5

Time Course Gene Expression Classification with Time Lagged Recurrent Neural Network

Yulan Liang[1*] and Arpad Kelemen[2]

[1] Department of Biostatistics
University at Buffalo the State University of New York
Buffalo NY 14214, USA
yliang@buffalo.edu

[2] Department of Neurology, Buffalo Neuroimaging Analysis Center
The Jacobs Neurological Institute, University at Buffalo
the State University of New York, 100 High Street
Buffalo NY 14203, USA
akelemen@buffalo.edu

Summary. Heterogeneous types of gene expressions may provide a better insight into the biological role of gene interaction with the environment, disease development and drug effect at the molecular level. In this chapter for both exploring and prediction purposes a Time Lagged Recurrent Neural Network with trajectory learning is proposed for identifying and classifying the gene functional patterns from the heterogeneous nonlinear time series microarray experiments. The proposed procedures identify gene functional patterns from the dynamics of a state-trajectory learned in the heterogeneous time series and the gradient information over time. Also, the trajectory learning with Back-propagation through time algorithm can recognize gene expression patterns vary over time. This may reveal much more information about the regulatory network underlying gene expressions. The analyzed data were extracted from spotted DNA microarrays in the budding yeast expression measurements, produced by Eisen et al. The gene matrix contained 79 experiments over a variety of heterogeneous experiment conditions. The number of recognized gene patterns in our study ranged from two to ten and were divided into three cases. Optimal network architectures with different memory structures were selected based on Akaike and Bayesian information criteria using two-way factorial design. The optimal model performance was compared to other popular gene classification algorithms, such as Nearest Neighbor, Support Vector Machine, and Self-Organized Map. The reliability of the performance was verified with multiple iterated runs.

5.1 Introduction

Understanding the function of each gene in the human/animal genome is not a trivial task. Learning the gene interactions with the changing environment, with the development of a disease or under different treatment is an even greater challenge and critical to improve human life. DNA microarrays allow the measurement of expression levels for thousands of genes, perhaps all genes of a cell or an organism, within

Y. Liang and A. Kelemen: *Time Course Gene Expression Classification with Time Lagged Recurrent Neural Network*, Studies in Computational Intelligence (SCI) **94**, 149–163 (2008)
www.springerlink.com

a number of different experimental conditions [1]. As an important step, extracting the knowledge from heterogeneous types of gene expressions may provide a better insight into the biological role of gene interactions with disease development and drug effect at the molecular level. Heterogeneous types of gene expressions contain different experimental conditions. The experimental conditions may correspond to different time points under different dosages of a drug, measures from different individuals, different organs or different diseases. The dynamic patterns of genes expressed under different conditions can be useful indicators about gene state-trajectories and may reveal possible states and trajectories of disease and treatment effects [2–8]. Also, the analysis of the gene state patterns can help identifying important and reliable predictors of diseases, such as cancer, in order to develop therapies and new drugs [9]. Biologists, computer scientists and statisticians have had more than a decade of research on the use of microarrays to model gene expressions [2, 10–12]. However, most of the studies are interested in the genes that co-express in homogeneous conditions, but there are few works on heterogeneous types of gene expressions. Moreover, most of these studies focus on the mean profiles of the gene expression time course, which can make the clustering or classification of gene expressions largely simplified but ignores the important time updated (varied) information.

One feature of gene expression data in time course microarray experiment is that it includes a large number of attributes with high correlation and with high level noise. Because of its massive parallelism, potential for fault and noise tolerance, an Artificial Neural Network (ANN) based information processing is capable of taking the task to deal with this feature. ANNs can adapt their structure in response to the change of the gene expressions under different conditions in order to extract knowledge, which contributes to a deep understanding of gene interactions and identifies certain causal relationships among the genes with diseases and drugs [13–14].

The study of the heterogeneous gene expressions under different experimental conditions in a multivariate nonlinear time series may involve the study of dynamic changing of the statistical variations of non-stationary processes of gene expressions. There are several types of artificial neural networks for temporal processing, which can be used to model the natural characteristics of the gene changing under different conditions and update the information in the training data over time. Recurrent Neural Networks (RNNs) have the ability of dealing with time varying input and output and they can define neurons as states of the network [15]. The output of the hidden layer is fed back to the input layer via time delay. An internal state of the network encodes a representation of some characteristics or a biological mechanism of gene interactions, based on the transition function of the state from a recursive neural network, eventually to control the production of the internal information. State space model can be viewed as a special case of RNN, which combines a stochastic process with observation data model uniformly based on the recursive neural network. Hidden Markov processes can also be used to model the gene activity systems in which the gene states are unobservable, but can be represented by a state transition structure determined by the state parameters and the state transition matrix while processing the patterns over time. Time Lagged Recurrent Neural Networks

(TLRNNs) are extensions of conventional RNNs and outperform them in the terms of network size. A TLRNN use short memory structure instead of static topology networks to develop advanced classification systems and use a complex learning algorithm: Back-Propagation Through Time (BPTT) to learn the temporal pattern [16–17]. This dynamic learning process is well suited to the heterogeneous time series gene expression domain. TLRNNs have been used in nonlinear time series prediction, system identification and temporal pattern classification.

The goal of this chapter is to investigate the performance of heterogeneous types of multivariate time series data using time lagged recurrent neural networks with dynamic trajectory learning. The question we are interested in is whether the dynamic heterogeneous gene activity patterns can be well identified or classified through the trajectory learning with a time lagged recurrent neural network. Gene expression time series data not only exhibit very high noise level, but is also significantly non-stationary. The study of gene expressions under different experimental conditions in a multivariate nonlinear time series may involve the study of dynamic changing of the statistical variations of non-stationary processes of gene expressions. Time Lagged Recurrent Neural Networks are used to model the natural characteristics of gene changing under different conditions and update the information in the training data over time.

To deal with non-stationarity in the gene data, one approach is to build models based on a short time period or window only, such as Time Lagged Recurrent Neural Networks, which use the short memory structure to confine the input for temporal processing. Another way is to try to remove the non-stationarity using data transformation. Both approaches were performed in the application discussed in this chapter. With the presence of high level noise in the gene expression, training is difficult, and the random correlations with recent data can make the model to be based on the earlier data difficult, and it is likely to develop into an inferior model. So before building the appropriate model, data preprocessing have to be done in order to achieve desired classification and prediction performance.

In gene expression data under different experimental conditions with different time points there is a high dependence among the inputs and a high correlation among the samples, so the training is not statistically independent. One way to deal with the dependence of inputs is to include additional inputs, called lagged variables or a tapped delay in the network. Thus, one can train an ordinary network with these targets and lagged variables. Using only inputs as lagged target values are called "autoregressive models", which are widely studied statistical models. Using lagged variables means that the inputs include more than one time constant, which makes the network "dynamic" instead of using one time point (present data) with static structure. The dynamic part is called "memory structure". Such a neural network models the human brain's work in the aspect of short term memory, which essentially helps to remember the recent past events. To use lagged variables, we have to consider which lags and which input variables to include in the network, how many hidden units to use, etc. This corresponds to the design of the memory structure of the network.

The use of a recurrent neural network with time lag is important from the viewpoint of the "curse of dimensionality" and ill-conditioned problems. Trying to take into account a greater history with a Feed Forward Neural Network means increasing the number of delayed inputs, which results in an increase in the input dimension. This is called the "curse of dimensionality". If we have a small number of data points then increasing the dimensionality of the space rapidly leads to the point where the data is very sparse, in which case it provides a very poor representation of the mapping [13]. Comparing with classical time series model, TLRNNs implement Nonlinear Moving Average (NMA) models. With global feedback from the output to the hidden layer, they can be extended to Nonlinear AutoRegressive Moving Average (NARMA) models.

The rest of the chapter is divided as follows: in Section 5.2 we describe how the data was acquired and preprocessed. In Section 5.3 TLRNNs, statistical criteria for searching for the optimal model and related learning algorithms are presented. Experimental results are given in section 5.4. We survey related work in section 5.5 and finally we provide some concluding remarks in section 5.6.

5.2 Data Acquisition and Preprocessing

5.2.1 Data Extraction

The widely studied set of yeast expression measurements data, produced by Eisen et al. [18-20] contained 2465 genes. Each data point represented the ratio of expression levels of a particular gene under two different conditions: CY5 and CY3 with red and green fluorescence intensity, respectively. The gene matrix contained 79 time points over a variety of heterogeneous experimental conditions, which are important biological parameters. The data was generated from spotted arrays using samples collected at various time points during diauxic shift, mitotic cell division cycle, sporulation, temperature, reducing shocks, and so on. We extracted the data from the Stanford genome research web site (http://www-genome.stanford.edu). In our study we used two third of the data for training and the rest for testing.

5.2.2 Functional Class Extraction

If one claims to be able to predict some gene patterns or classes with certain accuracy, one should be questioned about the definition of gene patterns used, whether the patterns to be identified are biologically meaningful, and whether the biologists and pathologists actually care about them.

Classification of biological function of gene expression is essentially a classification of molecular roles for all genes and proteins. The features of gene expressions and the complexity of the genetic information make this task daunting, but it can be dealt with by ontology design, which attempts to classify and further process various aspects of molecule functions under highly qualitative and rich features of domains.

Table 5.1. Three gene functional classes and their sizes for Eisen's data

Class	Size
Ribosomal protein genes	121
Transcription protein genes	159
Secretion protein genes	96

Table 5.2. Ten gene functional classes and their sizes for Eisen's data

Class	Size
CELL CYCLE	168
CHROMATIN	48
CYTOSKELETON	72
DNA	103
mRNA	103
NUCLEAR	43
PROTEIN	477
SECRETION	116
TRANSCRIPTION	136
TRANSPORT	129

The training labels of Eisen's data were extracted from the Saccharomyces cerevisiae functional catalogue databases. There are over 145 classes of gene functional classes in the databases. Some types of the gene function classes such as cell cycle could be used to distinguish types of cancers. So once the construction of a reliable and effective classifier to learn gene functional patterns has been completed, we can predict unknown genes and identify different types of diseases.

We studied Eisen's experimental data at three levels:

- Identify two classes of gene functional patterns: 121 genes that code for ribosomal proteins and 2346 genes that code for non-ribosomal proteins.
- Identify three classes of gene functional patterns: the three classes and their sizes are listed in Table 5.1.
- Identify multiple classes of gene functional patterns: four to ten classes. Evaluate network performance when the number of gene functional patterns is increased. The selected classes and their corresponding sizes are given in Table 5.2.

5.2.3 Data Preprocessing

Figure 5.1 gives scatter plots of Eisen's data with three classes of gene expression patterns under different experimental conditions. The plots provide us some useful information of the data, e.g. there is no linear association between alpha 0 and other variables, there are some outliers and also there are possible potential clusterings, e.g. triangles are grouped together.

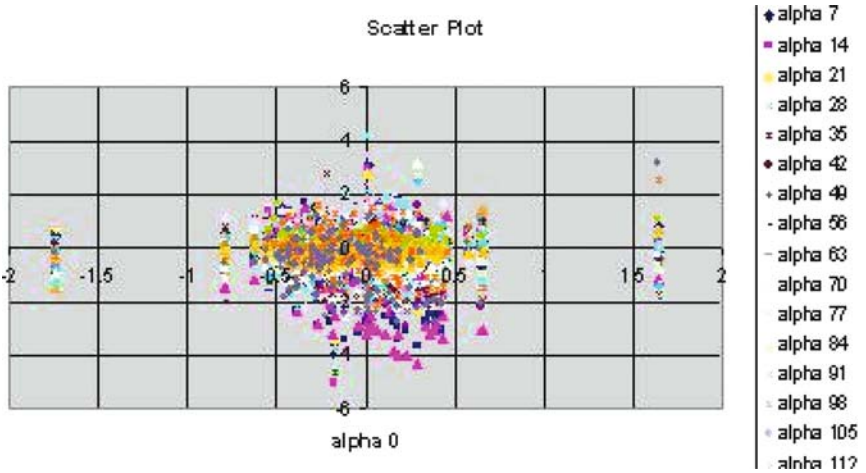

Fig. 5.1. Scatter plots of three classes of gene expression patterns (see Table 5.1) for Eisen's data (p87) under a variety of heterogeneous experimental conditions over 79 time points. Each experimental condition(time point) corresponds to one color with one shape; x-axis: experimental condition diauxic shift at alpha 0; y-axis: alpha 7 and so on.

Figure 5.2 provides time series plots for three classes of gene expression patterns under 79 experimental conditions (time points). The plot shows that the data is non-stationary, since the means and variances change with time.

5.2.4 Smoothing the Data

As it can be seen in the time series plot (Figure 5.2), the data oscillates with high frequency and high amplitude and is non-stationary, which makes direct modeling difficult. To remove these factors the raw time series data was transformed through first order difference and log compression. First order differencing reduces the non-stationarity of the time series. It can handle nondeterministic (stochastic) trends and remove the long-term trend. Log transformation can reduce the number of outliers and stabilize the variance. Figure 5.3 shows the time series plot after differencing and log transformation. As it can be seen in the figure, the transformations made the data more stationary than before transformations (Figure 5.2).

5.2.5 Input Selection

79 inputs may be too many for a Time Lagged Recurrent Neural Network, which is difficult to train, particularly if the data is noisy, and may result in overfitting problems, which do not provide good generalization. In order to select the neural network inputs, a statistical analysis has been carried out to determine the correlations between the inputs (time point) and the outputs (the class or pattern of genes).

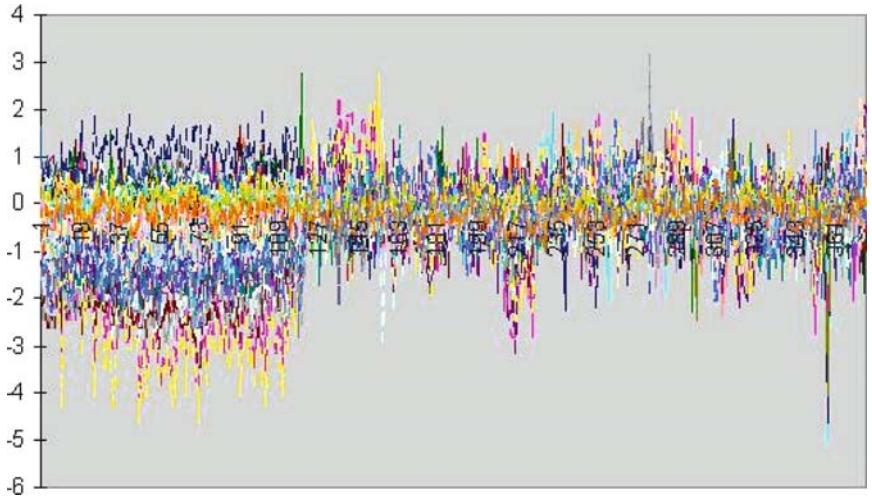

Fig. 5.2. Time series plot of three classes of functional gene expression patterns (see Table 5.1) for Eisen's data (p87) under a variety of heterogeneous experimental conditions over 79 time points. x axis: time points; y-axis: the ratio of expression levels of a particular gene under two different conditions: CY5 and CY3, respectively. Each gene time series was specified with given color, such as the gene "ORF YBL090W" was plotted with one color and so on. The means and variances of the time series are changing over time, which show that the series are nonstationary.

The Pearson correlation coefficients of inputs and outputs were computed first then the acceptance threshold was setup based on the p-values: if the p-value of the correlation coefficient was less than 0.0001, then correlation was considered and we accepted it as input, otherwise we dropped it. The selected inputs and computed correlation coefficients are given in Table 5.3. This way the number of inputs was reduced from 79 time points to 47. Several input permutation runs were also employed in order to find the combination, which produce the lowest error in the testing set. After filtering out the low correlation inputs, the data were fed into the Time Lagged Recurrent Neural Network.

When selecting input variables for a model, one must be careful not to include false predictors. A false predictor is a variable or input that is strongly correlated with an output class, but that is not available in a realistic prediction scenario. False predictors can easily sneak into a model because the process of extracting time lagged information from a database is difficult. The selection of the number of inputs is a delicate process. If the number of inputs is too small then noise makes it hard to find the true patterns in the data. On the other hand, if the number of inputs is too large then the non-stationarity of the data makes the data with statistics less relevant for the task when constructing the classifier.

One important advantage of using Time Lagged Recurrent Neural Networks is that they can use the memory function and the memory layer to confine the input,

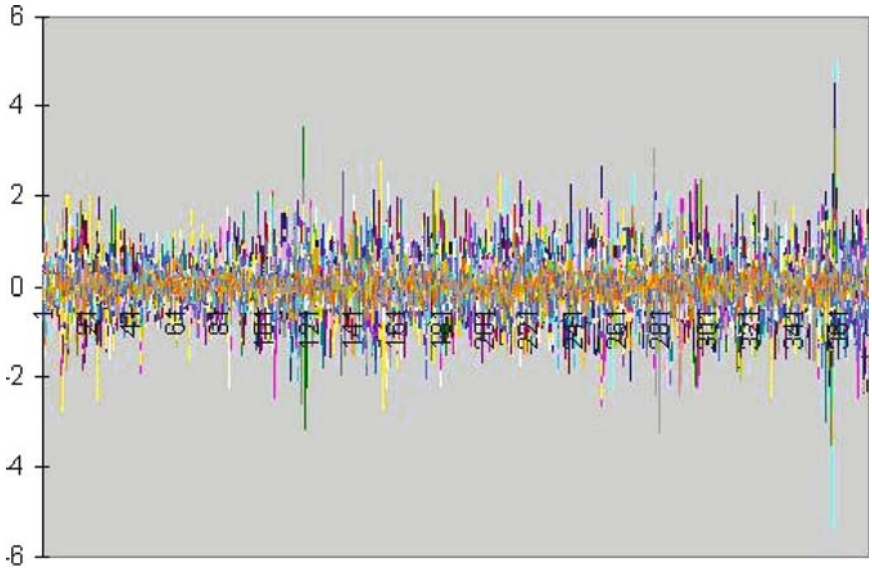

Fig. 5.3. Time series plot of three classes of functional gene expressions for Eisen's data (ribosomal, transcription, and secretion) under heterogeneous conditions after differencing and log transformation. x axis: time points; y-axis: the ratio of expression levels of a particular gene under two different conditions: CY5 and CY3, respectively. Each gene time series was specified with given color such as the gene named ORF YBL090W was plotted with one color and so on. The means and variances stay approximately constant, that indicates that the transformation made the time series closer to stationary.

which can be considered as further input preprocessors to select the inputs, and can reduce the redundant information and detect false predictors.

5.3 Design of Time Lagged Recurrent Neural Network

Time Lagged Recurrent Neural Networks are extensions of conventional Recurrent Neural Networks with short-term memory structures and local recurrent connections. We used the general network architecture with three layers and the feedback connection from the hidden layer back to the input layer. The input layer used the inputs delayed by L time points before presented to the network. Training of the TLRNN was done with Back-Propagation Through Time with trajectory learning and the parameters were learned via examples.

5.3.1 Memory Structures

There are several memory structures at the input layer to choose from. We have applied one time point delay and Gamma memory function to the data. In order to

Table 5.3. Input selection: Pearson correlation coefficients of inputs and outputs (class) for Eisen's data, Prob > |r| under H0: Rho=0

Inputs	Correlation coefficients	Inputs	Correlation coefficients
alpha0	−0.29655	cdc15210	−0.33316
alpha56	−0.26218	spo0	−0.22003
alpha63	−0.21622	spo2	0.63531
alpha70	−0.20749	spo5	0.61170
alpha84	−0.34190	spo7	0.58318
alpha91	−0.26009	spo9	0.41863
alpha98	−0.44897	spo511	−0.67660
alpha105	−0.2258	spoearly	0.67581
alpha112	−0.39606	spomid	0.67611
Elu0	0.39482	heat10	0.57206
Elu60	−0.43226	heat20	0.77191
Elu90	−0.57070	heat40	0.60434
Elu120	−0.63138	heat80	0.57094
Elu150	−0.59320	heat160	0.47246
Elu180	−0.52208	dtt15	−0.25992
Elu210	−0.48014	dtt60	0.51186
Elu240	−0.37021	dtt120	0.75011
Elu270	0.34820	cold20	0.32741
Elu300	−0.25142	cold40	0.38936
cdc1570	−0.26273	cold160	0.59074
cdc1590	−0.32061	diaua	0.33820
cdc15110	−0.40302	diauf	0.65262
cdc15130	−0.32735	diaug	0.67092
cdc15150	−0.23325		

search for the best network structure, the Akaike Information Criteria and Bayesian Information Criteria were applied. The Gamma memory function provided the lower value of AIC/BIC and the higher classification accuracy.

5.3.2 Learning Algorithms

BPTT can adapt the depth of the memory using different types of learning rules, instead of changing the number of inputs. Initial depth of the memory was setup to 10, which was later adapted by the network according to the Gamma memory function. The best learning rule for each layer for the studied data was back-propagation with gradient descent and momentum, where the momentum was setup to 0.7. As an activation function, tangent sigmoid worked best for the given data on the hidden layer and log sigmoid function on the output layer.

5.3.3 Statistical Criteria for Model Selection

The goal of model selection is to find the best network architecture that can achieve the balance between data fitting and model complexity in order to avoid overfitting and to maximize generalization performance. In a Time Lagged Recurrent Neural Network there are several dynamic parameters, such as the number of hidden neurons, the depth in the samples, and the number of trajectories in the search space that have to be optimized in order to achieve optimal model. The depth in the samples parameter can be adapted through BPTT. Two-way factorial arrays were designed to search for the best values of the trajectory and the number of hidden neurons. In this application the number of trajectories is ranged from 2 to 20 and the number of hidden nodes also ranged from 2 to 20. Statistical criteria, such as the Akaike Information Criteria and the Bayesian Information Criteria were computed in order to determine the optimal values for optimal network size and structure. We consider the best neural network to be the one with the highest classification accuracy and the lowest AIC/BIC. In case AIC and BIC don't agree we prefer BIC. The best model was chosen for the rest of the gene classification and future predictions.

5.4 Experimental Results

5.4.1 Two Classes: Ribosome and Non-Ribosome Protein Genes

After 1000 epochs of training the MSE dropped below 0.000059. The mean and the standard deviation of the correct classification rate for the testing data was $99.427\% \pm 0.366\%$ with 10 independent runs. This result is even better than the reported result by the prediction algorithm "CLEAVER", with a correct classification accuracy of 99.229834% for the same data [12].

Nearest Neighbor with Mahalanobis Distance and Self-Organized Map methods were also employed for comparison study, which gave the correct classification rates of 97.39% and 98.53%, respectively. Hierarchical Bayesian Neural Network with regularization was also employed to the same data, which provided 99.3932% correct classification rate.

5.4.2 Three Classes: Ribosomal, Transcription and Secretion Gene Functional Classes

Table 5.4 provides the computed statistical criteria for model selection. The average values of AIC and BIC of five independent runs are shown. Table 5.5 reports generalization error rates for the same runs. As it can be seen in Table 5.4 the AIC and BIC values increase rapidly and approximately linearly with the number of hidden nodes, but their values only increase slowly with the number of trajectories. The best AIC and BIC values are highlighted and they are concentrated at 2 and 4 hidden neurons and at 5 and 12 trajectories. Regarding Table 5.5 most of the low error rates are reported around the upper left corner, which corresponds to low number of hidden

Table 5.4. Factorial array for model selection with Back-Propagation Through Time and dynamic trajectory learning for Time Lagged Recurrent Neural Network for Eisen's data. Average values of 5 runs of AIC and BIC. T: number of trajectories; H: number of hidden neurons

T/H	2	4	8	12	15	20
2	1861/2143	**3300/4289**	7661/8698	11398/12970	14273/16234	19021/21629
5	**1846/2127**	3828/4367	7638/8694	11404/12978	14316/16263	19092/21699
8	1953/2234	3788/4327	7636/8692	11444/13018	14293/16254	19091/21700
10	1904/2184	3861/4400	7631/8687	11410/12985	14154/16345	19046/21653
12	**1773/2250**	3722/4582	7688/8744	11460/13033	14195/17325	19128/21735
15	1945/2225	3815/4354	7679/8735	11505/13078	14288/16250	19111/21719
18	1907/2187	3807/4346	7624/8680	11420/12992	14282/16243	19059/21666
20	1936/2217	3834/4373	7665/8721	11445/13018	14378/16339	19139/21747

Table 5.5. Generalization error rate in percentages with Time Lagged Recurrent Neural Network for Eisen's data. T: number of trajectories; H: number of hidden neurons

T/H	2	4	8	12	15	20
2	**4.42**	**3.96**	5.72	4.70	5.30	5.01
5	**3.74**	4.97	6.46	5.61	5.09	7.97
8	5.73	4.36	5.29	7.99	6.16	7.72
10	4.53	7.43	5.61	4.63	5.90	8.66
12	5.43	6.53	9.15	9.31	7.36	11.60
15	5.96	4.50	9.71	10.41	5.51	9.41
18	4.39	**3.48**	5.75	6.23	5.03	5.90
20	6.03	5.26	8.39	6.36	10.58	12.66

nodes with low number of trajectories. This is a good indication, meaning that the two tables mostly agree with each other. The optimal value from both tables, which provide the lowest generalization error rate and lowest BIC can be found at 2 hidden nodes with 5 trajectories and its value is 3.74. However, there is an even lower error rate at 4 hidden nodes and 18 trajectories, but we don't prefer it since it has high AIC/BIC. Since the number of classes to be recognized for this study is only three, it is not surprising that small number of hidden nodes and small number of trajectories can provide good performance. Results show that if we increase the number of patterns (classes) to be recognized, the number of trajectories and the number of hidden nodes have to be increased in order to get optimal performance. Table 5.5 also shows that the learning capability (generalization performance) of the model varies with the number of trajectories and the number of hidden neurons and these two may be largely determined by the complexity of the patterns to be recognized.

Table 5.6 provides results of some other popular learning approaches for gene expression classification for comparison purposes: results of Nearest Neighbor with Mahalanobis Distance (NNMD), Self Organized Map (SOM) and Support Vector Machine (SVM) are shown. Table 5.6 shows the means and standard deviations

Table 5.6. Correct classification rates with standard deviations of five runs for different methods for Eisen's data

Methods	Correct classification rate (%) ± STD (%)
NNMD	73.28 ± 0.012
SVM	74.65 ± 0.002
SOM	80.44 ± 0.053
TLRNN	**95.61 ± 0.018**
JERNN	94.04 ± 0.015

Table 5.7. Correct classification rates of Time Lagged Recurrent Neural Network with Back-Propagation Through Time and dynamic trajectory learning corresponding to the number of classes for Eisen's data

Number of patterns (classes)	Correct classification rate (%)
3	96.52
4	87.14
5	85.06
6	76.15
10	62.14

of the correct classification rates for five independent runs. SVM in this case did not provide the highest performance as opposed to most gene expression studies. The reasons may come from the heterogeneous expression data and the existence of multiple classes; TLRNN particularly performs well for this kind of time series data. We have also applied another popular recurrent neural network, the Jordan/Elman Recurrent Neural Network (JERNN) for our data set. As it can be seen in Table 5.6 the TLRNN worked best for the heterogeneous time series gene expression data.

5.4.3 Multiple Functional Classes

The data distribution for more broad gene functional classes is given in Table 5.2. The correct classification rates with TLRNN are given in Table 5.7, which are based on the optimal structure given by the AIC/BIC. As it can be seen in the table the correct classification rate decreases with the number of classes, which is not surprising. Again, as we have discussed above, both the number of hidden nodes and the number of trajectories increased as the number of classes increased in order to achieve better performance.

5.5 Related Works

A large number of approaches have been proposed, implemented and tested by computer scientists and statisticians in order to discover or identify the gene functional patterns with microarray experiments [26–27]. For example, a genetic

network approach was discussed and developed by Thieffry and Thomas [28] and D'haeseleer, et al. [10]. Time series was studied by Socci and Mitra [29] and so on. Self-organized hierarchical neural network was done by Herrero, et al. [11]. Unsupervised neural network and associated memory neural network was done by Azuaje [30] and Bicciato, et al. [31], classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks was investigated by Khan et al. [32]. Comparison of discrimination methods for the classification of tumors using gene expression data was done by Dudoit et al. [33]. We reported Bayesian neural network and regularised neural network approaches earlier [34–35]. Previous study showed that traditional statistical models can provide some insight into gene expressions and has precise results, but the weaknesses of statistical models are that they can not capture the dynamic changing of gene expressions from time to time well and are sensitive to noise and assumptions. Neural networks are more efficient and flexible for studying gene expressions. We, as an addition to our efforts reported in this chapter currently explore other kinds of neural network models for discovering correlation in gene patterns, and refine the Jordan/Elman neural network approach to study the heterogeneous time series gene expression patterns.

5.6 Conclusion

In this chapter, TLRNNs with BPTT and dynamic trajectory learning were proposed and explored in order to investigate multiple gene functional patterns with heterogeneous microarray experiments. Results show that the Time Lagged Recurrent Neural Network worked better than Nearest Neighbor with Mahalanobis Distance, Support Vector Machine and Self Organized Map. For the SVM this is a little surprise, since most well known results using SVM provided the highest performance and it has properties of dealing with high level noise and large number of attributes, which both exist in the gene expression data. The possible reasons may be found in the heterogeneous time series gene expression data and the existence of multiple classes. Another reason for the good performance of TLRNN is that it can iteratively construct the network for temporal patterns, train the weights, and update the time information. According to the results, the best generalization capability largely depends on the complexity of the patterns, which can be learned by TLRNN with BPTT and trajectory learning through monitoring the complexity of the trajectory with distinct types of states. With the increase in the number of gene functional patterns the generalization performance decreased. However, with changing the number of trajectories and the number of hidden nodes, the performance of the model can be improved based on the statistical criteria for model selection. In order to speed up the search for the best network architecture for dynamic parameters, such as the number of hidden neurons and the number of trajectories, two or three way factorial design with statistical criteria can be employed.

References

1. Gasch AP, Spellman PT, Kao CM, Carmel-Harel O, Eisen MB, Storz G, Botstein D, and Brown PO (2000) Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes. Mol. Biol. Cell 11: 4241–4257.
2. Holter N, Maritan A, Cieplak M, Fedoroff N, and Banavar J (2001) Dynamic modeling of gene expression data. PNAS, USA 98: 1693–1698.
3. Ramoni M, Sebastiani P, and Kohane I (2002) Cluster analysis of gene expression dynamics. PNAS, 99, 9121– 9126.
4. Yeung KY and Ruzzo WL (2001) Principal component analysis for clustering gene expression data. Bioinformatics, 17, 763–774.
5. Yeung KY, Medvedovic M, and Bumgarner RE (2003) Clustering gene-expression data with repeated measurements. Genome Biology, 4, R34.1–R34.17.
6. Hastiel T, Tibshirani R, Eisen MB, Alizadeh A, Levy R, Staudt L, Chan WC, Botstein D, and Brown PO (2000) Gene shaving as a method for identifying distinct sets of genes with similar expression patterns. Genome Biology 1(2): research0003.1–0003.21.
7. Romualdi C, Campanaro S, Campagna D, Celegato B, Cannata N, Toppo S, Valle G, Lanfranchi G (2003) Pattern recognition in gene expression profiling using DNA array: a comparative study of different statistical methods applied to cancer classification. Human Molecular Genetics, Vol. 12, No. 8 pp. 823–836.
8. Neal SH, Madhusmita M, Holter NS, Mitra M, Maritan A, Cieplak M, Banavar JR, and Fedoroff, VF (2000) Fundamental patterns underlying gene expression profiles: Simplicity from complexicity. PNAS, 97:8409–8414.
9. Brown MP, Grundy WN, Lin D, Cristianini N, Sugnet CW, Furey TS, Junior MA, and Haussler D (2000) Knowledge-based analysis of microarray gene expression data by using supported vector machines. PNAS. USA vol. 97(1): 262–267.
10. D'haeseleer P, Liang S, and Somogyi R (1999) Gene expression analysis and genetic networks modeling. Pacific Symposium on Biocomputing.
11. Herrero J, Valencia A, and Dopazo J (2001) A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics, 17:126–136.
12. Raychaudhuri S, Sutphin PD, Stuart JM, and Altman RB (2000) CLEAVER: A publicly available web site for supervised analysis of microarray data.
13. Bishop CM (1995) Neural Networks for Pattern Recognition. Oxford University Press.
14. Haykin S (1999) Neural Networks. Prentice Hall Upper Saddle River, New Jersey.
15. Pearlmutter BA (1995) Gradient calculation for dynamic recurrent neural networks. IEEE Transactions on Neural Networks 6(5):1212–1228.
16. Stornetta WS, Hogg T, and Huberman B (1988) A dynamic approach to temporal pattern processing, 750–759. In: Anderson DZ (Ed.) Neural information processing systems. Springer Verlag.
17. Werbos PJ (1993) Backpropagation Through Time: What it does and how to do it. Proc. of the ICNN, San Francisco, CA.
18. Chu S, DeRisi J, Eisen M, Mulholland J, Botstein D, and Brown PO (1998) The Transcriptional Program of Sporulation in Budding Yeast. Science 282, 699–705.
19. Eisen MB, Spellman PT, Brown PO, and Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. PNAS, Vol. 95, Issue 25, 14863–14868.
20. Spellman PT, Sherlock G, Zhang MW, Iyer VR, Anders K, Eisen MB, Brown PO, and Futcher B (1998) Comprehensive identification of cell cycle regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Mol. Biol. Cell 9: 3273–3297.

21. Jansen R and Gerstein M (2000) Analysis of the yeast transcriptome with structural and functional categories: characterizing highly expressed proteins. Nucleic Acids Research, Vol. 28, No. 6 1481–1488.
22. Miller DA and Zurada JM (1998) A dynamical system perspective of structural learning with forgetting. IEEE Transactions on Neural Networks, 9(3): 508–515.
23. Akaike H (1974) A new look at the statistical model identification. IEEE Transactions on Automatic Control 19:716–723.
24. Stone M (1974) Cross-validatory choice and assessment of statistical predictions (with discussion). Journal of the Royal Statistical Society, Series B, 36, 111–147.
25. Li W, Sherriff A, and Liu X (2000) Assessing risk factors of human complex diseases by Akaike and Bayesian information criteria (abstract). Am J Hum Genet 67(Suppl): S222.
26. Golub TR, Slonim, DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh, ML, Downing JR, Caliguiri MA, Bloomfield CD, and Lander ES (1999) Molecular classification of cancer: Class Discovery and class prediction by gene expression monitoring. Science, 286: 531–537.
27. Alter O, Brown PO, and Botstein D (2000) Singular value decomposition for genome-wide expression data processing and modeling. PNAS, 97, 10101–10106.
28. Thieffry D and Thomas R (1998) Qualitative analysis of gene networks. Pacific Symp. Biocomp, 3:77–88.
29. Socci ND and Mitra P (1999) Time series analysis of yeast S. cerevisiae cell cycle expression data. In Bronberg-Bauer E, De Beucklaer A, Kummer U, and Rost U (eds) Proceedings of Workshop on computation of Biochemical Pathways and Genetic Networks, Heidelberg.
30. Azuaje F (2001) Unsupervised neural network for discovery of gene expression patterns in B-cell lymphoma. Online Journal of Bioinformatics, Vol 1: 26–41.
31. Bicciato S, Pandin M, Didone G, and Bello C (2001) Analysis of an Associative Memory neural Network for Pattern Identification in Gene expression data. BIOKDD01, Workshop on Data mining in Bioinformatics.
32. Khan J, Wei JS, Ringner M, Saal LH, Ladanyi M, Westermann F, Berthold F, Schwab M, Antonescu CR, Peterson C, and Meltzer PS (2001) Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nature Med. Vol.7, No. 6. 673–679.
33. Dudoit S, Fridlyand J, and Speed T (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. JASA, 97 (457):77–87.
34. Liang, Y and Kelemen A (2005) Temporal Gene Expression Classification with Regularised Neural Network. International Journal of Bioinformatics Research and Applications, 1(4), pp. 399–413.
35. Liang, Y and Kelemen A (2004) Hierarchical Bayesian Neural Network for Gene Expression Temporal Patterns. Journal of Statistical Applications in Genetics and Molecular Biology: Vol. 3: No. 1, Article 20.