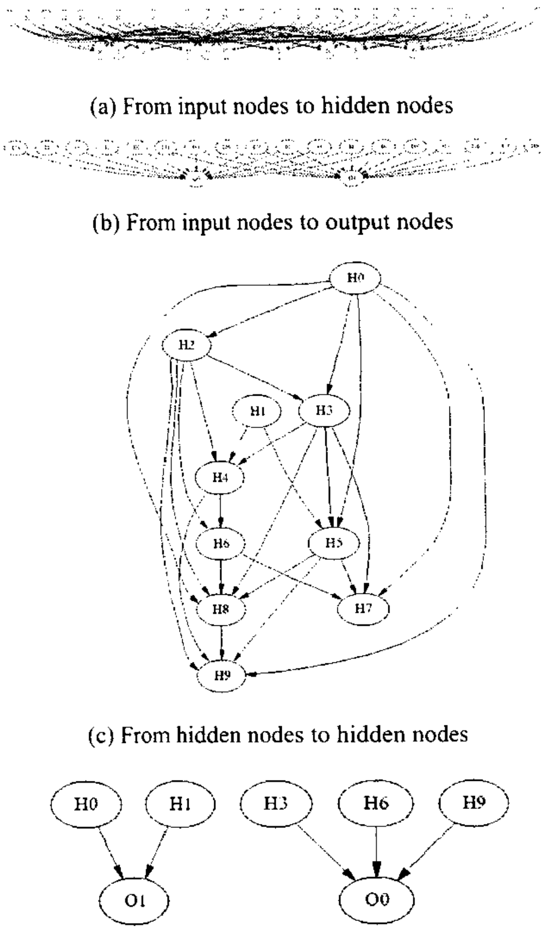
[20] J.-H. Ahn and S.-B. Cho, "Speciated neural networks evolved with fitness sharing technique," *Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 390-396, 2001.

[21] A. Narayanan, E. Keedwell, and B. Olsson, "Artificial intelligence techniques for bioinformatics," *Applied Bioinformatics*, vol. 1, no. 4, pp. 191-222, 2003.

[22] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.

[23] D. W. Taylor, D. W. Corne, D. L. Taylor, and J. Harkness, "Predicting alarms in supermarket refrigeration systems using evolved neural networks and evolved rulesets," *Congress on Evolutionary Computation*, pp. 1988-1993, 2002.

[24] D. Montana, L. Davis, "Training feedforward neural networks using genetic algorithms," *Proceedings of the Eleventh International Conference on Artificial Intelligence*, pp. 762-767, 1989.



(a) From input nodes to hidden nodes

(b) From input nodes to output nodes

(c) From hidden nodes to hidden nodes

(d) From hidden nodes to output nodes

Figure 10. Connection of nodes by the four different types (The best evolved network)

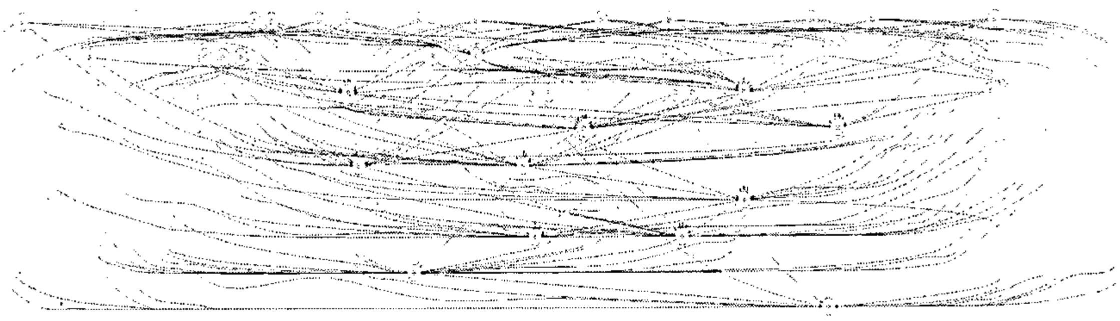


Figure 11. The best evolved neural network architecture