Classification Study on DNA Microarray with Feedforward Neural Network Trained by Singular Value Decomposition

Hieu Trung Huynh[1], Jung-Ja Kim[2] and Yonggwan Won[1]

[1]Department of Computer Engineering, Chonnam National university
300 Yongbong-dong, Buk-gu, Gwangju 500-757, Korea
hthieu@hcmut.edu.vn, ykwon@chonnam.ac.kr
[2] Dicision of Bionics and Bioinformatics, Chonbuk National University
664-14 St. #1 Dukjin-dong, Dukjin-gu, Chonbuk 561-756, Korea
jungjakim@chonbuk.ac.kr

Abstract

DNA microarray is a multiplex technology used in molecular biology and biomedicine. It consists of an arrayed series of thousands of microscopic spots of DNA oligonucleotides, called features, of which the result should be analyzed by computational methods. Analyzing microarray data using intelligent computing methods has attracted many researchers in recent years. Several approaches have been proposed, in which machine learning based approaches play an important role for biomedical research such as gene expression interpretation, classification and prediction for cancer diagnosis, etc. In this paper, we present an application of the feedforward neural network (SLFN) trained by the singular value decomposition (SVD) approach for DNA microarray classification. The classifier of the single hidden-layer feedforward neural network (SLFN) has the activation function of the hidden units to be 'tansig'. Experimental results show that the SVD trained feedforward neural network is simple in both training procedure and network structure; it has low computational complexity and can produce better performance with compact network architecture.

Keywords: *DNA microarray, classification, feedforward neural network, singular value decomposition.*

1. Introduction

DNA microarray is a high-throughput multiplex technology used in molecular biology and biomedicine. It consists of an arrayed series of thousands of microscopic spots of DNA oligonucleotides, called features, of which the result should be analyzed by computational methods. It offers the ability to study samples with small amount and to carry out the expression of thousands of genes at once. This makes possible to obtain vast amounts of information very quickly, from which the diagnosis and prognosis of disease based on the gene expression profiles can be well established. One of the typical applications of machine learning in computational biology is DNA microarray classification. The aim of this work is to identify patterns of expressed genes, from which it can predict class membership for new patterns.

Many methods for classification tasks such as support vector machine (SVM), neural networks, or statistical techniques have been used for analysis of microarray data. However, the statistical techniques are improper because the microarray data is very high dimensional with the limited number of patterns and very little replication. SVM

approach may take long time to select its model. The problem facing in applications of neural networks is training algorithms. Traditionally, training networks is based on gradient-descent algorithms which are generally slow and may get stuck in local minima. These problems have surmounted by extreme learning machine (ELM) algorithm which was recently proposed by Huang et al. [1] and is suitable for training single hidden-layer feedforward neural networks (SLFNs).

One of the most attractive points of ELM algorithm is that the network parameters are determined by non-iterative steps, in which the input weights and hidden layer biases are randomly assigned and the output weights are determined by pseudo-inverse of hidden layer output matrix. This algorithm can obtain better performance with higher learning speed in many applications. However, it often requires a large number of hidden nodes which makes the trained networks respond slowly to new input patterns. These problems have been overcome by the improved ELM algorithms such as Least Squares Extreme Learning Machine (LS-ELM) [2], Regularized Least Squares Extreme Leaning Machine (RLS-ELM) [3], Evolutionary Extreme Learning Machine (E-ELM) [4] and evolutionary least-squares extreme learning machine [5].

In our previous work [6], we introduced an effective training algorithm for SLFNs, of which the hidden layer activation function is '*tansig*'. The classifier based on this training algorithm was named as *SVD-Neural classifier*, of which the parameters are determined by non-iterative simple steps based on Singular Value Decomposition (SVD). In this paper, we investigate an application of our SVD-Neural classifier for DNA microarray classification. Experimental results show that the SVD-Neural classifier can obtain better performance with fast learning speed.

The rest of this paper is organized as follows. Section 2 describes the SVD-Neural classifier and its application for DNA microarray analysis. Experimental results and analysis on real datasets for cancer diagnosis are shown in Section 3. Finally, we make conclusion in Section 4.

2. Single Layer Feedforward Neural Network Trained by SVD

A single layer feedforward neural network with the activation function of '*tangsig*' can be trained by singular value decomposition (SVD) [6]. The architecture of this SVD-Neural classifier, which is the SLFN, is shown in Figure 1. It consists of P nodes in input layer, N nodes in hidden layer and C nodes in output layer. Unlikely to other neural network, the activation of hidden node must be '*tansig*' function defined by

$$f(x)=\frac{1-e^{2x}}{1+e^{2x}}. \tag{1}$$

Let $\mathbf{w}_m=[w_{m1}, w_{m2},\ldots, w_{mP}]^T$ be the input weights connecting from the input layer to the m-th hidden node, and b_m be its bias. The hidden layer output vector corresponding to the input pattern $\mathbf{x}_j$ is given by

$$\mathbf{h}_j=[f(\mathbf{w}_1\cdot\mathbf{x}_j+b_1), f(\mathbf{w}_2\cdot\mathbf{x}_j+b_2),\ldots, f(\mathbf{w}_N\cdot\mathbf{x}_j+b_N)]^T, \tag{2}$$

and the i-th output is given by

$$o_{ji}=\mathbf{h}_j\cdot\boldsymbol{\alpha}_i, \tag{3}$$

where $\boldsymbol{\alpha}_i=[\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{iN}]^T$ is the weight vector connecting from the hidden nodes to the i-th output node.

Assume that there are n profiles or arrays in the training set $\mathbf{S}=(\mathbf{x}_j, \mathbf{t}_j)$, $j = 1, \ldots n$; each profile consists of P genes, $\mathbf{x}_j \in \mathbf{R}^P$; and let $\mathbf{t}_j=[t_{j1}, t_{j2},\ldots, t_{jC}]^T$ be the desired output corresponding to the input $\mathbf{x}_j$. We have to find parameters $\mathbf{G}$ consisting of input weights $\mathbf{w}$, biases b and output weights $\boldsymbol{\alpha}$ that minimize the error function defined by

$$E=\sum_{j=1}^{n}\left(\mathbf{o}_j-\mathbf{t}_j\right)^2=\sum_{j=1}^{n}\sum_{i=1}^{C}(\mathbf{h}_j\cdot\boldsymbol{\alpha}_i-t_{ji})^2 . \tag{4}$$

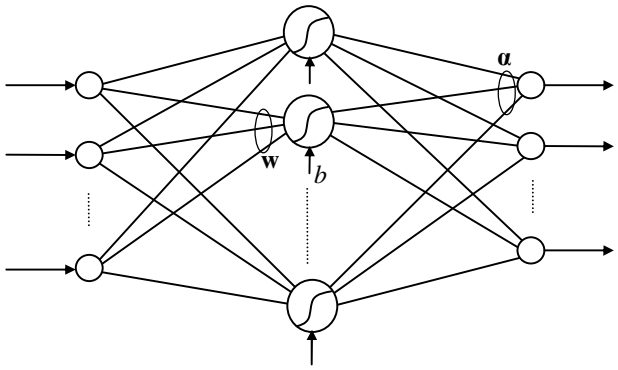

Figure 1. Architecture of classifier

This is a least-squares approach of the linear problem defined by

$$\mathbf{HA}=\mathbf{T}, \tag{5}$$

where $\mathbf{H}$ is called the hidden layer output matrix and defined by:

$$\begin{aligned}\mathbf{H} &= [\mathbf{h}_1, \mathbf{h}_2,\ldots, \mathbf{h}_n]^T \\ &= \begin{bmatrix} f(\mathbf{w}_1\cdot\mathbf{x}_1+b_1) & \cdots & f(\mathbf{w}_N\cdot\mathbf{x}_1+b_N) \\ \vdots & \ddots & \vdots \\ f(\mathbf{w}_1\cdot\mathbf{x}_n+b_1) & \cdots & f(\mathbf{w}_N\cdot\mathbf{x}_n+b_N) \end{bmatrix},\end{aligned} \tag{6}$$

$$\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n]^T \tag{7}$$

and

$$\mathbf{A}= [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2,\ldots, \boldsymbol{\alpha}_C]. \tag{8}$$

From (5), we can see that the matrix $\mathbf{T}$ is composed of the multiplication of two matrices $\mathbf{H}\in\mathbf{R}^{nxN}$ and $\mathbf{A}\in\mathbf{R}^{NxC}$. Thus, if we can reasonably decompose the matrix $\mathbf{T}$ into two matrices with sizes of nxN and NxC then parameters $\mathbf{G}$ of classifier can be determined simply.

In singular value decomposition (SVD) of $\mathbf{T}$, there exist an unitary matrix $\mathbf{U}\in\mathbf{R}^{nxn}$, a diagonal matrix with nonnegative real numbers $\mathbf{D}\in\mathbf{R}^{nxC}$, and an unitary matrix $\mathbf{V}\in\mathbf{R}^{CxC}$ so that

$$\mathbf{T}=\mathbf{UDV}^T \tag{9}$$

Let $\mathbf{U}=[\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N,\ldots, \mathbf{u}_n]$, $\mathbf{D}=[\sigma_1, \sigma_2,\ldots, \sigma_n]^T$, and $\mathbf{V}=[\mathbf{v}_1, \mathbf{v}_2, \ldots., \mathbf{v}_C]^T$. Based on the properties of SVD decomposition, matrix $\mathbf{T}$ can be approximated by:

$$\mathbf{T}\approx\mathbf{U}_N\mathbf{D}_N\mathbf{V}^{\mathrm{T}}, \tag{10}$$

where $\mathbf{U}_N=[\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N]$ and $\mathbf{D}_N=[\sigma_1, \sigma_2,\ldots, \sigma_N]^T$. Note that $\mathbf{U}_N\in\mathbf{R}^{n\mathrm{x}N}$ and $\mathbf{D}_N\mathbf{V}^{\mathrm{T}}\in\mathbf{R}^{N\mathrm{x}C}$. Therefore, matrices $\mathbf{H}$ and $\mathbf{A}$ can be determined by:

$$\mathbf{H}=\mathbf{U}_N \tag{11}$$

and

$$\mathbf{A}=\mathbf{D}_N\mathbf{V}^T. \tag{12}$$

Next, we have to determine the input weights and hidden layer biases. From (6) and (11), we have

$$\begin{bmatrix} f(\mathbf{w}_1\cdot\mathbf{x}_1+b_1) & \cdots & f(\mathbf{w}_N\cdot\mathbf{x}_1+b_N) \\ \vdots & \ddots & \vdots \\ f(\mathbf{w}_1\cdot\mathbf{x}_n+b_1) & \cdots & f(\mathbf{w}_N\cdot\mathbf{x}_n+b_N) \end{bmatrix}=\mathbf{U}_N. \tag{13}$$

The input weights and hidden layer biases can be determined by using the linear system:

$$\mathbf{XW}=f^{-1}[\mathbf{U}_N]. \tag{14}$$

Note that $f^{-1}[\mathbf{U}_N]_{ij}=f^{-1}([\mathbf{U}_N]_{ij})$,

$$\mathbf{X}=\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}^T$$

and

$$\mathbf{W}=\begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \ldots & \mathbf{w}_N \\ b_1 & b_2 & \ldots & b_N \end{bmatrix}.$$

Finally, the minimum norm solution for $\mathbf{W}$ among all possible solutions is given by:

$$\hat{\mathbf{W}}=\mathbf{X}^{\dagger}f^{-1}[\mathbf{U}_N]. \tag{15}$$

In summary, the training algorithm for SVD-Neural classifier can be described as follows:

Given a training set $\boldsymbol{S}=\{(\mathbf{x}_j,\mathbf{t}_j) \mid j=1,\ldots,n\}$*, and number of hidden nodes N.*

1. Determine SVD of $\mathbf{T}$, and then calculate $\mathbf{U}_N$ and $\mathbf{D}_N$.
2. Determine the input weights and hidden layer biases by using (15).
3. Determine the output weights by using (12).

Thus, parameters of classifier can be simply determined by three steps. It is simple and has low computational complexity. It can obtain a compact network with small number of hidden nodes and produce better performance for microarray data classification as shown in the following section.

3. Classification of DNA Microarray for Cancer Diagnosis

In this section, the performance of the SVD-Neural classifier and comparison with ordinary ELM [1] and RLS-ELM [3] are reported. The datasets for this study is

described in Table 1. They have been widely used for the benchmark problems. They consist of two binary cancer classification problems: Leukemia data set [7] and colon cancer dataset [8]. The initial leukemia data set consisted of 38 bone marrow samples obtained from adult acute leukemia patients at the time of diagnosis, of which 11 suffer from acute myeloid leukemia (AML) and 27 suffer from acute lymphoblastic leukemia (ALL). An independent collection of 34 leukemia samples contained a broader range of samples: the specimens consisted of 24 bone marrow samples and 10 periferal blood samples were derived from both adults and children. The number of input features was 7,129. The objective is to separate the AML samples from the ALL samples. The training set consisted of 38 patterns and 34 patterns were used for testing.

Table 1. Specification of the data sets

Data set	Training set	Testing set	Gene expression levels
Leukemia	38	34	7,129
Colon	40	22	2,000

Table 2. Architecture comparison of SLFN training algorithms

Dataset	Methods	# hidden nodes
Leukimia	SVD-Neural	2
	RLS-ELM	2
	ELM	20
	BP	2
Colon	SVD-Neural	2
	RLS-ELM	2
	ELM	20
	BP	2

The colon cancer data set contains the expression of the 2000 genes with highest minimal intensity across the 62 tissues derived from 40 tumor and 22 normal colon tissue samples [8]. The gene expression was analyzed with an Affymetrix (Sata Clara, CA U.S.A.) oligonucleotide array complementary to more than 6,500 human genes. The gene intensity has been derived from about 20 feature pairs that correspond to the gene on the DNA microarray chip by using a filtering process. Details for data collection methods and procedures are described in [8], and the data set is available from the website *http://microarray.princeton.edu/oncology/*.

The average results of fifty trials on two data sets are shown in Table 2 and Table 3, which shows comparison results of SVD-neural classifier approach with other training

algorithms for SLFN such as ELM, RLS-ELM, and back-propagation (BP). The training algorithms were implemented in MATLAB 7.0 environment. The input features were normalized into the range [-1, 1]. The number of hidden nodes was gradually increased by 1 to find the near-optimal number of nodes based on cross-validation method.

From Table 2, we can see that the number of hidden nodes for SVD-Neural classifier was 2 for two data sets, which is equal to that of RLS-ELM and BP algorithm, and is about 10 times smaller than that of ELM algorithm. As shown in Table 3, the SVD-Neural classifier can obtain the classification accuracies of 95.93% and 83.63% for testing sets of Leukemia and Colon datasets, respectively. This show that the SVD-Neural classifier is comparable to RLS-ELM while outperforming classifiers based ELM and BP algorithms.

Table 3. Performance comparison with other classification methods (%)

Dataset	Method	Training	Testing
Leukimia	SVD-Neural	**100±0.00**	**95.93±5.11**
	RLS-ELM [3]	100±0.00	95.60±4.45
	ELM [3]	91.35±4.78	67.65±11.93
	BP [3]	98.85±9.96	88.52±14.36
Colon	SVD-Neural	**100±0.00**	**83.63±6.15**
	RLS-ELM [9]	99.75±0.79	83.27±6.61
	ELM [9]	88.35±5.06	64.18±10.50
	BP	95.70±10.45	80.27±10.53

4. Conclusion

DNA microarrays can be used to measure changes in expression levels, to detect single nucleotide polymorphisms (SNPs), and to genotype or re-sequence mutant genomes. Also, the DNA microarray experiment can be used for diagnosis of many diseases, especially for cancer diagnosis, which needs accurate and reliable data analysis methods. Microarray classification is one of the typical applications in computational biology using the machine learning approaches. It can help to identify patterns of expressed genes from which class membership for new patterns can be predicted. In this paper, we introduce SVD-Neural classifier and investigate the performance comparison with other popular training algorithms for SLFNs. In SVD-Neural classifier, the SLFN has '*tansig*' activation function and the parameters of the classifier are determined by Singular Value Decomposition (SVD) approach. This training approach is simple and requires low computational complexity.

Many non-iterative training algorithms for the single hidden layer feedforward neural networks were compared for DNA microarray classification; they were ELM, RLS-ELM and SVD approach. Also, the back-propagation algorithm which is the well-known gradient-descent based iterative training method was evaluated and compared in

terms of in terms of the number of hidden nodes and classification accuracy on test data sets. Data sets we used for this study were two binary cancer data sets of DNA microarray: leukemia and colon cancers. SVD approach, RLS-ELM and BP algorithms require the same number of hidden nodes, while ELM needs more hidden nodes. For classification accuracy, SVD-approach and RLS-ELM algorithms are comparable each other, while better than ELM and BP.

References

[1] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications", Neurocomputing, vol. 70, 2006, pp. 489-501.

[2] H.T. Huynh and Y. Won, "Small number of hidden units for ELM with two-stage linear model", IEICE Trans. on Information and Systems, vol. E91-D, no. 4, 2008, pp. 1042-1049.

[3] H.T. Huynh, Y. Won, and J.-J. Kim, "An improvement of extreme learning machine for compact single-hidden-layer feedforward neural networks", International journal of neural systems, vol. 18, no. 5, 2008, pp. 433-441.

[4] Q.-Y. Zhu, A.K. Qin, P.N. Suganthan, and G.-B. Huang, "Evolutionary Extreme Learning Machine", Pattern Recognition, vol. 38, 2005, pp. 1759-1763.

[5] H.T. Huynh and Y. Won, "Evolutionary Algorithm for Training Compact Single Hidden Layer Feedforward Neural Networks", Proc. of 2008 IEEE Int'l joint conf. on neural networks (IJCNN), HongKong, 2008, pp. 3027-3032.

[6] H.T. Huynh and Y. Won, "Training Single Hidden Layer Feedforward Neural Networks by Using Singular Value Decomposition", Proc. of Int'l Conf. on Computer Sciences and Convergence Information Technology (ICCIT2009), Korea, 2009, pp. 1300-1304.

[7] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring", Science, vol. 286, no. 5439, 1999, pp. 531-537.

[8] 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. of National Academy of Sci. of the USA, vol. 96, April 1999, pp. 6745-6750.

[9] H.T. Huynh, Y. Won, and J.-J. Kim, "DNA microarray classification with compact single hidden-layer feedforward neural networks", Proc. of the frontiers in the convergence of bioscience and information technologies (FBIT2007), 2007, pp. 193-198.

Authors

Hieu Trung Huynh received the B.S. and M.S. in Computer Engineering from HoChiMinh City University of Technology in 1998 and 2003, and Ph.D. degree in Computer Engineering from Chonnam National University in 2009. He is currently a postdoctoral researcher at Chonnam National University. From 1998 to 2005, he worked as a lecturer and researcher at the Faculty of Electrical and Electronics Engineering, HoChiMinh City University of Technology. His research interests focus on the computational intelligence for image analysis, pattern recognition, network and communication security, biological and medical data analysis.

Jung-Ja Kim received the M.S. and Ph. D. degrees in 1988 and 1997, respectively, from Computer Science at Chonnam National University in Korea. She worked with Electronics & Telecommunication Research Laboratory at Chonnam National University from 2002 to 2004, and Korea Bio-IT Foundry Center at Gwangju from 2004 to 2006. Since 2006, she has worked with Chonbuk National University as an assistant professor. Her major research interest is the bio and medical data analysis, pattern recognition, and database systems.

Yonggwan Won received the B.S. in Electronics Engineering from Hanyang University in 1987, and M.S. and Ph. D. degrees in Electrical and Computer Engineering from University of Missouri-Columbia in 1991 and 1995, respectively. He worked with Electronics, and Telecommunication Research Institute (ETRI) from 1995 to 1996, and Korea Telecomm(KT) from 1996 to 1999. He is currently a professor in Chonnam National University in Korea, and the director of Korea Bio-IT Foundry Center at Gwangju. His major research interest is the computational intelligence for image analysis, pattern recognition, computational intelligence, and bio-medical data analysis.