12-1991

Chemometric data analysis using artificial neural networks

Ying Liu

To the Graduate Council:

I am submitting herewith a thesis written by Ying Liu entitled "Chemometric data analysis using artificial neural networks." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Nuclear Engineering.

Belle R. Upadhyaya, Major Professor

We have read this thesis and recommend its acceptance:

R. E. Uhrig, L. Tsoukalas

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Ying Liu entitled "Chemometric Data Analysis Using Artificial Neural Networks." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Nuclear Engineering.

Belle R Upadhyaya

Belle R. Upadhyaya, Major Professor

We have read this thesis
and recommend its acceptance:

Tsou Ft

Robert E Uhrig

Accepted for the Council:

Lew Minkel

Associate Vice Chancellor
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Ying Liu

Data Nov. 25, 1991

CHEMOMETRIC DATA ANALYSIS USING ARTIFICIAL NEURAL NETWORKS

A THESIS PRESENTED FOR THE

MASTER OF SCIENCE DEGREE

THE UNIVERSITY OF TENNESSEE, KNOXVILLE

YING LIU

DECEMBER 1991

DEDICATION

This thesis is dedicated to my parents, Mrs. Xiulan Bai and Mr. Kueinian Liu.

ACKNOWLEDGMENT

I would like to thank Dr. B. R. Upadhyaya, who has enlightened me so much in directing my research. As my advisor and the chair of my graduate committee, Dr. B. R. Upadhyaya dedicated tremendous amount of time in correcting thesis drafts. I am also very grateful to him for his continuous encouragement during these two schooling years. I would also like to thank the University of Tennessee Measurement and Control Engineering Center for the financial support for this research.

My appreciation also goes to Dr. R. E. Uhrig and Dr. L. Tsoukalas for their time and patience in reading the drafts and for their helpful comments. I would also like to thank Dr. T. W. Kerlin, whose initiative made it possible for my study in the Nuclear Engineering field.

I would also like to thank Mrs. M. Gacem and Mr. M. Naghedolfeizi for their time in reviewing the thesis drafts. My gratitude also goes to Mrs. B. Campbell for her friendly help.

Finally, I would like to acknowledge my husband, Yuexin Wu, for his loving support and candid comments which have given me much motivation during the whole process.

ABSTRACT

The on-line measurement of chemical mixture composition under different operating conditions is an important problem in many industries. Effective control of industrial processes based on the measurement of mixture composition will result in a reduction in the overall energy consumption. Raman and infrared spectroscopy are often used to estimate the composition of chemical streams. The inelastic scattering of photon energy to or from the energy levels near infrared (NIR) range is called the Raman effect. The Raman spectroscopy has several advantages over the infrared spectroscopy, especially for aqueous solutions. Multiple linear regression is one of the methods currently used in spectroscopic analysis. This thesis presents a new approach which utilizes a hybrid signal preprocessing and artificial neural networks for chemometric data analysis. This method provides a general relationship between spectral signatures and percent composition of chemical samples. This approach can be easily extended to power plant applications such as lubrication oil analysis, effluent gas analysis, chemistry of reactor coolant analysis and boiler water chemistry analysis, application in the nuclear medicine field and others.

A multi-layer perceptron, with a back-propagation algorithm to train the network connection weights, was utilized in this study. Different approaches were examined to reduce the estimation error and the learning time. Preprocessing the learning data is necessary if the data contain a large number of training patterns. A Kohonen network may be invoked as a preprocessor to cluster the patterns into

different classes with associated networks. This reduces the learning time by decreasing the number of training patterns necessary to train a network. Since the architecture of the network affects learning convergence, different numbers of hidden nodes are examined in order to obtain the best network performance. Experimental results were used to explore the issue of the proper number of hidden nodes for the networks used in the present application. The network with the smallest estimation error was used to determine the optimal number of hidden nodes. The network performance was also evaluated using both one and two hidden layers. The average estimate from an ensemble of networks, that are trained with different initial conditions, was used to improve the overall estimation error. Sensitivity of the network estimation for uncertainties in the input pattern, and due to regional perturbation of spectral signatures were also studied. Analysis of network connection weights was another subject of this study. Studies were also carried out to determine the behavior of the convergence of connection weights during training. A statistical study was performed to study the distribution of connection weights.

Several spectral pre-processing approaches were used to enhance the sensitivity of composition estimation. These include spectral averaging, bias removal by differencing, and the use of different target vectors.

The results of this research and development demonstrated the feasibility of applying the neural networks technology to chemical composition analysis. Application to the estimation of power plant variables and process parameters are

also presented. A set of guidelines for developing and applying neural networks for chemometric analysis were developed as part of this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Statement of the Problem

The analysis of energy spectra in various ranges of the electro-magnetic spectrum has been used extensively in many applications. Of specific interest in process and power industries is that of composition analysis of chemical samples. On-line analysis of stream composition is important for effective control of complex chemical processes. Other industrial applications include analysis of effluent gases from power plants, air pollution monitoring, lubrication oil analysis (both for composition and wear particle sizes), and chemical control of reactor coolant and boiler water chemistry monitoring.

Infrared (IR) spectroscopy is one of the widely used techniques in analytical chemistry [1]. This involves the energy levels in the IR region of the spectrum (wavenumbers from 10,000 to 10 cm^{-1}), with the region from 10,000 to 4,000 cm^{-1} known as the near infrared (NIR). Most molecular energy levels are associated with internal vibrations of molecular structure and IR spectroscopy is the measurement of absorption or emission of photons to or from one of these states [2]. "Raman spectroscopy is the inelastic scattering of photons" about the IR energy levels. It does not involve absorption or emission from the energy levels directly, but shifts in frequencies about the energy levels. Raman spectroscopy has several advantages

over IR spectroscopy (see Chapter 2 for more discussion). The objective of the research and development reported in this thesis is to apply neural networks technology for extracting information from IR and Raman spectra of chemical samples. Some of the new features can be applied to the estimation of nuclear power plant variables and the solution of problems using neural networks.

Multiple linear regression is one of the mathematical techniques used to estimate useful properties of materials and extract the pertinent information present in Fourier-transformed Raman spectrum. This technique assumes that the properties of the material are linearly related to spectral features. Therefore, the prediction value of an unknown sample composition is mostly reliable in the linear range covered by the calibration samples [3]. The methodology developed in this thesis provides a general relationship between spectral signatures and percent composition of chemical samples, wear particle sizes in lubrication oil, and characteristics of effluent gases.

The Back-Propagation Network (BPN) algorithm [4] is currently the most widely used neural network paradigm for training multi-layer perceptrons. This algorithm was implemented in relating spectral data with sample concentrations. Multi-layer neural networks are capable of creating complex decision surfaces and have been applied to many pattern-mapping problems [5]. However, there are two critical issues in network learning: estimation error and training time. These issues may be affected by two factors: the factor related to the neural network architecture which includes the number of hidden nodes, number of hidden layers, and values of

learning parameters; the factor related to the training set are the number of training patterns, inaccuracies of the input data, and preprocessing of the data.

An extensive numerical analysis was performed in this study. This thesis will address the following issues as applied to the estimation problem:

1. Selection of the number of hidden nodes and hidden layers.
2. Modification of the network configurations.
3. Utilization of an ensemble of networks to improve network performance.
4. Pre-processing of training patterns based on the features of the data.
5. Analysis of network performance as a function of signal-to-noise ratio.
6. Analysis of the sensitivity of network estimation to variations in selective portions of the input pattern.
7. Study of the behavior of network connection weights during training.

Several Raman and NIR data were acquired from industry (DOW Chemical, 3M, AMOCO) and from the University of Tennessee Raman Laboratory. A portion of the data, in each case, was used for developing the neural network, and the remaining served as the test set.

1.2 Review of Prior Work

During the past ten years, artificial neural networks have been applied to a wide variety of industrial problems. Examples of specific areas include image processing [6], vision [7], speech synthesis and recognition [8], sonar and seismic signal classification [9], financial analysis problems, robot motion control and knowledge processing [5], signal validation [10], power plant status identification,

vibration monitoring, and others. The topic of neural computing has generated widespread interest and popularity in many areas.

The back-propagation network (BPN) algorithm has been central to most of the current study on learning in neural networks. Its inherent ability to build arbitrary nonlinear boundaries between input and output layer representations allows the back-propagation network to solve a variety of pattern mapping problems, including handwritten character recognition by Burr in 1987 [11], text reading by Sejnowsky and Rosenberg in 1987 [12], and medical diagnosis [13]. Another typical pattern recognition application of a back-propagation network is the identification of undersea targets from sonar returns [9]. The use of artificial neural networks for the estimation of chemical composition through spectroscopic data analysis is a new application area. Previously the estimation is performed by mathematical modeling which utilizes multiple linear regression or the partial least squares algorithm [3]. Recently, the interest in applying artificial neural networks in this area has been steadily increasing [14].

The features of back-propagation algorithm are found to be quite appropriate for the analysis of chemometric data, though certain signal pre-processing is often necessary. Thus, it is used as the estimation tool for chemical composition analysis in the present research.

1.3 Development of the Methodology

1.3.1 Optimal Structure of Neural Networks

Neural networks with back-propagation training are good for pattern recognition, signal prediction, complex mapping, and many other tasks. However, the network development often requires long training times. The network configuration is one of the major factors influencing its performance. In the back-propagation network learning phase, the computational complexity is significantly dependent on the number of hidden nodes. The number of hidden nodes must be large enough to create adequately complex decision boundaries which are required by a given problem. With too many hidden nodes, the decision boundaries may perfectly cover the training patterns, however, the generalization ability of the network is decreased. Also training of an excessively large number of synaptic weights may be computationally costly. Therefore, choosing an "optimal" number of hidden nodes is highly desirable. In this study, different number of hidden nodes are examined in order to obtain the best network performance. In the present applications experimental results were used to select the optimal number of hidden nodes. The appropriate number of nodes in the hidden layer was selected such that the overall estimation error is a minimum among the networks with different number of hidden-layer nodes. The number of hidden layers and size of target vector also affect network learning. Therefore, the network performance was evaluated using one and two hidden layers, as well as various choices of output vector presentations.

Combining two or more elements of the output target vector is another technique used to improve network performance.

1.3.2 Ensemble of Networks

A method of "Network Ensembles" [15] was applied to further reduce the estimation error of the network. Instead of a single network, several networks with different initial conditions were developed. The network output may be estimated using the ensemble average of the outputs of all these networks.

1.3.3 Inaccurate Input Patterns and Sensitivity Analysis

Different levels of Gaussian noise were generated to study the neural network performance in a noisy (uncertain) environment. Different noise levels were added to one of the patterns used for training the network. The resultant signals were used to test the networks. Two types of tests were performed in this study. The first utilized noise containing different percentages of individual amplitudes of the input spectrum. The other used noise computed from the average amplitude of the input spectrum. The standard deviation of error was then calculated for each resultant noise-corrupted input pattern. The results of this study demonstrated the robustness of the network in the presence of noise in input patterns.

Sensitivity analysis was performed for the purpose of determining the relationship between a selected region of a given spectrum and one or more network outputs. Three different methods were used in this sensitivity study: (1) adding

random noise to a selected spectral region (for example, one of the peaks) of input patterns, (2) removing all but the selected region of interest, and (3) removing only the selected region of input patterns. The modified input patterns were then used to test the performance of the network.

1.3.4 Selection of Training Patterns

The size of training data set is another factor which affects network learning. When the number of training patterns is large, the network learning time increases dramatically. Utilization of a two-stage procedure is one way to overcome this problem. The purpose of using the two-stage network is to reduce the number of training patterns necessary to train the networks. To accomplish this, first a pattern classification was made to organize the training patterns into different classes. The Kohonen self-organization network [5] was implemented as the first stage of the network to perform the classification. Then a separate multi-layer network was developed as the second stage, to train each class of data. Another technique, called the combined subset training which can be used to reduce the network training time, was also studied in this research.

1.3.5 Information Preprocessing

In order to increase the sensitivity of sample composition to spectral features, it is often necessary to preprocess these features. Several approaches of information preprocessing are discussed in this thesis.

1.4 Contributions of the Thesis

The major accomplishment of this research project is the design and implementation of a neural network algorithm as an on-line chemometric data analysis tool. The following tasks were completed to accomplish the goal of this thesis:

1. Development of an optimal network based on experimental evaluation of the number of hidden layer nodes in a multi-layer perceptron.
2. Study of network performance to changing signal-to-noise ratio of input spectra.
3. Study of the sensitivity of sample composition to change in certain pre-defined spectral regions.
4. Study of the convergence of network connection weights during training.
5. The use of an ensemble of networks to obtain an improved estimation of percent composition of chemical samples.
6. Appropriate preprocessing of both input and output information to achieve maximum sensitivity in estimating desired properties.
7. Demonstration of the methodology with applications to data from Measurement and Control Engineering Center member companies, and data from nuclear medicine field.
8. Development of a set of guidelines for implementing this approach for processing near infrared and Raman spectra.

1.5 Outline of the Thesis

This thesis is organized in eight chapters. Chapter 1 introduces the problem related to the present application and the methods used in this analysis. The neural network approach and the features of chemometric data are described in Chapter 2.

The methodology that is utilized for on-line chemometric data analysis is discussed in Chapter 3. The back-propagation algorithm and the principle of Kohonen self-organization network are also presented in Chapter 3. Extensive numerical studies related to the issue of optimizing the back-propagation network architecture are described in Chapter 4. The optimal network structure for estimation of chemical composition is discussed in this chapter. A two-stage architecture using the Kohonen mapping and a multi-layer perceptron is also introduced in Chapter 4. Chapter 5 evaluates the effects of input noise on network estimation and the sensitivity of the network for the selected regions of the input spectra. The behavior of the network connection weights is also described in Chapter 5. Chapter 6 presents the experimental results and a detailed analysis of composition estimation using spectral data. The discussion of this approach to similar problems in the nuclear industry such as lube oil analysis, chemical control of reactor coolant as well as application to nuclear medicine field are described in Chapter 7. Conclusions and recommendations for future work are given in Chapter 8.

CHAPTER 2

GENERAL NETWORK ARCHITECTURE AND DATA STRUCTURE

2.1 Scope of the Current Application

2.1.1 Basic Principles of Raman Spectroscopy

Infrared (IR) spectroscopy and Raman spectroscopy are used extensively in analytical chemistry. "Infrared spectroscopy involves the actual photon energy levels that fall in the IR region of the spectrum [16]." The IR region is characterized by wave numbers in the range from 10,000 to 10 cm^{-1}. The near infrared (IR) spectrum (10,000 to 4,000 cm^{-1}) is commonly used in chemometric analysis. The energy levels are associated with internal vibration of molecules. Raman spectroscopy is concerned with the inelastic scattering of photons. It explores energy by examining the frequencies present in the light scattered by molecules and "does not involve absorption or emission from the energy levels directly, but rather implicates intermediate virtual states [16]." The Raman spectrum appears as a shift in the frequency of light scattered by the molecule. It is centered around the IR frequency and is often very low in energy.

Monochromatic radiation from a laser may be used to excite a chemical sample which scatters some of the light elastically. Raman scattering occurs symmetrically to higher and lower frequencies. "When a monochromatic incident

beam passes through the sample, some of the photons collide with the molecules, give up some of their energy, and emerge with a lower energy in a different direction, therefore, a lower frequency. Other photons may collect energy from the molecules, and emerge as higher frequency radiation. By analyzing the frequencies present in the scattered radiation, the energy levels of the molecules can be deduced". Figure 2.1 illustrates a typical Raman spectrum.

"Since Raman scattering from molecular vibrations can be measured in the visible region of the spectrum, the optics of the instrument are relatively simple. Sensitive detectors with high signal-to-noise ratio are available. Therefore, the use of an intense monochromatic light source, such as a laser, can overcome the intrinsic weakness of the Raman effect. Another advantage of Raman spectroscopy is that the entire spectrum is obtained with the same instrument and cell, giving more information in a short time [1]."

Raman spectroscopy is found to be highly suitable as a composition analyzer in distillation column process control. Raman spectroscopy has certain advantages over IR spectroscopy, both in sample handling and instrumentation. Because water is not Raman active, this technique is highly suitable for aqueous solutions. Its principal disadvantage is with highly colored or fluorescing materials.

2.1.2 On-line Chemical Composition Analysis

The energy consumed in distillation processes in the U. S. represents nearly 3% of the total energy consumption. It has been estimated that effective control of

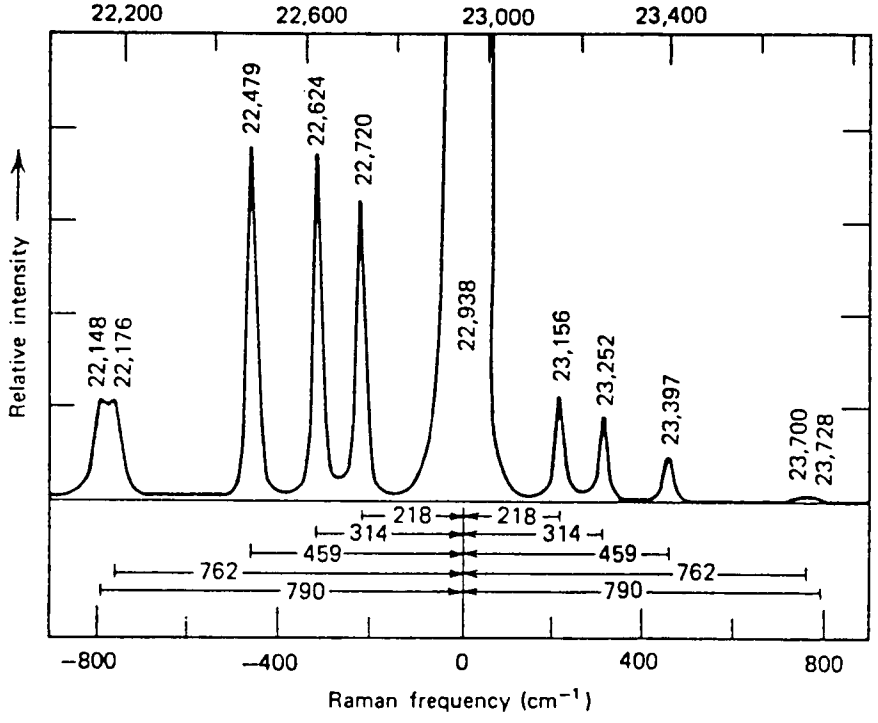


Figure 2.1. A Typical Raman Spectrum

distillation columns can reduce the cost of chemical products. As a result of industrial and government interests, the Measurement and Control Engineering Center at the University of Tennessee is jointly sponsoring with the U. S. Department of Energy, a research effort to develop a prototype on-line system to measure distillation column composition.

Raman spectroscopy is chosen as the detection technology for on-line chemical composition analysis. The feasibility of Raman spectroscopy for analysis of distillation mixture is judged based on the sufficient accuracy of chemometric composition prediction by analyzing different concentration of representative distillation mixtures [1]. The methodology developed in this research can be easily applied to the analysis of spectrum for characterizing lubrication oil and wear particle sizes, chemical analysis of boiler water , analysis of effluent gases from power plants, and many others.

The optical diagram of a Fourier transform (FT)-Raman interferometer, used to generate the spectral data, is shown in Figure 2.2 [1]. The sample is excited by a Neodymium (Nd):Yttrium-Aluminum-Garnet (YAG) laser which has several filters mounted in front of the output coupler to remove stray light from the pump lamps. This sample is illuminated by a collection lens with beam deflection mirror. The essential component of this system is beamsplitter which is used to split a beam of radiation into two and then recombine the two after introducing a path difference. The beamsplitter has approximately equal reflectance and transmittance. Therefore, When a beam from the deflection mirror reaches the beamsplitter, it is divided into

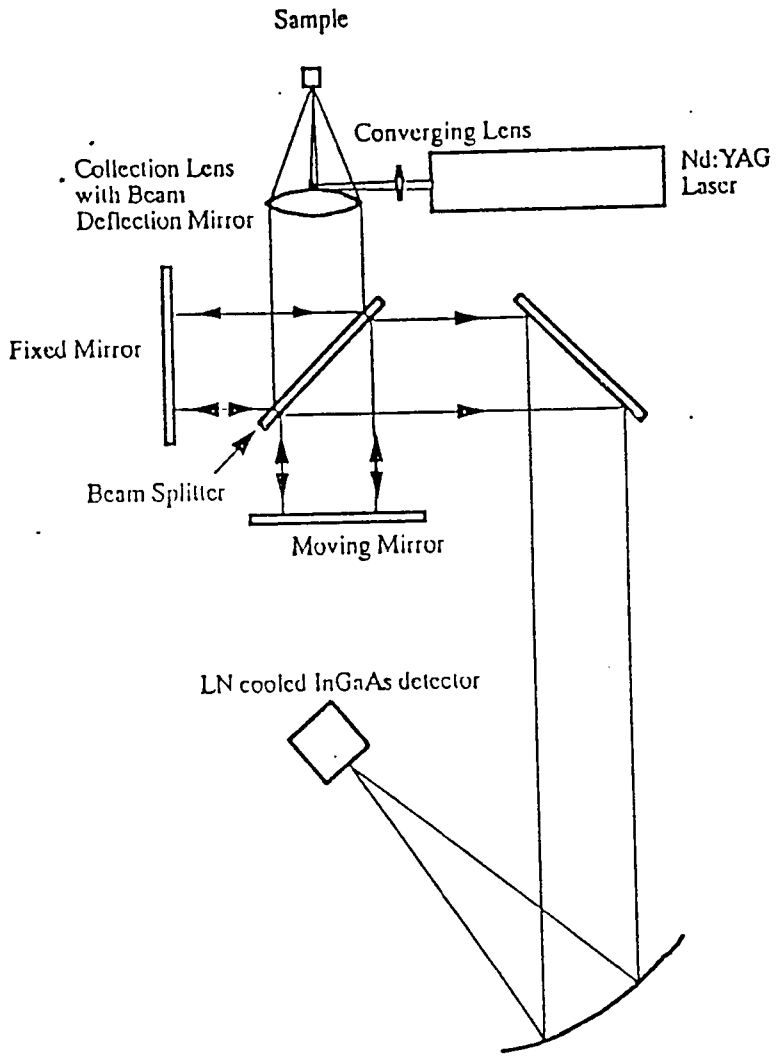


Figure 2.2. A Fourier-Transformed (FT) Raman Detector System [1]

two parts: about 50% is transmitted to the moving mirror; the other 50% is reflected to the fixed mirror. The moving mirror is used to introduce a varying path difference. As the beams from the fixed and moving mirrors are recombined, an interference pattern is obtained as the path difference is varied. The interferogram is measured by recording the detector signal as a function of the path difference between the two beams. The measurement has to start on one side of the point of zero path difference and continue out on the other side to a maximum path difference. The spectrum showing energy as a function of frequency (cm^{-1}) can be obtained from the interferogram by Fourier transformation. In this research, a liquid nitrogen (LN) cooled Indium-Gallium-Arsenide (InGaAs) detector was used to generate the interferogram.

Different methodologies are applied to quantify chemical compositions using spectroscopic data. Multiple Linear Regression methods are widely used in many applications. This method can be used to extract the wealth of information present in the Fourier transform infrared spectrum of a material and to estimate properties of the material. In this research, the artificial neural networks technology is utilized to estimate chemical composition from spectral data.

2.2 Features of Chemometric Data

A chemical sample containing n-hexane, iso-octane, toluene, p-xylene and decane was used in the basic study of Raman spectroscopy data. Figure 2.3 shows a Raman spectrum of a five compound mixture with 10.38% n-hexane, 9.98% iso-

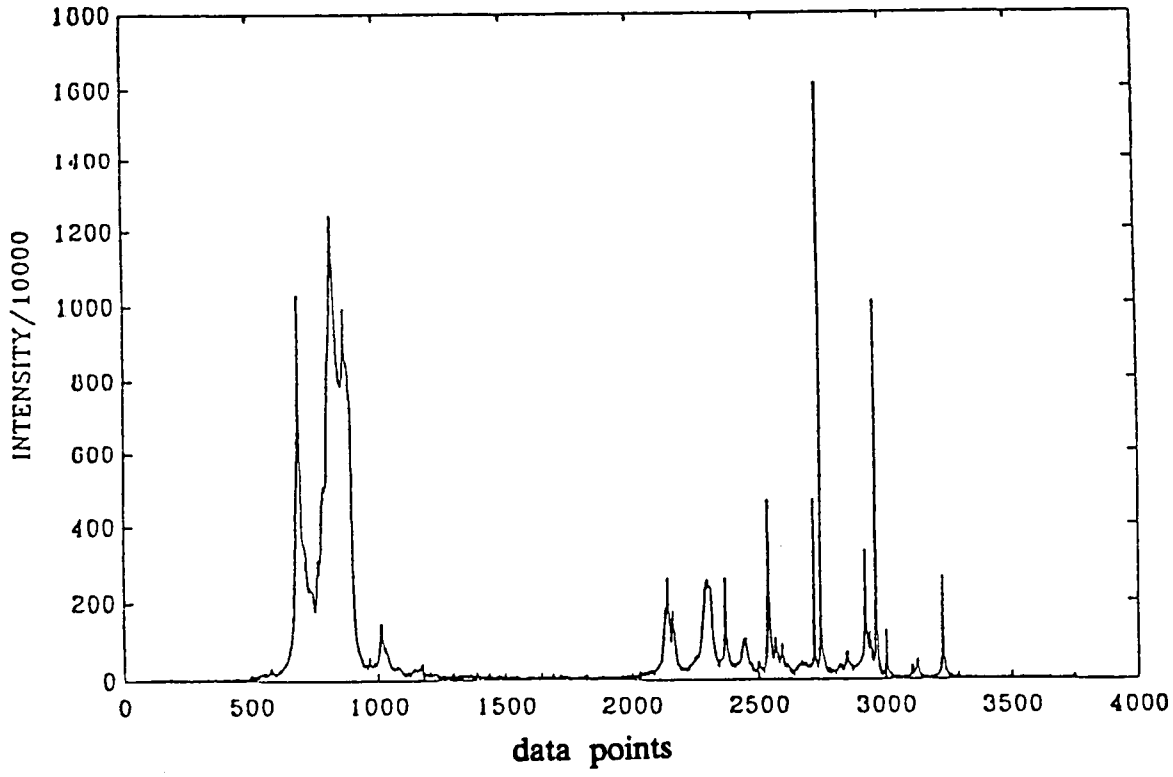


Figure 2.3. A Raman Spectrum of a Sample with Five Components

octane, 45.77% toluene, 12.95% p-xylene, and 20.92% decane. Note that each of the spectral signatures in the data base contains 3761 points (wave numbers). A heuristic analysis of the signatures shows peaks that lie in certain ranges of the spectrum. There are mainly two active ranges (see Figure 2.3), one is located between 500-1100 points, the other is between 2000-3300 points. The rest are mostly noise signals and do not directly affect the estimation of percent composition. The Raman spectra which contain a single component are presented in Figures 2.4 through 2.7.

It is important to note that specific peaks in the spectra have direct relationship to specific chemical components. For example, the occurrence of the peaks located in the range 500 to 1000 points in Figures 2.3 and 2.4 indicate the existence of n-hexane component. The spectral feature from 2700-2800 points is due to the presence of toluene. The relationship between the chemical components and the spectral range may not be always one-to-one. There are often overlaps among the various spectral features, thus reducing spectral sensitivity to sample composition. The known features of a Raman spectrum will be very useful while performing data preprocessing and in verifying the capabilities of the methodology in real applications.

2.3 Artificial Neural Networks for Chemometric Data Analysis

2.3.1 General Neural Networks Approach

Artificial neural networks have been successfully applied in many areas during the past few years. In this research, neural networks methodology was applied to model Raman and IR spectra of chemical mixtures to produce quantitative estimation

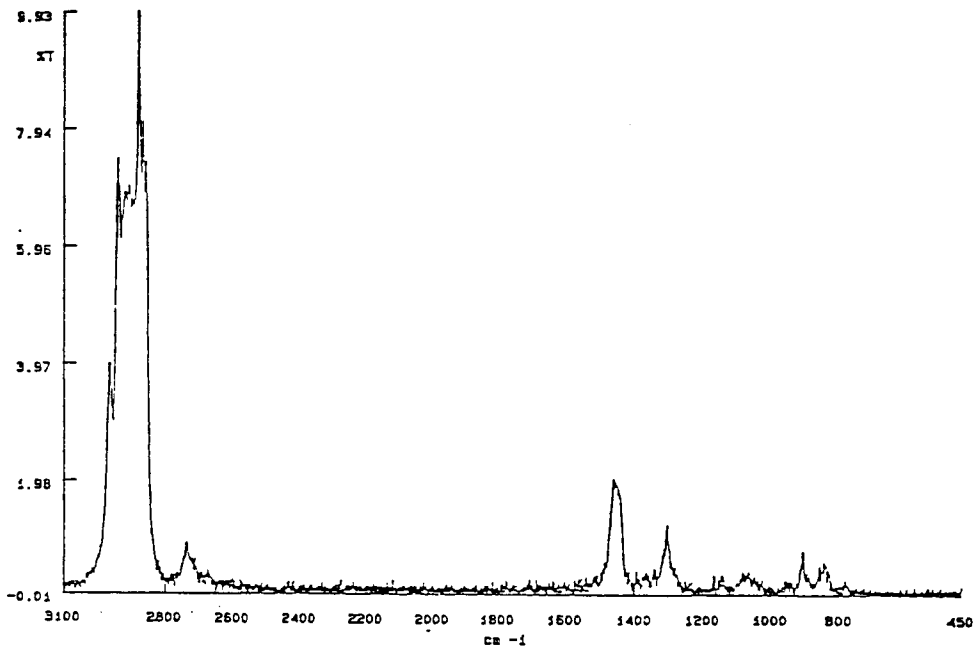


Figure 2.4. A Raman Spectrum of a Sample Containing only N-HEXANE

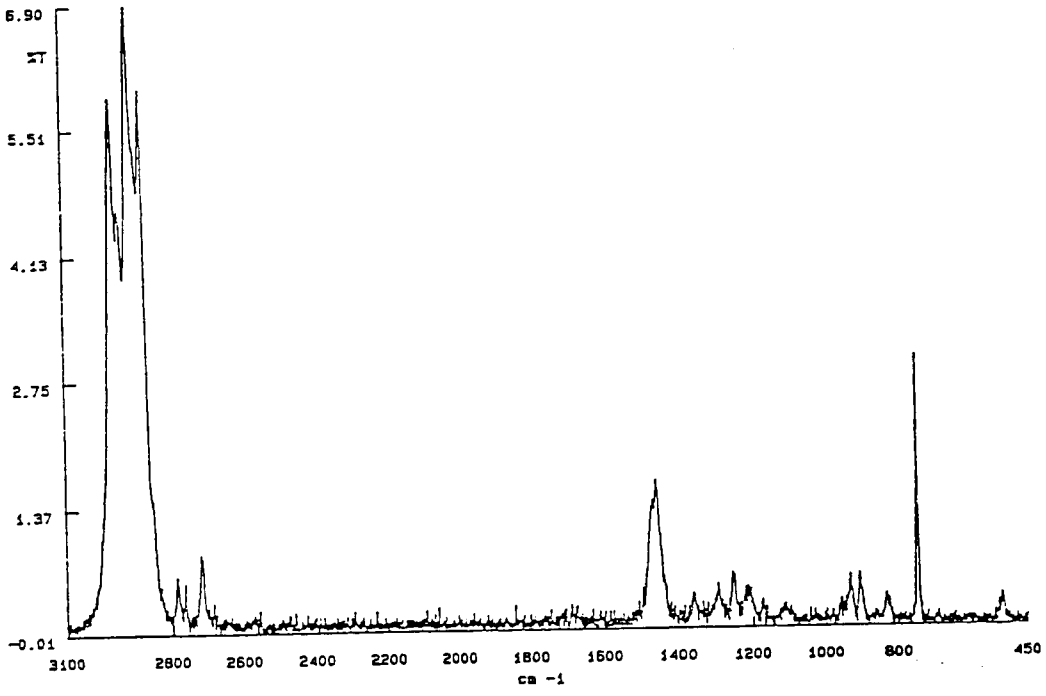


Figure 2.5. A Raman Spectrum of a Sample Containing only ISO-OCTANE

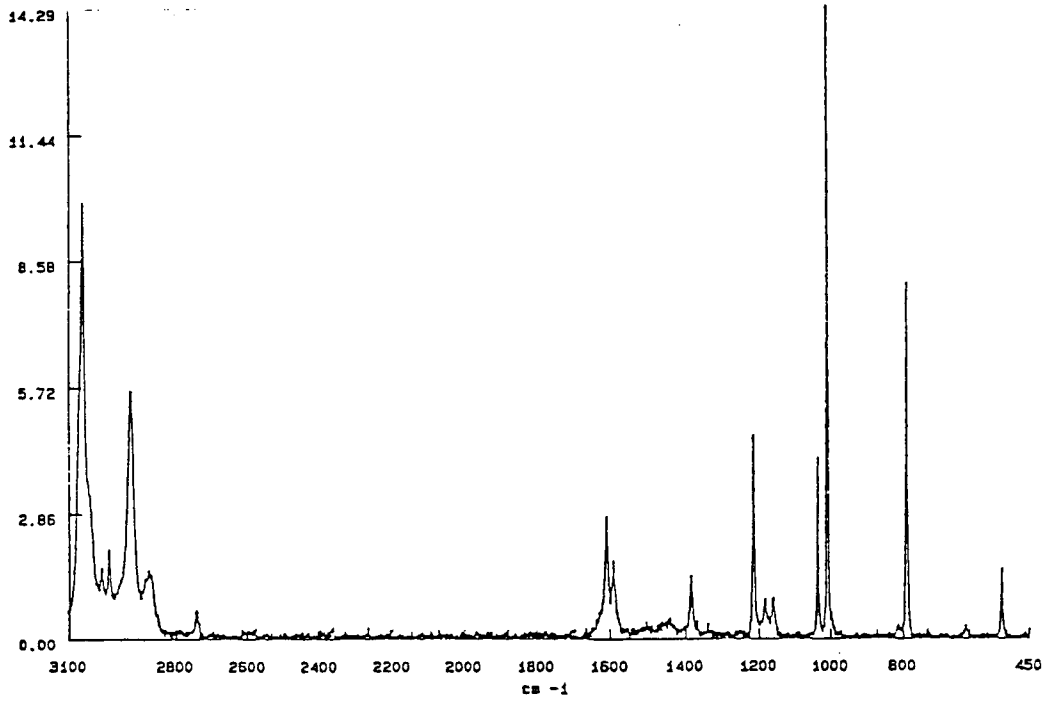


Figure 2.6. A Raman Spectrum of a Sample Containing only TOLUENE

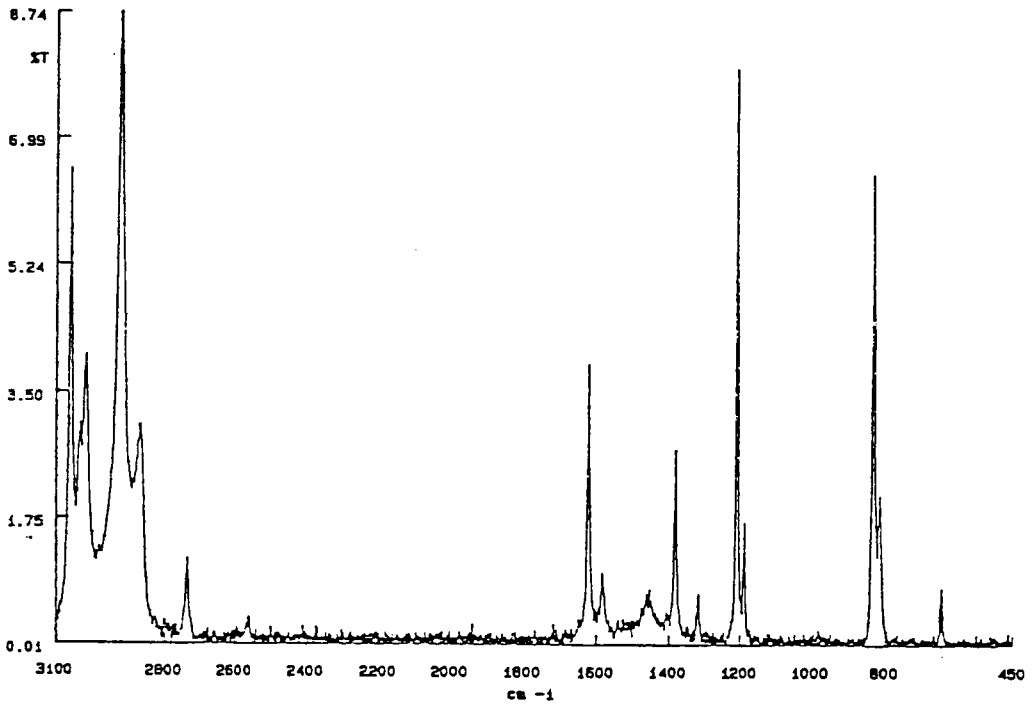


Figure 2.7. A Raman Spectrum of a Sample Containing only P-XYLENE

of concentration of chemical components. Spectra from samples containing five components (n-hexane, iso-octane, p-xylene, toluene, and decane) were used in the general study presented here. The multi-layer feedforward network, trained with a back-propagation algorithm, was chosen to estimate the concentration of chemical components in the mixture using Raman spectral signatures. The feasibility of using back-propagation network for this application was studied based on the following observations:

1. Artificial neural networks are very effective in relating quantities for which a physical or empirical model is not fully described.
2. The given data set requires a supervised learning model.
3. Multi-layer feedforward neural network is the most suitable network for generating nonlinear relationship for a given problem.
4. The pattern-mapping is the most suitable problem to model using back-propagation networks
5. A preliminary test using the back-propagation network showed very encouraging estimation results.

A typical multi-layer neural network model is shown in Figure 2.8. The input to the network is a set of selected intensities in the Raman spectrum, and the output of the network is a vector of percent concentration of chemical components. The neural network was first trained to learn the functional relationship between the input and output signals by presenting a sequence of inputs and a set of expected outputs. In this application, the training data pair was composed of a sequence of intensities of a Raman spectrum and fractional concentrations of the five components. Once the network has been trained, it can be used to estimate the concentration by

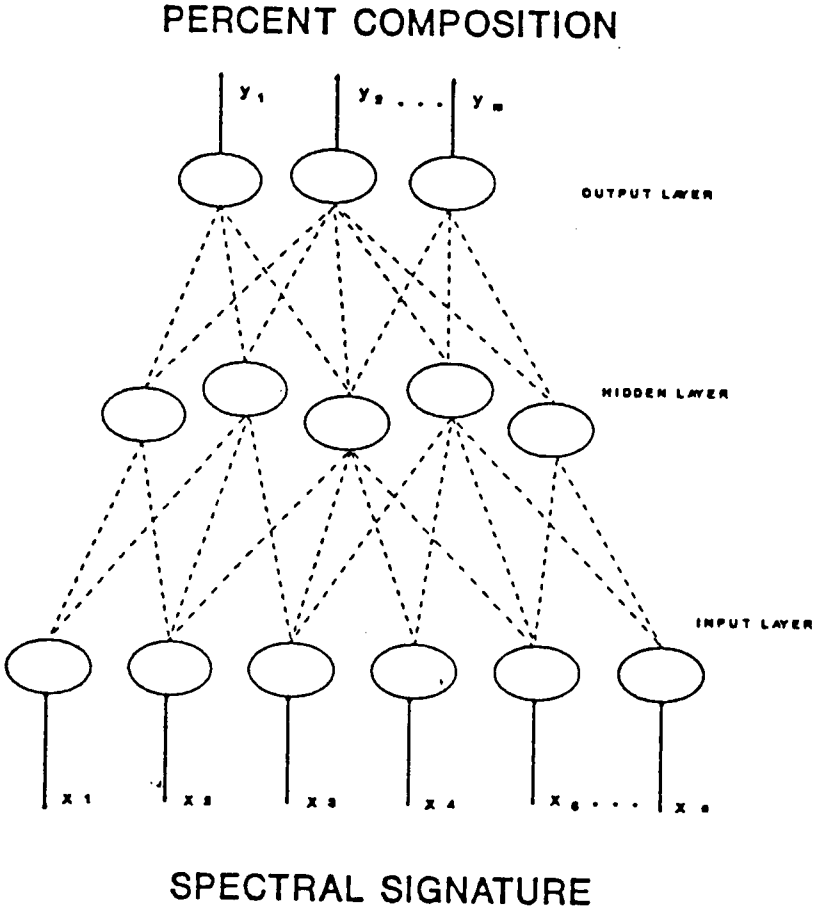


Figure 2.8. A Diagram of Neural Network System

presenting any "unknown" Raman spectral data (intensities). In order to build a good model that would interpolate parameter values in the future, it is necessary that the training data span the domain of interest completely.

2.3.2 Data Preprocessing

Data preprocessing is very important in this analysis. Some of the preprocessing was made through iterative trials, whereas other methods were used based on consultation with analytical chemists (at industry and universities). The spectra shown in Figures (2.3-2.7) each contains 3,761 points. If all the data points were to be used, this would result in a huge neural network (from the standpoint of memory, computational speed, and accuracy). Therefore, before training the network, the following steps must be taken to process the training data:

1. Dimensionality reduction,
2. Amplitude sensitive smoothing,
3. Data normalization.

The problem of dimensionality reduction is closely related to feature extraction. In this application, the dimensionality reduction is done by selecting the active ranges of the spectra, as described in Section 2.2. The selected ranges for the data points are located between 500 and 1200 and between 2000 and 3300 points, which makes a total of 2000 points. These points correspond to wave numbers. Processing such a big data set was still difficult, thus the selected ranges of spectra were further smoothed using an amplitude sensitive moving average method. This

method takes the average of the spectrum within a user specified window width and amplitude threshold. For the Raman spectra used in the present study, this smoothing process was performed by choosing the amplitude threshold as 100, and the window width for averaging as 10 points. By this procedure, the final training data size was reduced to 190 points. The last step in data preprocessing was to normalize the amplitudes of the 190 points within a range of [0.1, 0.9] for each spectrum. The resultant data were finally utilized as a data learning set to train the neural networks.

The application of the Probabilistic Neural Network (PNN) or the Kohonen Self-organizing Feature Maps (KSFM) to the initial set of data was another approach utilized to reduce the training data size. This data reduction can be done by classifying the entire learning patterns into several different groups. The method of preprocessing data using a second network is described in Chapter 5.

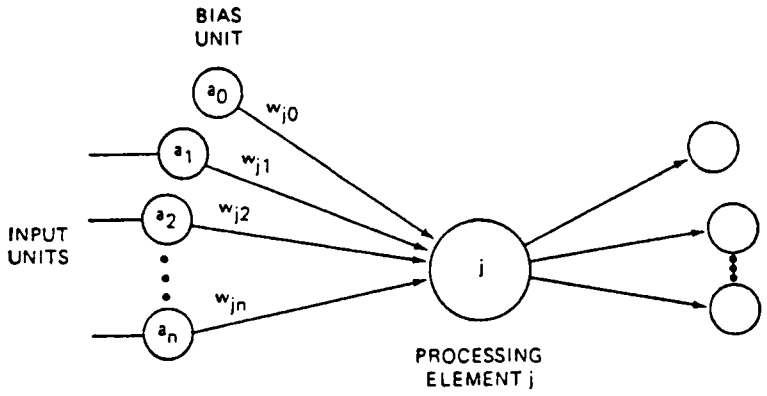
CHAPTER 3

METHODOLOGY FOR ON-LINE CHEMOMETRIC DATA ANALYSIS

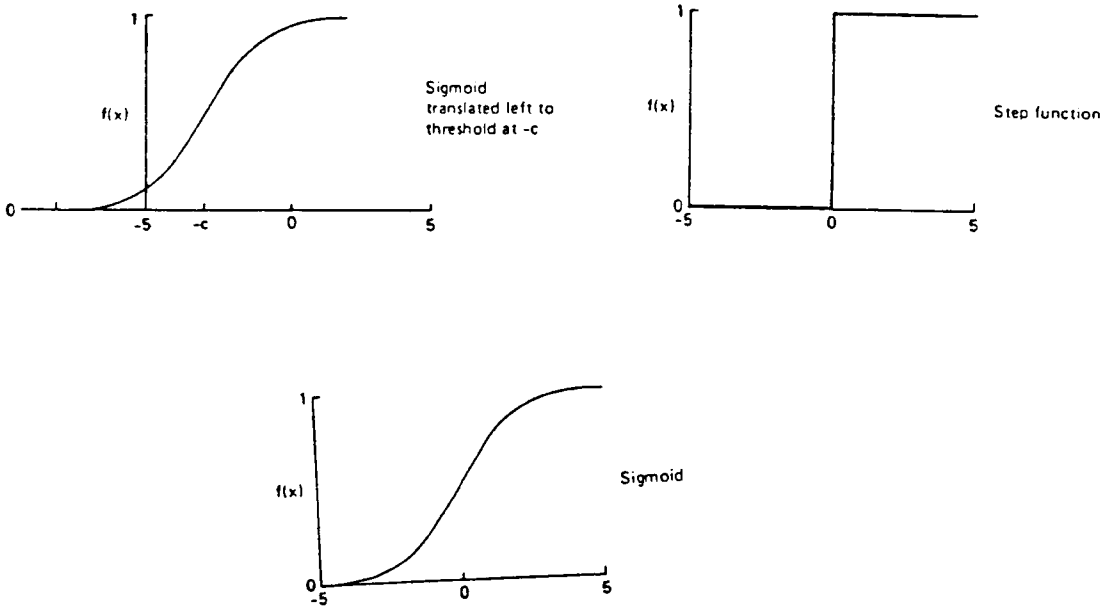
3.1 Description of Artificial Neural Networks

Artificial neural networks are developed to simulate the most elementary functions of neurons in the human brain, based on the present understanding of biological nervous systems. These network models attempt to achieve good human-like performance such as: learning from experiments and generalization from previous samples. The network models are composed of many nonlinear computational elements (nodes) that operate in a parallel distributed processing architecture. Computational elements or nodes used in neural networks are connected by links with variable weights.

The simplest computational node sums m weighted inputs (x_1, x_2, x_3, ... x_m) and passes the result through a nonlinear function as shown in Figure 3.1(a). Note that w_1, w_2, w_3, ... w_m are the connection weights. Figure 3.1(b) illustrates three common types of nonlinear functions: step, threshold logic, and sigmoid.



(a). A Typical Computational Node.



(b). Three Nonlinear Activation Functions.

Figure 3.1. A typical Computational Node and
Three Nonlinear Activation Functions.

A three-layer feedforward neural network structure is illustrated in Figure 3.2. The first layer of the network is the input layer. The function of the input layer is to receive external input vector values. The weighted input signals are transferred to the nodes above input layer, known as hidden layer nodes. Each node in a hidden layer computes the sum of its weighted inputs and transforms this sum using an activation function (for example, sigmoidal function). The outputs from the hidden layer are then sent to every node in the output layer.

The potential benefits of artificial neural networks extend beyond the high computation rates provided by massive parallelism. Neural networks have many advantages over traditional computing and modeling: no need for a specific model form, prediction from incomplete data, detection of data features, and construction of generalized mapping. Another important advantage of neural networks is that they are robust and fault tolerant. This is because neural networks encode information in a globally distributed fashion.

There is a variety of neural network architectures: Hopfield network, multi-layer perceptron with back-propagation training, Kohonen self-organizing feature map, counter-propagation network, and others. In this study, multi-layer perceptrons, with back-propagation training, are used. A detailed discussion of this algorithm is given in the following section.

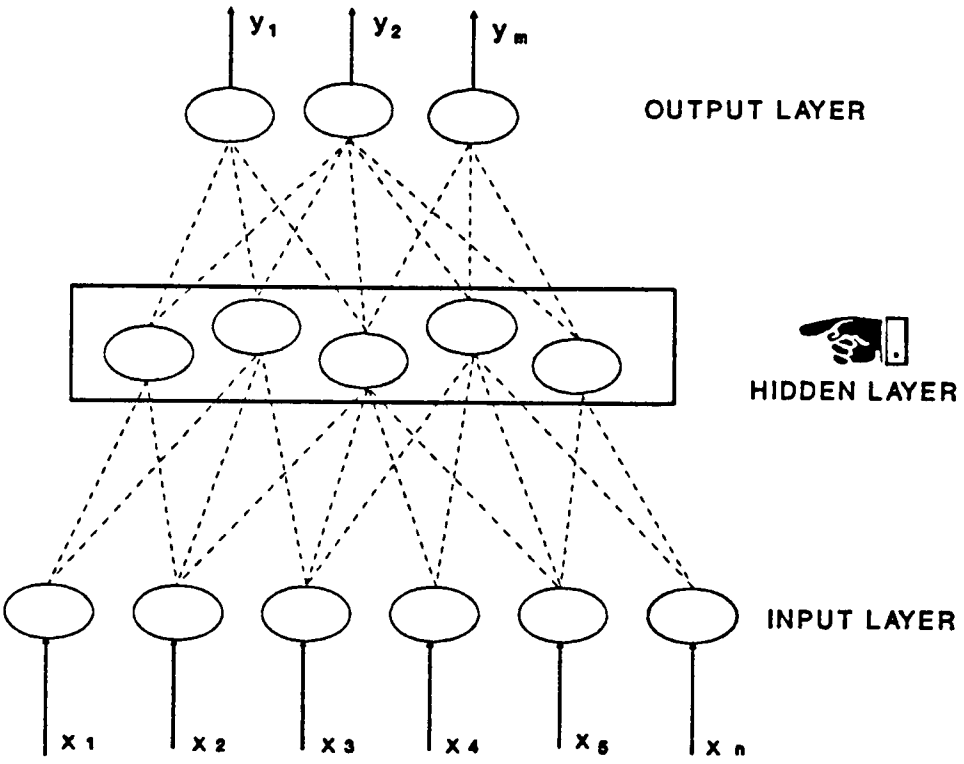


Figure 3.2. Schematic Representation of a Three-layer Feedforward Network.

3.2 The Back-Propagation Network (BPN) Algorithm

The back-propagation algorithm was first proposed by Rumelhart et al [4]. It is the most widely used systematic algorithm for supervised learning in multi-layer neural networks. The goal of the back-propagation algorithm is to teach the network to associate specific output patterns (target patterns) by adjusting the connection weights in order to minimize the error between target output and actual output of the network. A gradient descent algorithm is generally used to perform the optimization.

A back-propagation neural network is trained by supervised learning. During the learning procedure, a series of input patterns (e.g., Raman spectra) with their corresponding output values (e.g. fractional chemical concentrations) are presented to the network in an iterative fashion while the weights are adjusted.

The back-propagation learning algorithm is composed of two types of passes: the forward-propagation (forward pass) and backward-propagation (reverse pass). The forward-propagation executes the computation of network outputs layer by layer. The output of one layer serves as input to the next layer. The forward-propagation procedure for a three-layer feedforward network is shown in Figure 3.3. The notational conventions are also shown in the figure. Given an input vector X (x_1, x_2,... x_n) to the neural network, the node j in the hidden layer receives a net input as a summation of its weighted input and a node bias:

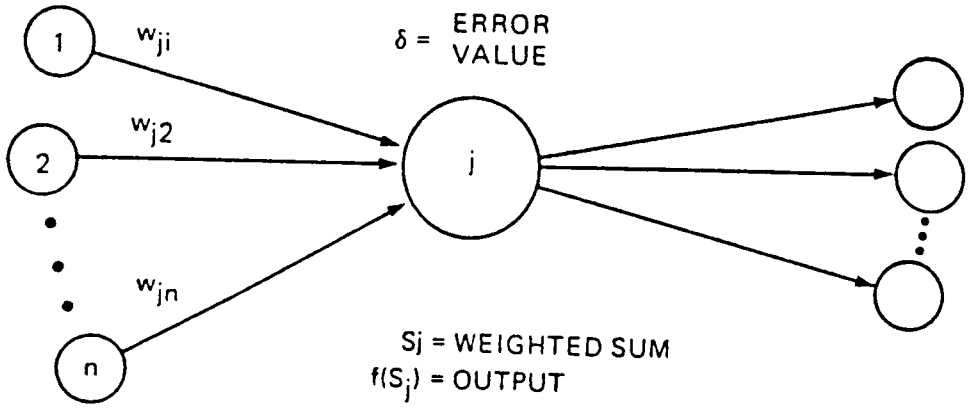


Figure 3.3. The Forward-Propagation Pass [5].

$$Net_j = \sum_{i=1}^{n} w_{ij} x_i + \theta_j \tag{3.1}$$

where

w_{ij} = the connection weight between unit i in the input layer and unit j in the hidden layer

x_i = the i-th output from the input layer node

θ_j = j-th node bias

The output of unit j is evaluated as

$$P_j = f(net_j) \tag{3.2}$$

where f is an activation function.

The net value for an output layer node k is :

$$Net_k = \sum_{j=1}^{m} W_{jk} P_j + \theta_k \tag{3.3}$$

The final output is produced as follows:

$$O_k = f(Net_k) \tag{3.4}$$

The activation function used in this network is a sigmoidal function given by

$$f(x) = \frac{1}{1+e^{-\beta x}}, \quad \beta > 0, \tag{3.5}$$

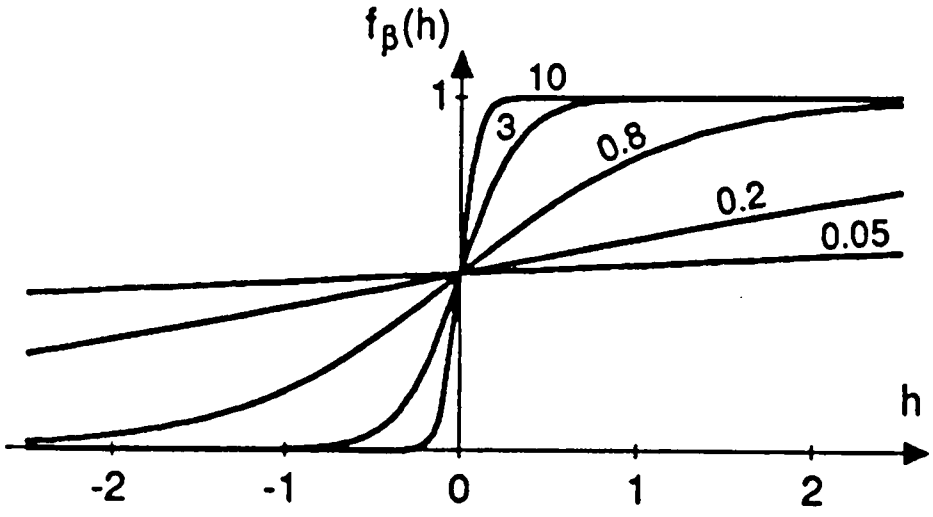
Thus, the output of unit j in the hidden layer becomes

$$P_j = \frac{1}{1+e^{-\beta(net_j+\theta_j)}} \tag{3.6}$$

where θ_j is the nodal bias.

In Equation (3.6) the parameter β describes the shape of the sigmoidal function. The effect of different values of β is illustrated in Figure 3.4. The bias for neuron j in layer p is θ_j which is used to shift the activation function along the X-axis. As the forward-propagation completes, the error between the network output and the target values are calculated.

In the back-propagation pass, the connection weights are corrected to reduce the error found after the forward propagation. This error-correction procedure is made from the output layer to the hidden layer. The Generalized Delta Rule (GDR) is utilized to adjust the interconnection weights so as to reduce the square of the error for each pattern as rapidly as possible. One important parameter in the learning phase is the error value (δ) which is associated with each processing unit. It reflects the amount of error associated with that unit. The error parameter is used during the weight-correction procedure. A large value of δ indicates that a large



$$f_\beta(h) = \frac{1}{1+e^{-\beta h}}, \quad \beta > 0$$

Figure 3.4. Plot of Sigmoidal Function with Different Values of β

correction should be made to the connection weights. The parameter δ is defined as follows.

For the nodes in the output layer ,

$$\delta_{pk} = (t_{pk} - o_{pk}) f'(net_{pk}) \tag{3.7}$$

where

t_{pk} = target output of node k

o_{pk} = actual network output of node k

p = a subscript denoting the pattern number

f' = $\partial f / \partial net$.

For the nodes in the hidden layer ,

$$\delta_{pj} = (\Sigma \delta_{pk} w_{kj}) f'(net_{pj}) \tag{3.8}$$

where w_{kj} is the connection weight between node k in the output layer and node j in the hidden layer.

The error parameter δ_{pk} can be evaluated in the highest layer of the network by using Equation (3.7). Then the error is propagated in a backward manner to the lower layers. This will allow us to calculate the parameter δ_{pj} at the hidden node in terms of the δ_{pk} at the upper layer. According to the generalized delta rule, the weight adjustments are made as follows. It is important to note that the "backward propagation" of the error is a natural consequence of the gradient descent algorithm.

$$\Delta W_{ji} = \eta \delta_j o_i \tag{3.9}$$

The parameter η is the learning rate defining the step size of training, δ_j is error parameter of upper layer node j and o_i is the active value of lower layer node i.

To improve the training time of back-propagation algorithm and enhance the stability of the learning process, a momentum term is added to Equation (3.9). Therefore, Equation (3.9) is changed to the following expression:

$$\Delta W_{ji}(n+1) = \eta \delta_j o_i + \alpha [W_{ji}(n) - W_{ji}(n-1)] \tag{3.10}$$

The second term in the above equation is the momentum term. The parameter α is the momentum coefficient which is usually initialized around 0.9. The integer (n + 1) indicates the training iteration number.

Thus, the resultant connection weight is computed as

$$W_{ji}(n+1) = W_{ji}(n) + \Delta W_{ji}(n+1) \tag{3.11}$$

An expression similar to Equation (3.10) is used to adjust the connection weights between nodes in the input layer and the hidden layer. Prior to the start of training, all the weights in the network are set to random values. Equations (3.10) and (3.11) are used to correct the connection weights in each training pattern until the error reaches an acceptable value for the entire training patterns.

The step-by-step procedure of back-propagation learning rule is summarized below.

1. Initialize the connection weights to small random values in the range [-1,1].

2. Apply a pattern to the input layer.

3. Propagate the input pattern in a forward fashion through the network using Equations (3.1-3.4) until the final network outputs are calculated.

4. Compute the error parameter δ for the nodes in the output and hidden layers using Equations (3.7) and (3.8).

5. Adjust the connection weights using the Generalized Delta Rule, Equations (3.9) and (3.10).

6. Repeat steps 2 through 5 for the next training pattern.

7. Stop training when the root mean square (RMS) error of network output reaches an acceptable level.

There are several issues that need to be considered when utilizing the back-propagation algorithm to train a neural network. For example, the optimal number of hidden layers and the corresponding number of nodes in each layer, the optimal values of the learning parameters which improve network training, and a format for presenting training data. A detailed study on some of these issues is presented in Chapter 4.

3.3 The Kohonen Self-organizing Feature Map

The objective of using a Kohonen network is to find the natural relationships among the learning patterns and then classify them into different groups. Kohonen self-organization map (KSOM) is a clustering algorithm which is used to classify an unknown pattern into a specific class based on topological properties [5]. The KSOM neural network utilizes a heuristic algorithm using a number of training patterns to search out an unknown goal. The Kohonen network provides advantages over classical pattern-recognition techniques because it utilizes the parallel architecture of a neural network and provides a graphical organization of pattern relationships.

The KSOM network is usually composed of two layers, an input layer and a competitive layer. This neural network structure is illustrated in Figure 3.5. These two layers are fully connected by the associated weight matrix $[w_{ij}]$. An unsupervised learning algorithm is used for KSOM network training. In the learning phase, when an input pattern is presented, the nodes in the input layer take on the values of the corresponding entries. Then, each node in the second layer sums its weighted inputs and competes to find a single winning unit by comparing the distance between its weight and the input vectors.

If an input pattern to the KSO network is given as

$$X = (x_1, x_2, x_3, \ldots x_n) \ ,$$

and the connection weights between neuron i in the input layer and neuron j in the

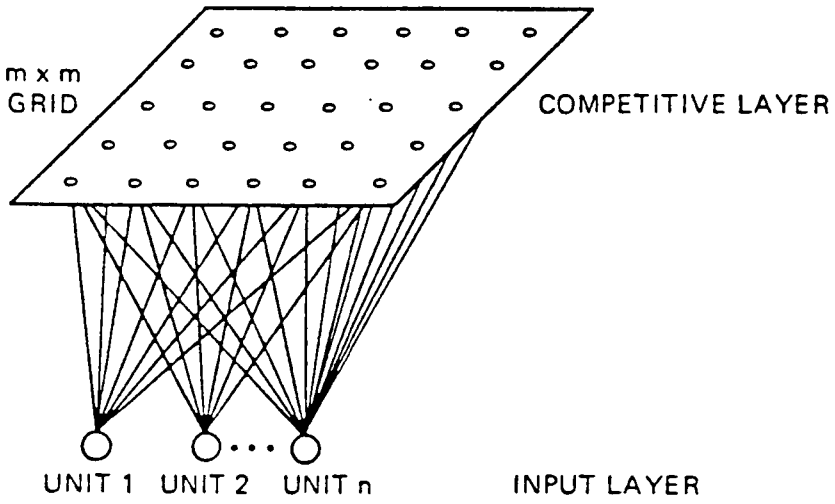


Figure 3.5. Kohonen Network Structure [5].

output layer are w_{ij} (Figure 3.5), then the distance between vector X and vector W_j is defined as

$$d_j = \|X - W_j\| = \sqrt{\sum_{i=1}^{n}(x_i - w_{ij})^2} \tag{3.12}$$

At the competitive layer (output layer) the node with the lowest value of d_j (the best match) wins the competition. Let this output node be denoted as c. Then

$$d_c = \|X - W_c\| = MIN\|X - W_j\| \tag{3.13}$$

$$\text{for } n \geq j \geq 1$$

If more than one output neurons have the same value of d, then the one with the lowest index j will win. After the winner node is found, the next step is to identify the neighborhood around it. The box presented in Figure 3.6 illustrates an example of the neighborhood region of neuron c. Weights connecting neuron c and its neighbors are modified by the following learning rule

$$\Delta w_{ij} = \alpha(x_j - w_{ij}) \tag{3.14}$$

$$w_{ij}^{new} = w_{ij}^{old} + \Delta w_{ij} \tag{3.15}$$

$$\text{for } j \leq N_{jc}(k) \text{ and } 1 \leq i \leq n.$$

There are two parameters in the above equation. N_{jc} defines the size of neighborhood about neuron c; it is initially set to a half or a third of the width of the competitive layer of process nodes, and it decreases during the training procedure.

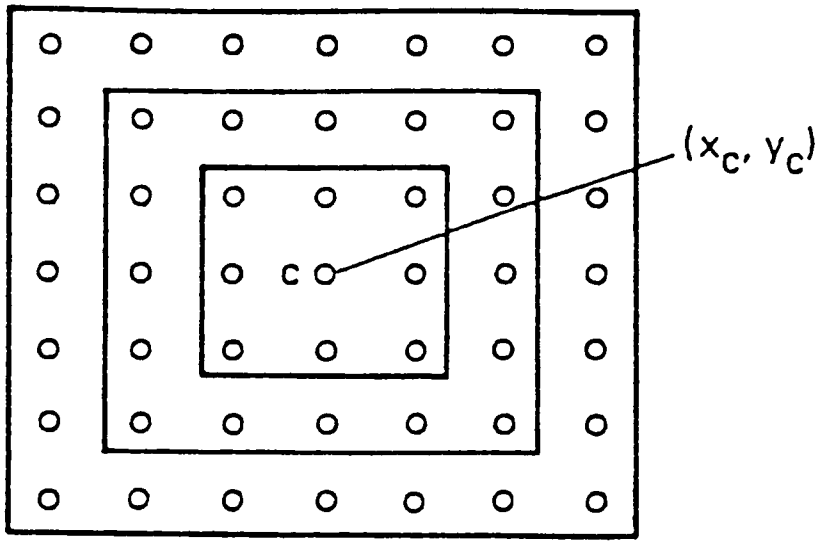


Figure 3.6. Neighborhood of Node C in the Kohonen Network.

α is the learning rate, it is initially set to a large number denoted as α_0 and decreases during the learning phase.

$$\alpha_k = \alpha_0(1 - \frac{K}{T}) \tag{3.16}$$

where

T = the total number of iterations

K = the current iteration number

Thus, α begins at α_0, and finally decreases to zero when T training iterations are completed.

After the training is completed, pattern relationships and groupings are observed from the competitive layer. Therefore, when a new input pattern is presented to the trained network, the best match node in the output layer will be identified, and its corresponding group will be matched.

In summary, the learning algorithm of Kohonen self-organizing feature map is as follows.

1. Initialize the network connection weights to small random values.
2. Present an input pattern, and find the best match node in the competitive layer by using Equation (3.12) to compute the distance between input and associated weight vectors.
3. Adjust this node and its neighboring weights using Equations (3.14) and (3.15).
4. Gradually decrease the size of the neighborhood and the amount of change in the weight during the learning phase.

5. Go to step 2 and repeat the process until all training patterns are included.

3.4 Advantages and Limitations

Artificial neural networks have proved their capability and demonstrated their advantages in solving a variety of problems. They are robust and fault tolerant and are able to process incomplete and noisy data. Neural networks can operate at a high speed via massive parallelism, and they are able to learn and generalize from numerical samples of data. Neural network techniques are well-suited for processing complex data information. However, most of the capability of neural networks depends on a large number of design choices in their implementation. In the back-propagation algorithm, the choice of the number of hidden layers and nodes, learning parameters, and training data representation are the key issues in establishing the best network performance. Consequently, an optimal network structure is desirable in overcoming the drawbacks of a large amount of learning time and high estimation errors. There is no general rule to determine the size of network hidden nodes. In most cases, it is defined experimentally. In the next chapter, an approach developed in this thesis for selecting the optimal network architecture is discussed.

CHAPTER 4

OPTIMAL NETWORK STRUCTURE FOR CHEMOMETRIC DATA ANALYSIS

4.1 Introduction

Artificial neural networks are extremely attractive in solving real-world problems. However, in practice, there are many questions to be answered prior to implementing a neural network. A fundamental question is "what is the best network architecture for a given problem ?" Answers to this question are focused on selecting the number of layers and nodes per layer in a multi-layer perceptron. To obtain a good generalization one has to build into a network as much knowledge about the problem as possible and limit the number of internal connections in the network appropriately. Therefore it is desirable to find a method that not only optimizes the connection weights for the given architecture but also optimizes the architecture itself. With the optimal neural network structure, one can decrease both the network learning time and the estimation error. In this chapter, different approaches of designing the network structure are reviewed and the method used in the present research is described. These studies include selection of the number of hidden nodes, the number of hidden layers, the number of output nodes, utilization of ensemble networks and a two-stage network approach for the general solution.

4.2 Selection of Number of Hidden Nodes

In the multi-layer network structure, the hidden layer is a critical part in the network's learning phase. With appropriate number of hidden nodes, a multi-layer perceptron is capable of realizing arbitrary continuous functions defined on a hypercube [17]. Thus finding the right number of hidden nodes is the most challenging task in designing a back-propagation network structure.

The hidden nodes (also called feature detectors) play an important role in network learning. With too few hidden nodes, the network is unable to create adequately complex decision boundaries. However, if there is a large number of hidden nodes, the training may be computationally costly [17], and it is more difficult for the trained network to create a generalized mapping using the training data set. Furthermore, keeping the number of hidden nodes to a minimum reduces the computational time needed for training. During the past few years, many researchers have investigated different approaches for selecting the number of hidden nodes. An algebraic projection to analyze the optimal hidden nodes was developed by Kung and Huang [18]. Based on the algebraic projection analysis, if the training patterns are completely irregular, then the optimal number of hidden nodes is the same as that of the training patterns. The hidden nodes in this case are also called grandmother cells, meaning that each cell could respond to a specific input pattern (recognizing their grandmother). "When there is some regularity embedded in the patterns, the number of hidden nodes will be dictated by the number of regularity features instead of the number of training patterns". In other words, because of the regularity

inherent in the training patterns, the effective dimension of the subspace spanned by the activation values of the hidden layer will in turn decrease. Therefore, the number of hidden nodes should be less than the number of training patterns.

Another way of setting the number of hidden nodes is to find a theoretical upper bound on the number of nodes needed. In the paper by Huang and Huang [19], a least upper bound (LUB) is derived for the number of hidden nodes needed to realize an arbitrary function which maps from a finite subset of E^n into E^d, where E^n and E^d are n- and d-dimensional Euclideam spaces. According to this method, if a finite subset S is a K-element of E^n, then K-1 is the least upper bound for the number of hidden nodes, being capable of realizing an arbitrary function f in a multi-layer perceptron.

Some researchers have found that the number of hidden nodes depends on the number of input or output nodes, rather than the number of training patterns. Kudrycki [17] predicted that the maximum number of hidden nodes for a single hidden layer case should equal OPx3 where OP is the number of output nodes. Hecht-Nielsen [20] showed that 2N+1 hidden nodes is an upper bound for a network with one hidden layer to compute an arbitrary function, (where N is the number of network inputs). This does not specify the number of independent patterns.

It is obvious that there is no general rule to select the optimal number of hidden nodes matching all the applications. This number is depends on the application and is selected based on the given problem and the type of training data. Recently, some studies have been carried out using a genetic algorithm [21] in order

to learn the most appropriate architecture. The selection using this method is made from network size, training time, running time, etc. There are two other approaches to arrive at a hidden layer structure. One approach is to start with no-hidden-layer network, and add the hidden nodes gradually during training [22]. The other is to start with many hidden nodes in the network and remove unimportant ones from the network during training. All these studies are at the preliminary stage; however, they have given encouraging results for simple test problems. A systematic and detailed simulation study is necessary to resolve this issue.

In this research, a multi-layer feedforward network with back-propagation training is developed to estimate the concentration of chemical composition. The number of hidden layer nodes is optimized by numerical experimentation. Considering the sensitivity to initial conditions, each training procedure is repeated ten times with different, random initial weights. This method is used to train the networks with varying number of hidden nodes in order to select the optimal one for this application. In this method, the numbers of input and output nodes are fixed. The hidden layer nodes are varied from two to twenty-five. These networks are trained and tested by using identical training and testing data. Finally, the performance of the networks are compared to select a network with the smallest estimation error. This approach is used to determine the optimal number of hidden nodes.

4.3 Analysis of Different Network Configurations

The configuration of the neural network is another factor that may affect the network's performance. A careful modification of the network's structure will significantly increase network learning efficiency. Besides the number of hidden nodes, the number of output nodes is another parameter to be considered for the optimization of network architectures. Normally, the output vector sizes is set to be equal to the number of parameters (percent compositions). In the study of chemometric data, three different configurations were studied. First, the network with five output nodes was developed. The network was trained using data set containing five target outputs. Therefore, five components (n-hexane, iso-octane, toluene, p-xylene, and decane) are estimated simultaneously through the network. Since there is an internal chemical relationship between components n-hexane and decane, a combination of these two components as a single output may improve the performance of the network. A network with four output nodes was then established. Four chemical components (iso-octane, toluene, p-xylene, and n-hexane+decane) were used as target outputs to train the network. The resultant network was tested using the database in this application. The third configuration is designed to model each component separately. Thus five (or four) independent networks were developed. Each network was used to estimate the percent of one of the chemical components. In some cases, training multiple networks may require much more time than training a single network to estimate all parameters simultaneously. However, this may still not improve the performance of the network. Thus, the network

configuration depends on the feature of the data and the requirements of the problem under study.

4.4 Networks with One and Two Hidden Layers

The number of hidden layers is another important issue for the optimization of network architecture. When using the network with back-propagation training to approximate an output as a function of the inputs, the question is "how many hidden layers are needed for the give problem ?" Cybenko [23] has shown that two hidden layers are sufficient to compute an arbitrary output function of the inputs. The capabilities of an upper bound of two hidden layer are also emphasized by Lapedes and Farber [24]. The essential points of their proof on this hidden layer issue can be summarized as follows. (1) Any mapping function F{X} can be represented by a linear combination of localized bumps that are each non-zero only in a small region of the domain $\{x_i\}$; (2) Such bumps can be constructed with two hidden layers. Detailed analysis is given in their paper [24]. A further argument on the issue of the number of hidden layer was presented by Chester [25]. In this paper, the author proved that the number of nodes in a network with one hidden layer might grow without bound in order to improve accuracy. This dramatic increase of hidden nodes will in turn induce an extremely high computational cost. One can achieve high level of accuracy for network learning with a smaller training time by using a network with two hidden layers. This may be because the neurons in the first hidden layer partition the input space into small regions, thus the neurons in the second layer can

compute the desired function within that region. However, Hecht-Nielsen [20] has proven that a network with only one hidden layer can compute any arbitrary function of its inputs. There is no strict proof that shows a two hidden layer network is always better than one hidden layer network, or vice versa. Based on the current knowledge, we may conclude that a feedforward network with a maximum of two hidden layers can approximate arbitrary mapping between two sets of information.

During the study of chemometric data, both one and two hidden layer networks were developed to test the network performance with various numbers of hidden layers and hidden nodes. The training and testing procedures are repeated ten times for each case. The experimental results will be discussed in Chapter 6.

4.5 Utilization of Network Ensembles

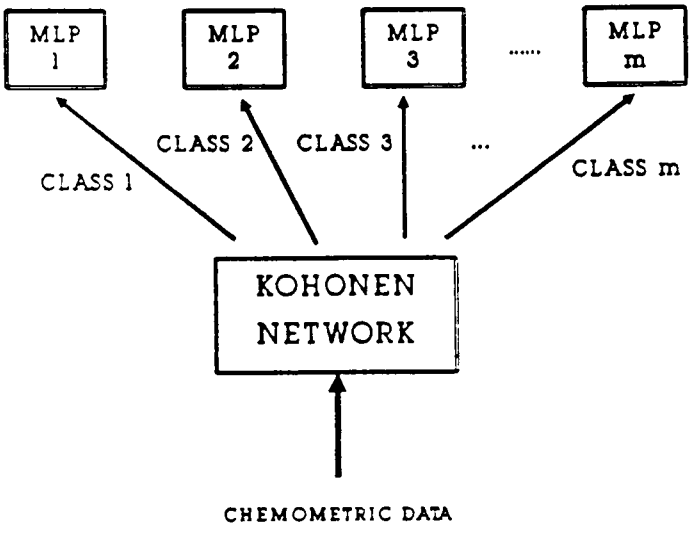
In the previous sections, the approaches to select an optimal network architecture (for example, the number of hidden layers and the hidden nodes per layer) were discussed. It is important to note that even after optimizing the network's architecture, the estimation error may not be small for one pass through the network. This is because, a given training sequence may act as a random selection of the network model. To further reduce the network estimation error, a method called "network ensembles" is introduced in this research. The basic idea of this approach is to establish an ensemble of neural networks instead of using a single network. Each network is trained using the same data set with different initial conditions. When the training is completed, all similar networks will be tested using the same data. This

in turn provides their estimation results for a given input pattern. The final result of estimation is obtained by taking the average value over all network outputs. Due to the underlying randomness in the initial values of connection weights, the local minima in network learning may differ greatly from one run of the algorithm to the next [9]. This may cause different final values of connection weights for each run. Such randomness tends to differentiate the errors of the networks, so that the network will have different estimation for the same input pattern. Ensemble averaging is desirable from a statistical point of view; the collective decision produced by a set of networks is less likely to be in error in comparison with the decision made by any one of the individual networks. In this research, ten networks were implemented and trained using same spectral patterns to estimate chemical concentrations. The comparison of single network with ensembles is performed. The conclusion from this study is demonstrated in Chapter 6.

4.6 A Two-stage Network

To obtain good estimation, a neural network usually needs a sufficient number of training patterns. However, as described in Section 4.1, a large number of training patterns may require a large number of hidden nodes in order to achieve the best network performance. This will increase the number of connection weights, and consequently will increase the network's learning time. Preprocessing the training data is one way to improve this problem. A two-stage network which combines back-propagation and Kohonen self-organizing neural networks is introduced, with the

purpose of achieving a better performance when there is a large number of training patterns. Figure 4.1 shows the structure of a two-stage network. Kohonen self-organizing network may be used as a preprocessor, so that a large number of training patterns can be first clustered into different classes. In the second stage, several back-propagation networks may be developed, one for each class of data. Individual back-propagation network is used to train and test each class of training patterns with similar properties. This reduces the learning time by decreasing the number of training patterns necessary to train the network. By using such a two-stage network, one can obtain better network performance for a problem that requires a large number of training patterns, and has distinct features among the training sets.



MLP is Multi-layer Perceptron

Figure 4.1. Schematic Representation of A Two-stage Network

CHAPTER 5

QUANTIFICATION OF ESTIMATION ERROR AND LEARNING STRATEGY

5.1 Introduction

The inaccuracy of training data is another major factor that affects the performance of neural networks. In real world applications, measurement noise contributes to the errors seen in training data. Phase shifting of the training patterns is another problem which will influence network learning. Therefore, it is necessary to minimize these effects and to test the robustness of the networks by simulation studies. This chapter focuses on four topics: (1) Study of the effect of random noise in the Raman spectral patterns. (2) Study of the sensitivities of the neural network to both regional perturbations and phase shifts in the input spectra. (3) Study of the behavior of the convergence of network connection weights during training. (4) Study of the method called combined subset training.

5.2 Effect of Random Noise in the Input Pattern

5.2.1 Effect of Measurement Noise on Network Recall

The objectives of the study of measurement noise effect in the input pattern was to test whether a neural network can provide reliable estimation in a noisy

environment and to quantify estimation errors on the network recall for a given noisy input pattern.

To pursue the stated objectives, a FORTRAN program was developed to generate various levels of Gaussian noise. Two different types of tests were performed. In the first test, random Gaussian noise with standard deviation from 0.5% to 5% of average amplitude of input spectral patterns was generated. One hundred sets of noise data were generated for each noise level case. This random noise was then added to the original spectra and the resultant spectra were used to test the network. The second test used random noise with standard deviation 0.5% to 5% of individual spectral amplitude. The noise was then added to the input spectra to test the network performance in estimating composition values. The overall standard deviation of network prediction errors for 100 sets of data (statistical average) was calculated for each case. The performance of the network to estimate concentration of chemical components under noise corruption was compared with the result of the noise-free case.

5.2.2 Network Training with Additive Noise

Training a network with additive noise data is one way to improve network performance in a noisy circumstance. To study the effect of additive noise on network training, three different types of training data were considered: (1) The training used spectra with additive noise only. (2) The training used both noisy and noise-free spectra. (3) The training used spectra as in (2), but in reversed order of

presentation. Random Gaussian noise with a standard deviation equal to 5% of the average amplitude of spectral pattern was generated and added to each of the training patterns. In the first study, twenty two spectral patterns with this additive Gaussian noise were used to train the network. The trained network was then tested using the remaining eight patterns. Two cases were considered while testing the network: one used the noisy test patterns, and the other used the noise-free test patterns. The estimation results in this case showed that there was no advantage when a network was trained with additive noise. This is due to the fact that the network treats the additive noise pattern as a totally new spectrum and tries to compute the output function of these new inputs. The second approach is to train the network with spectral patterns which include both noisy and noise-free spectra. Therefore, instead of 22 training patterns, 44 patterns (22 with additive noise and 22 without noise) were used to train the network. The resultant network was tested using the same data as in case one. The estimation results obtained in this case were much better than the results obtained by using the noisy spectra only. In the third study, noise-free and noisy spectra were presented alternatively during network training. This last case used 44 spectra and produced better estimation results than the second case. The experimental results of the study on network training with additive noise are demonstrated in Chapter 6.

5.3 Analysis of Network Sensitivity

5.3.1 Network Sensitivity to a Regional Perturbation of the Input Spectrum

The sensitivity of network estimation to changes in a portion of an input pattern is another way of testing the network stability. The characteristics of Raman spectrum discussed in Chapter 2 shows that a specific portion of the spectrum is related to one or more of the sample components. Therefore, the purpose of the sensitivity study was to determine whether a neural network can be trained for learning the inherent relationship between a spectral feature and sample property. In other words, the results of this study can be used to verify whether the network can learn the internal relationship between the chemical components and a specific region of the Raman spectrum. In this research, a detailed analysis was made to study the internal relationship between a peak in the Raman spectrum and one or more chemical components. Three methods were used to perform the analysis : (1) Adding random noise to the region of interest in the input spectrum. (2) Removing all the amplitudes from an input spectrum except the region of interest. (3) Removing only the amplitudes of the interested region from an input spectrum. The spectra that have been modified in one of the above forms were then used to test the previously trained network. The component with the largest estimation error among the various component concentrations indicates the maximum sensitivity to the selected spectral region.

5.3.2 Network Sensitivity to a Phase-shifted Input Pattern

In actual applications, one of the frequent problems is the shift in the measured spectrum as a function of wave number. This may be caused by instrumentation errors during the measurement. To demonstrate the sensitivity of the network output to a phase shifted pattern, the original input spectrum was shifted to the right (increasing wave number) by two points (wave number). To study the effect of network estimation on input pattern, the phase shifted pattern was presented to estimate the composition using the previously trained network. The estimation error for the new pattern was calculated. A comparison of the network estimation results obtained from the original spectrum and the phase shifted spectrum was made.

5.4 Behavior of Network Connection Weights

In the multi-layer feedforward network, the connection weights are adjusted using the back-propagation algorithm in order to minimize the error between the target and the network outputs. The behavior of the internal connection weights can therefore represent the network learning situation. In some cases the fluctuation of connection weights during training indicates instability in network learning. It is very valuable to understand the behavior of connection weights as the training iterations progress. This behavior can also provide insight into the effect of applying this methodology to solve certain problems. Another objective of this study is to find the best value or some specific rule to set up initial weights which now are mostly

randomly generated. A proper choice of initial weights would help to speed up network learning, and in turn improve network performance. In the present research, the study of the behavior of connection weights was made by observing the weights between the hidden layer and the output layer processing elements in a three layer network. The study was made by plotting each of the connection weight for different iteration numbers during training.

5.5 Combined Subset Training (CST) Method

Besides the network architecture, the size and order of training set are also very important factors that influence network learning. As described in Chapter 4, when there is a large number of training patterns, Kohonen feature mapping may provide a desirable preprocessing approach to reduce the number of patterns to train each network effectively. Another learning strategy applied in this study is called Combined Subset Training (CST) [16]. In this strategy, different sets of training patterns are organized to train the neural network according to a certain schedule. Using this method, a network may reduce its learning time for a certain range of accuracy. The basic idea of this method is the following: first pick a manageable randomly selected subset of training patterns to train the network. When this phase is completed, add another selected subset to the first training set and then train the network with the combined set. Repeat this procedure until the entire data is included. In this study, seven spectra were first selected to train the network. After the training was completed, another seven spectra were added to the first training set,

and the previously trained network was further trained using these fourteen patterns. Finally, all of the 22 spectra were used to retrain the network. The prediction of chemical composition was made using the network trained by this procedure.

CHAPTER 6

DISCUSSION OF RESULTS OF APPLICATION TO CHEMOMETRIC DATA

6.1 Experimental Results on the Study of Raman Spectroscopic Data

6.1.1 Experimental Results Using Different Network Structures

As described in Chapter 4, multi-layer feedforward neural networks were implemented to estimate concentrations of five chemical components using Raman spectroscopy data. To obtain the best network performance for this application, different approaches were studied in this research for choosing optimal network architecture. Studies of the network structure include the selection of the number of hidden nodes, number of hidden layers, and the number of output nodes.

The appropriate number of hidden nodes for the application of chemometric data analysis was selected from the experimental analysis. The three-layer perceptrons with a given number (2, 15-30) of hidden nodes were trained using the back-propagation algorithm. Regarding the effect of initial random connection weights, the network training procedure for each number of hidden nodes was repeated ten times with different initial conditions. The fractional concentration of each component was obtained by averaging these ten estimation values for each experiment. The spectra used in network training and testing are shown in Appendix

B. Figure 6.1 shows the overall average results of learning and testing for different number of hidden nodes. A conclusion of this study is presented in Table 1. The result shows that the best average performance was achieved by the network with 22 nodes in the hidden layer. This number is equal to the number of training patterns.

The effect of choosing various target (output) vectors was studied next. Network with five, four (after combing n-hexane and decane as one parameter) and single output nodes were trained to estimate the concentrations of chemical components. Figure 6.2 and Table 2 illustrate the network performance for these three types of network configurations. The experimental results clearly indicate that the network with one output node provides the smallest estimation errors for the chemometric data analysis. Therefore, the best number of output nodes for this application is one. From this it is concluded that the analyst must develop as many networks as there are chemical components in a sample, each network providing the estimate of one component.

The comparison of results of network with one (22 hidden nodes) and two hidden layers (8 and 14 nodes respectively) is demonstrated in Figure 6.3. The number of interconnections for the single hidden-layer network is: (190+1) * (22+1) + 23 * 4 = 4485, where 190 is the number of inputs, 22 is the number of hidden nodes, 4 is the number of output nodes, and 1 is the bias added to one factor of each term. The number of interconnections for the network with two hidden layers equals: (190+1) * 8 + (8+1) * 14 + (14+1) * 4 = 1714. The average prediction error using two hidden layers for the remaining 8 unlearned patterns is 0.0195, while with

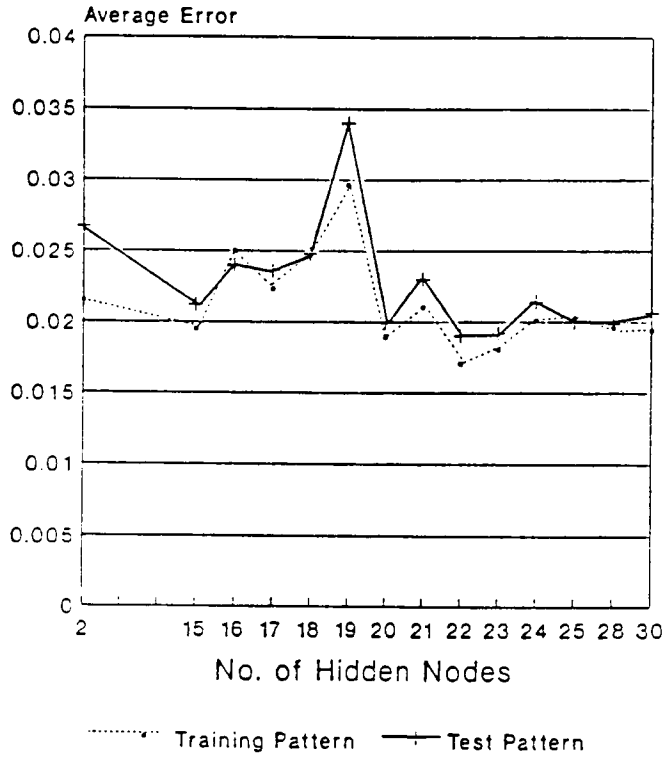


Figure 6.1. Network Performance with Different Number of Hidden Nodes.

Table 1. Result Comparison for Different Number of Hidden Nodes

Learning rate	Learning shape	Momental term	No. of iterations
0.7	0.8	0.9	1000
Input nodes	Hidden nodes	Output nodes	No. of layers
190	2, 15-30	4	3
No. of hidden nodes(min.RMS)	Min. RMS for training set	Min. RMS for testing set	Number of networks
22	0.013	0.017	10

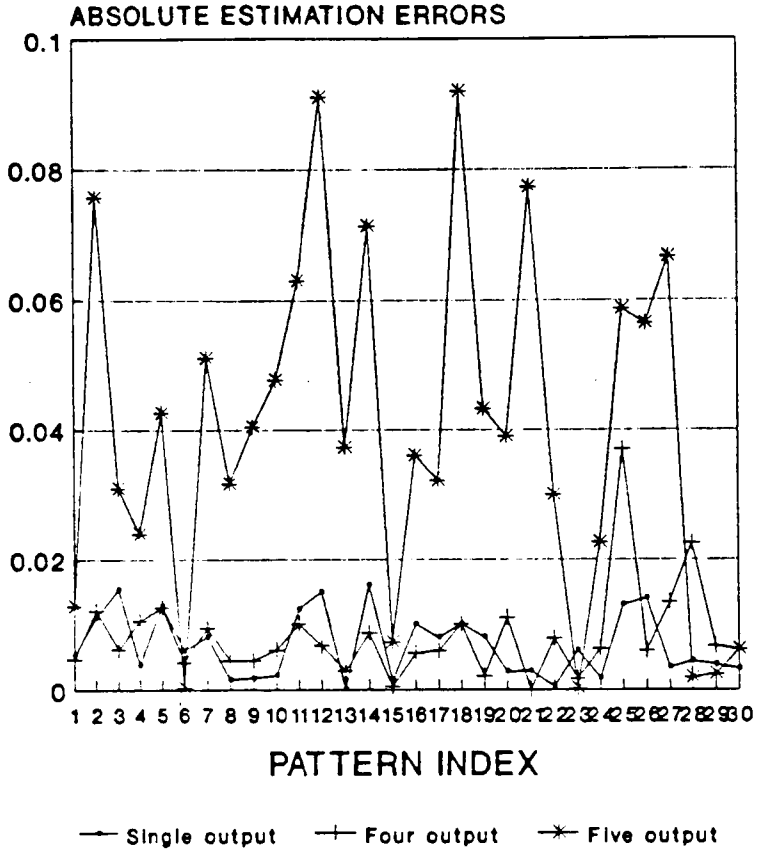


Figure 6.2. Network Performance with Different Output Nodes.

Table 2. Result Comparison for Different Number of Output Nodes

Learning rate	No. of iterations	Learning shape	Moment term
0.7	1000	0.8	0.9
No. of layers	Input nodes	Hidden nodes	Output nodes
4	190	22	1, 4, 5
RMS for single output node	RMS for four output nodes	RMS for five output nodes	Number of networks
0.015	0.018	0.07	10

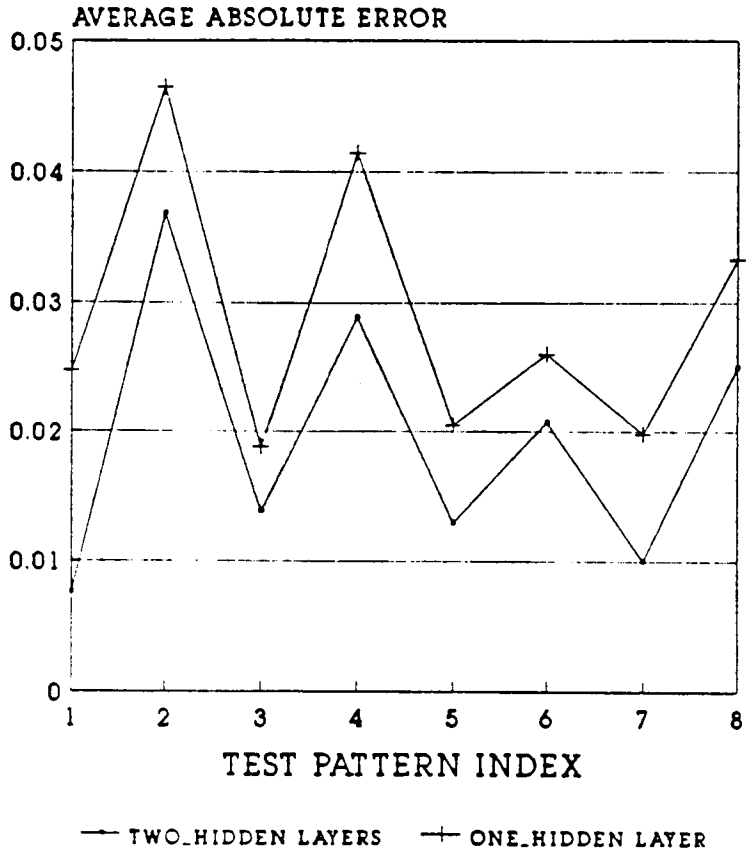


Figure 6.3. Network Performance with Different Number of Hidden Layers.

Table 3. Result Comparison for Different Number of Hidden Layers

Learning rate	Learning shape	Momenta term	No. of iterations
0.7	0.8	0.9	1500
Numbers of hidden layers	RMS for training set	RMS for testing set	Numbers of networks
1	0.010	0.03	10
2	0.008	0.0195	10

one hidden layer the average error of predicting concentration of chemical components for the same set of spectra is 0.0301. Moreover, because of the smaller number of interconnection weights (1714), the training time for the network with two hidden layers is much less than the training time for the network with single hidden layer. Table 3 gives the conclusion of this experiment. These results indicate that for this application the network with two hidden layers provides better estimation than a network with single hidden layer.

Another method used to improve the performance of the neural network is the utilization of an ensemble of networks. In this study, ten networks were implemented and trained using the same data set with different initial conditions. Our argument for using network ensembles was strongly supported by the experimental results. Figure 6.4 shows the average performance of 10 networks and the results with single network. A comparison of their performance is shown in Table 4. The smaller estimation errors show that using an ensemble of networks can significantly improve network performance.

6.1.2 Network Performance for Chemical Composition Estimation

Figures 6.5 - 6.8 demonstrate the network performance for estimating iso-octane, toluene, p-xylene and n-hexane + decane. A three-layer network with 22 hidden nodes are used for this study. The learning parameters and the RMS values of errors for these four parameter estimations are summarized in Table 5.

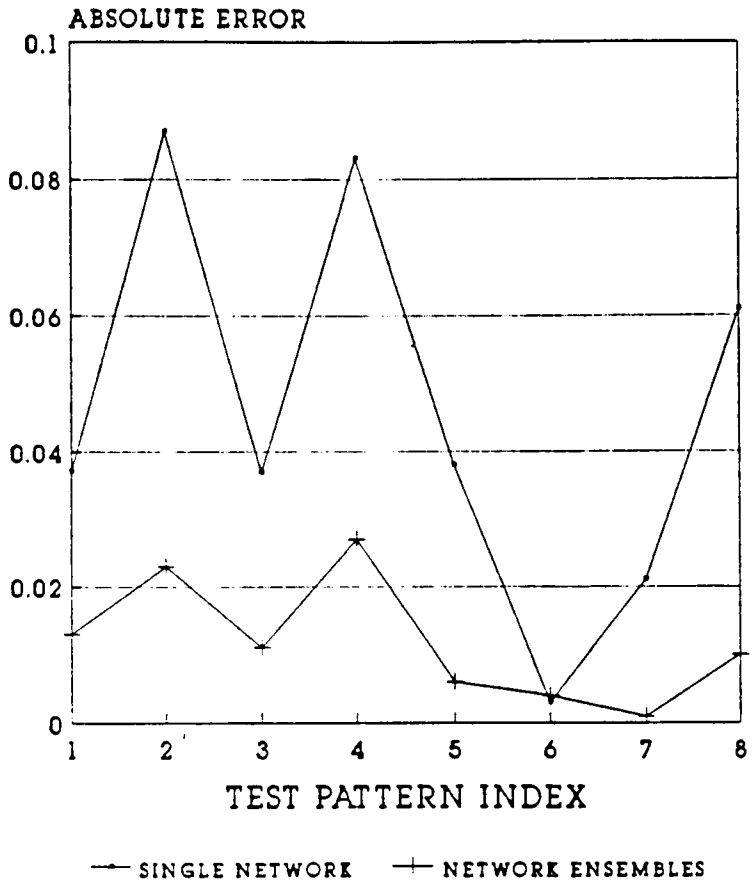


Figure 6.4. Ensemble versus Single Network Performance.

Table 4. Result Comparison for Single versus Ensemble of Networks

Learning rate	Learning shape	Momental term	No. of iterations
0.7	0.8	0.9	1500
Numbers of neural networks	RMS for training set	RMS for testing set	Numbers of hidden nodes
1	0.006	0.03	22
10	0.002	0.005	22

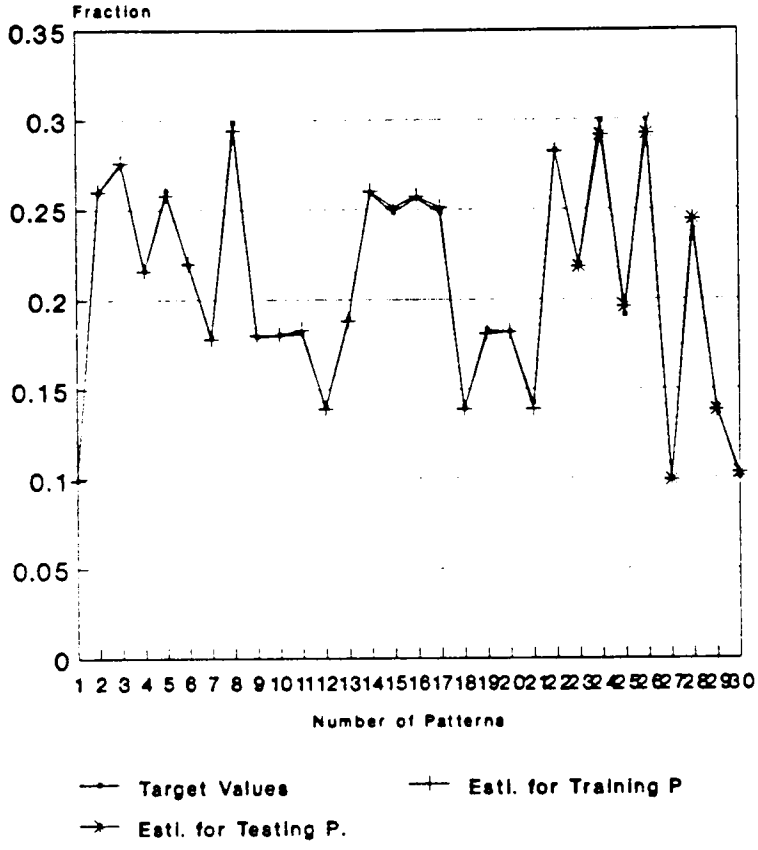


Figure 6.5. Network Estimation for ISO-OCTANE.

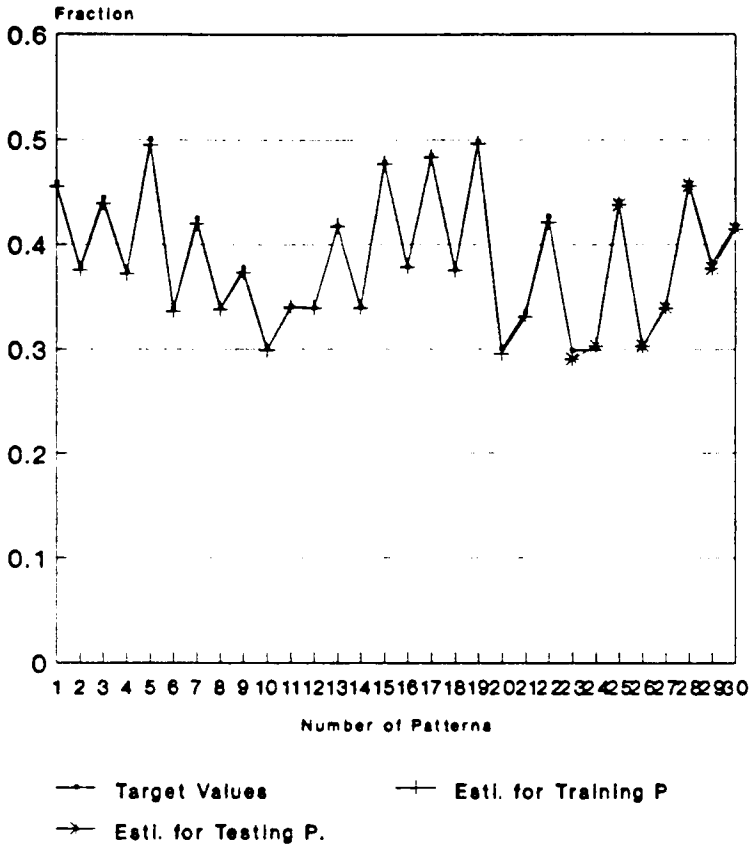


Figure 6.6. Network Estimation for TOLUENE.

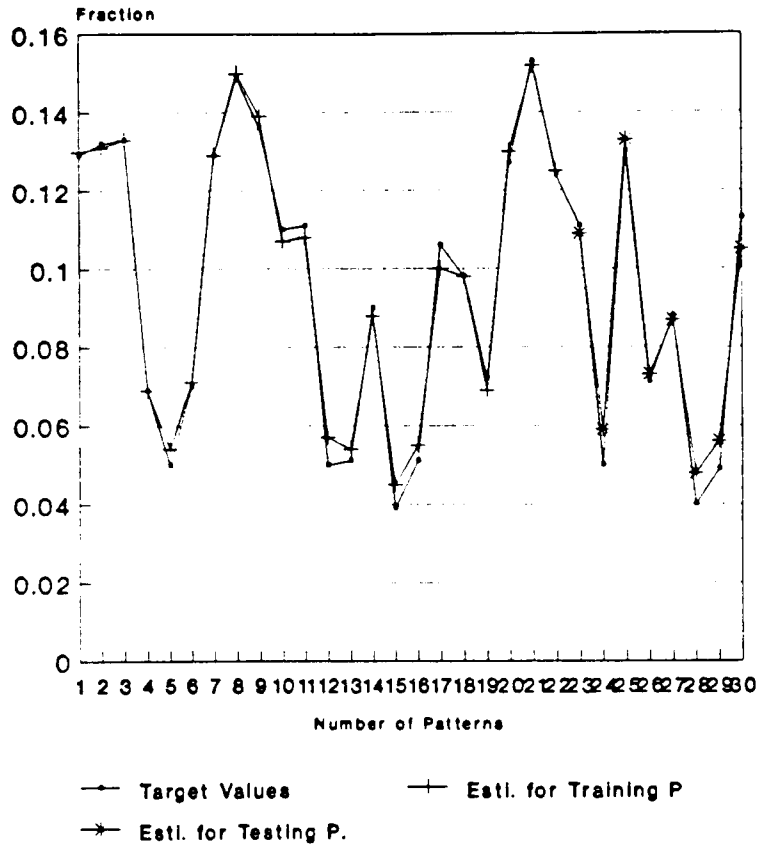


Figure 6.7. Network Estimation for P-XYLENE.

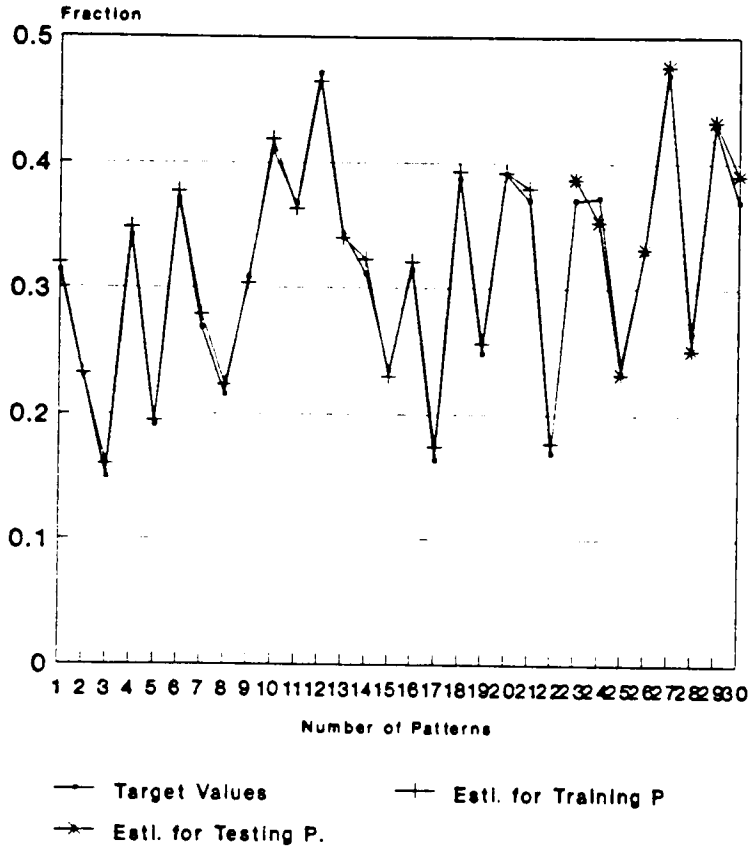


Figure 6.8. Network Estimation for N-HEXANE + DECANE.

Table 5. Summary of Chemical Composition Estimation (I)

Learning rate	Learning shape	Momental term	No. of iterations
0.7	0.8	0.9	2000
No. of layers	Input nodes	Hidden nodes	Output nodes
3	190	22	1
Chemical Components	RMS for training set	RMS for testing set	Number of networks
iso-octane	0.007	0.01	10
toluene	0.0052	0.006	10
p-xylene	0.0047	0.004	10
n-hexane+decane	0.0049	0.0058	10

6.1.3 Effect of Input Noise to Network Performance

In Chapter 5, two types of random noise were described for the noise analysis. Tables 6 and 7 summarize the experimental results for both types of noise signals. In Figure 6.9, a bar plot is made to demonstrate the network performance for the noise case. The standard deviation of each network output, as reported in the tables, was obtained using 100 trials of the random noise testing. That is, for each noise level 100 estimation were obtained. From Figure 6.9 it is observed that the network prediction error increases almost linearly as a function of percent input noise. For an input noise of 3%, the deviation of prediction error is approximately 2%. This result illustrates that the network is noise tolerant upto a certain level of noise in the input features.

One way to obtain a better performance in a noise environment is to train the network with an additive noise. Figure 6.10 shows the comparison of the results for three types of training patterns. The network was trained with additive noise in the spectrum, with no noise, and with and without noise learning patterns. The presentation of alternative noise and noise-free patterns provides the best performance of composition estimation. Both noise-free and noise data were used to test the trained networks. The network estimation errors for the eight unlearning patterns presented in the Figure 6.10 were obtained by taking the average of both noise free and noise testing data.

Table 6. Network Performance for Additive Noise in the Spectrum (Constant Noise Level)

Random noise (%)	Standard deviation of sample components (%) No. 1	No. 2	No. 3	No. 4
0.5	0.5	0.3	0.65	0.3
1	1	0.5	1	0.67
2	1.8	1	2.2	1.04
3	2.7	1.6	3.2	1.5
4	3.85	2.15	4.8	2.4
5	5.2	2.7	5.8	2.9

Table 7. Network Performance for Additive Noise in the Spectrum (Spectral Dependent Noise Level)

Random noise (%)	Standard deviation of sample components (%) No. 1	No. 2	No. 3	No. 4
0.5	0.6	0.2	0.48	0.417
1	1.28	0.57	1.16	0.75
2	2.38	1.03	1.9	1.47
3	4.19	1.5	3.55	2.5
4	5.3	1.78	3.89	3.16
5	7.2	2.6	5.56	4.8

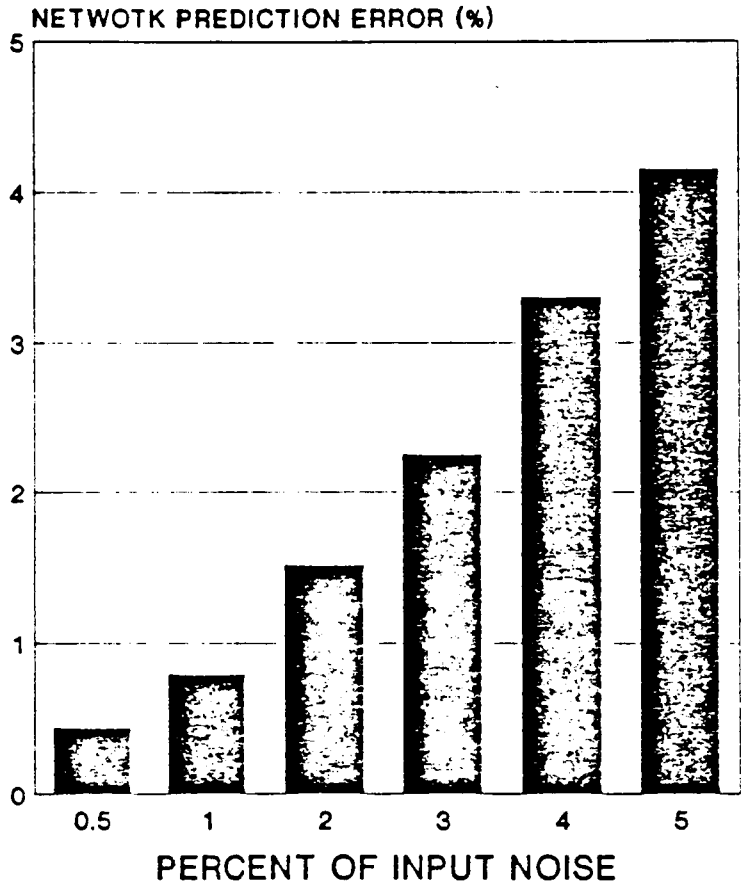


Figure 6.9. Effect of Input Noise on Network Estimation

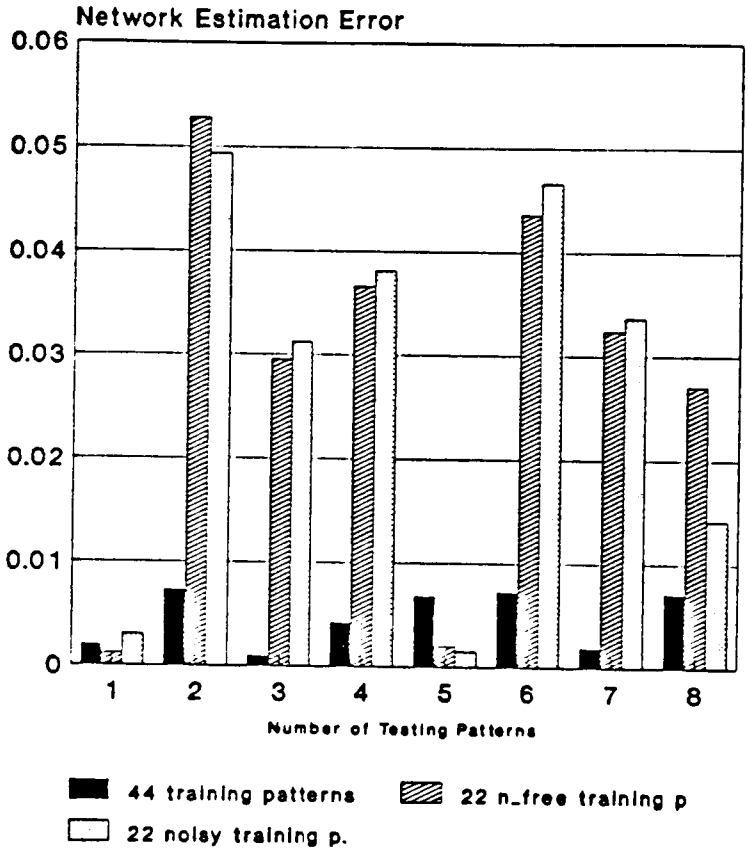


Figure 6.10. Comparison of Network Estimation Using Different Training Patterns

6.1.4 Sensitivity Analysis of Network Prediction

The objective of this study is to determine the sensitivity of a region of the spectrum on one or more components in a given sample. A specific region of input spectrum was selected in order to study the sensitivity of network performance. Figure 6.11 shows the Raman spectrum used to relate a portion of this spectrum with one or more chemical components in the sample. As an illustration of this study, approximately 5% bias error was added to the indicated portion of the spectrum in the figure. The resulting fractional components as estimated by the network and the error are shown in the Figure 6.12. The black bar indicates the network prediction value with original input spectrum (without additive bias error). The cross-hatched bar shows the network prediction results for the new pattern (after adding bias error to the portion of the spectrum). The third bar shows the error between two results for each component. It is obvious that the estimation of n-hexane and decane has maximum error. This result indicates that n-hexane and decane fraction are the most sensitive components affected by the bias error.

Further study of this sensitivity is made by selectively excluding the indicated region from the spectrum, and by excluding all the spectral features except the portion of interest. The results in both cases showed that the perturbation had maximum effect on the prediction of n-hexane + decane concentration. Thus the current analysis is capable of associating Raman spectral region(s) with specific chemical components.

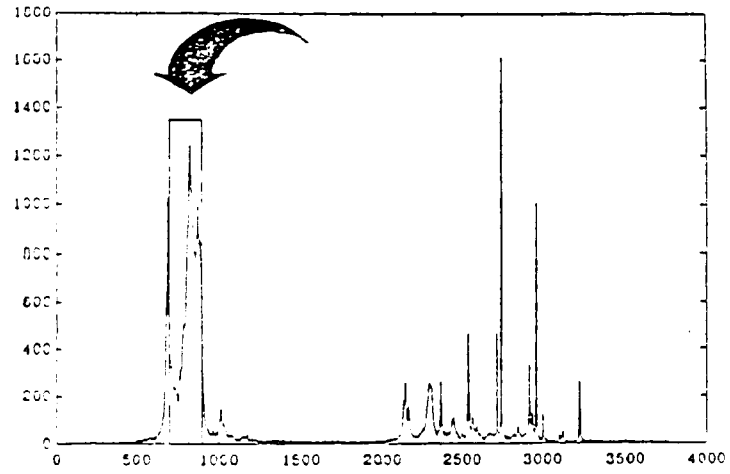


Figure 6.11. Regional Perturbation in a Raman Spectral Pattern

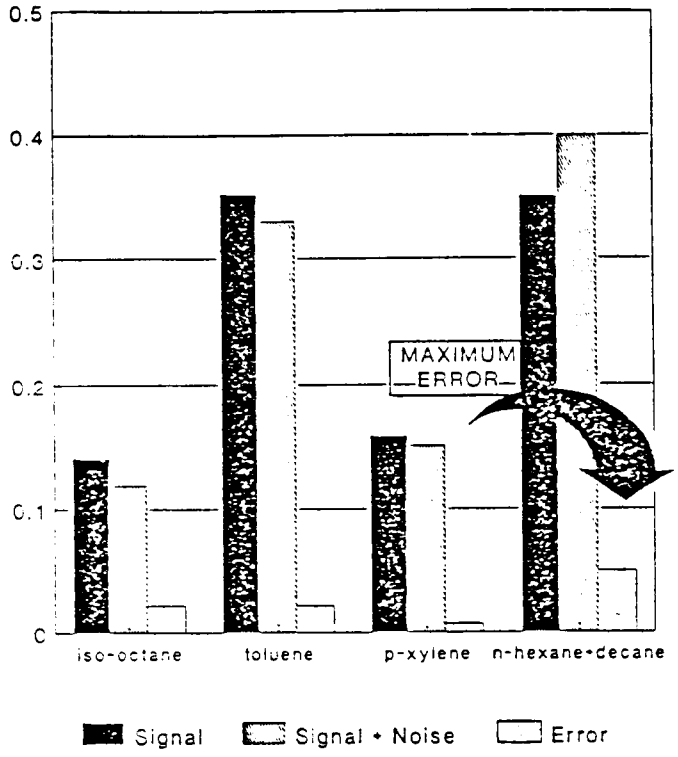


Figure 6.12. Sensitivity Analysis with Regional Perturbation in Spectrum

6.1.5 Study of Phase Shifted Spectrum

Figure 6.13 shows the original and phase-shifted (to the right) Raman spectral patterns. To study the sensitivity of neural network to the phase shifted input patterns, both of original and shifted patterns were used to test a network which was trained using non-phase shifted patterns. Table 8 shows the comparison of network estimations for these two input patterns. The experimental result indicates that the network is very sensitive to the spectral phase position. Furthermore, the network is more sensitive to phase shifting than to the addition of random noise to the input patterns. Therefore, the use of phase shifted spectrum produces large error in concentration estimation. Consequently, preprocessing the training pattern is necessary if it is suspected that spectral shift problems exist in the data.

6.1.6 Study of Network Connection Weights

Two types of plots were made to analyze the behavior of the convergence of network connection weights. One plot shows a connection weight behavior as a function of training iterations. The other shows changes in all the connection weights during network learning phase. Figure 6.14 presents a typical internal connection weight between the hidden layer and the output layer as a function of iteration. This plot shows the connection weight converging to a certain value as the number of iterations increases (no fluctuations are noticed). Figure 6.15 shows the behavior of all 22 connection weights during network learning. Figure 6.15 (a) and 6.15 (b) illustrate the cases of the first 50 and 2000 learning iterations respectively. From the

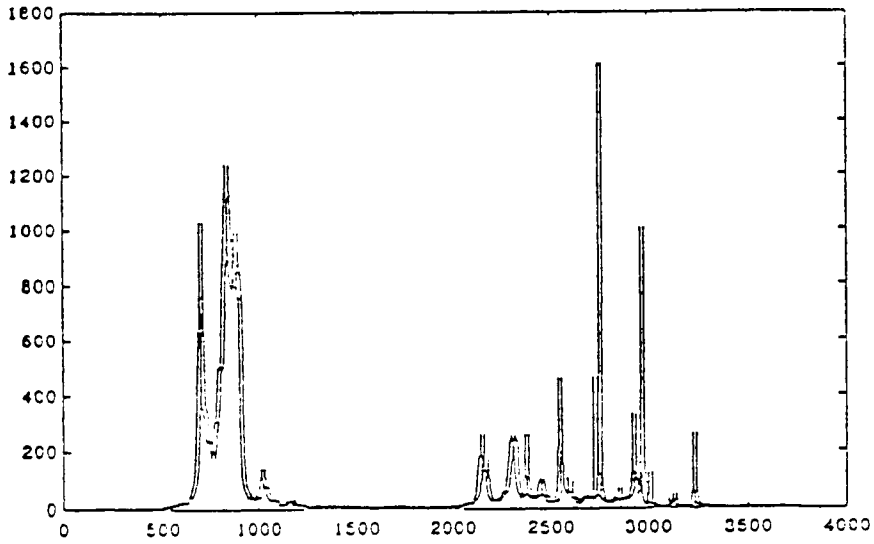


Figure 6.13. A Phase-shifted Spectral Pattern

Table 8. Effect of Network Estimation for a Phase-shifted Input Pattern

Network output	Estimation Error	
	Before shift	After shift
No. 1	0.0027	0.113
No. 2	0.0166	0.036
No. 3	0.0034	0.052
No. 4	0.0078	0.164

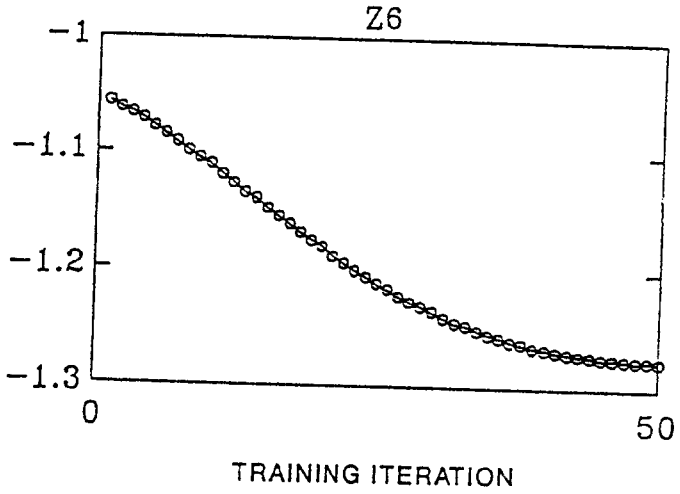
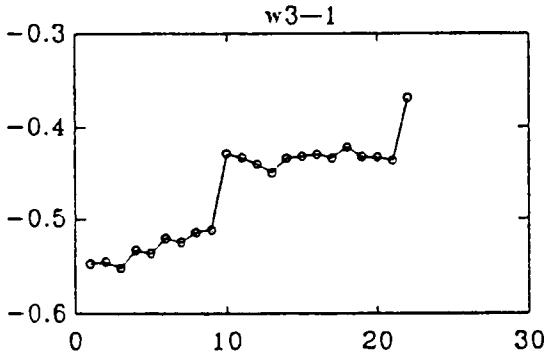
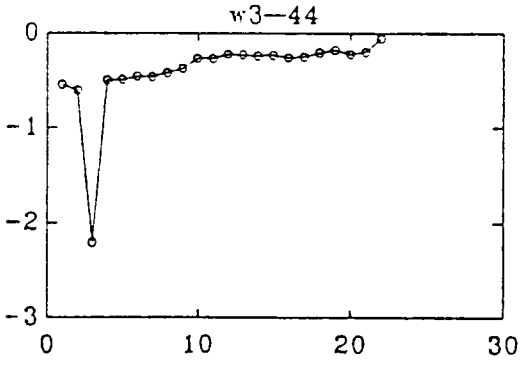


Figure 6.14. Behavior of a Connection Weight between Hidden Layer and Output Layer During Network Learning.



(a)



(b)

Figure 6.15 Behavior of 22 Connection Weights After (a) 50 Training Iterations and (b) 2000 Training Iterations.

variations in the connection weights it is observed that the values of weights become stable after about 1500 learning iterations.

6.2 Experimental Results on the Study of Near Infrared (NIR) Spectroscopic Data

The data used in this application consisted of 31 spectral patterns. Each pattern has 700 intensity values and the sample consists of three unknown chemical components (x, y, z). Two different methods were used to process the data. In the first approach, the data reduction was performed to compress the size of data in each spectrum. The intensities in each pattern were averaged over 5 points (wave numbers), resulting in 140 intensity values for each spectrum. Data normalization was then made for the 140 spectral values in each spectrum. The resultant data were used to train and test the networks. A four-layer feedforward network with two hidden layers was developed for this study. Twenty seven spectral patterns were selected as training data, the rest four patterns served as testing data. Several pairs of number of hidden nodes were selected during this study, the choice of 12+16 is found to be the best for both network training and estimation. It was observed that the estimation errors for this data set were too large. In fact, the network learning could not be improved after the local error reached 0.013. In order to improve the performance of the network estimation, an approach to enhance the feature of the training spectrum was implemented. The second derivative of the values in each pattern was calculated to emphasize the spectral features. Figures 6.16 and 6.17 show the compressed spectrum before and after taking the second difference. Note

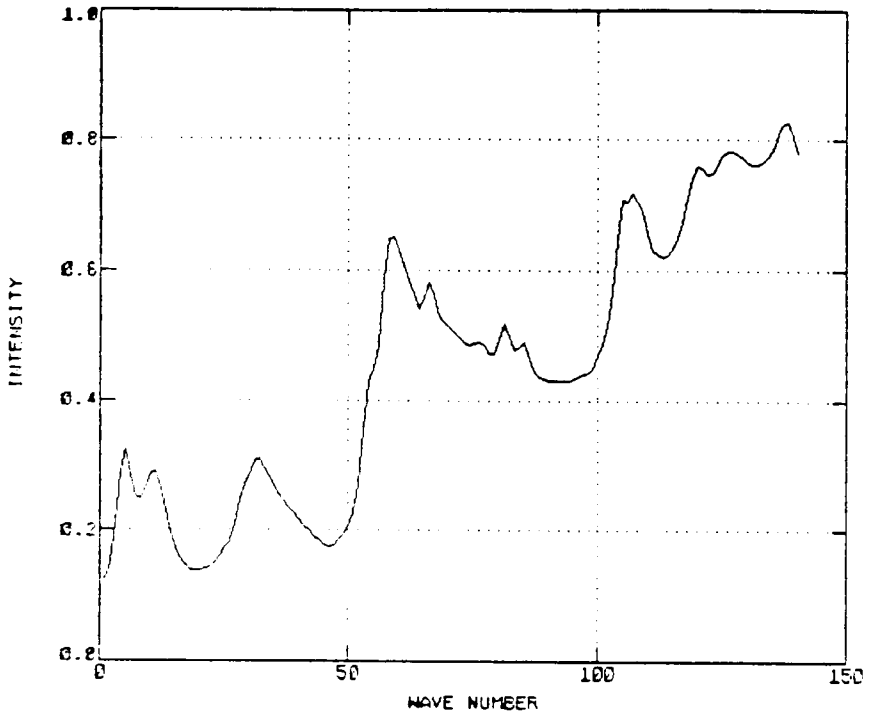


Figure 6.16. The Smoothed Spectrum.

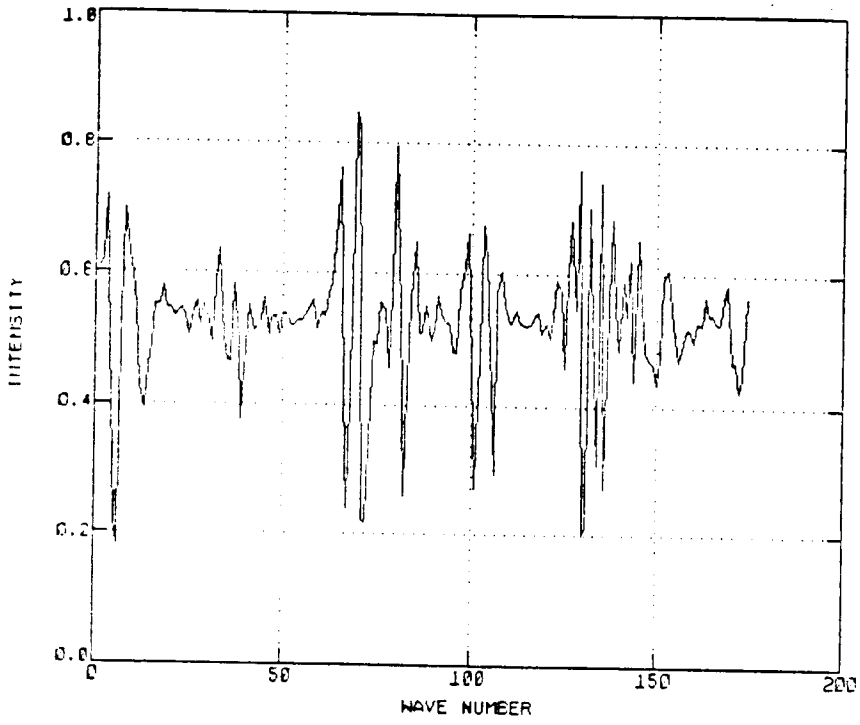


Figure 6.17. The Compressed Spectrum After Second Differencing

that the spectrum shown in Figure 6.17 is obtained after taking second difference of original spectrum and then averaging the resultant data over 4 points. Therefore, the final training data consists of 174 intensities for each spectrum. Figure 6.18 presents the network estimation for the same chemical components as the first case, except for the second differencing of the spectral points. This procedure was suggested by one of the member companies who supplied the data. The background effect would be minimized by this approach. The overall estimation error in the second case was 0.002. The effect of a first derivative is also studied for this data. The estimation results showed that the first derivative was also effective in enhancing the features of the spectra and in improving network performance.

The network structure and its performance of estimating three chemical components are summarized in Table 9. This indicates that an artificial neural network trained with the back-propagation algorithm can provide a very effective method to process spectral data and to study the properties contained in these data.

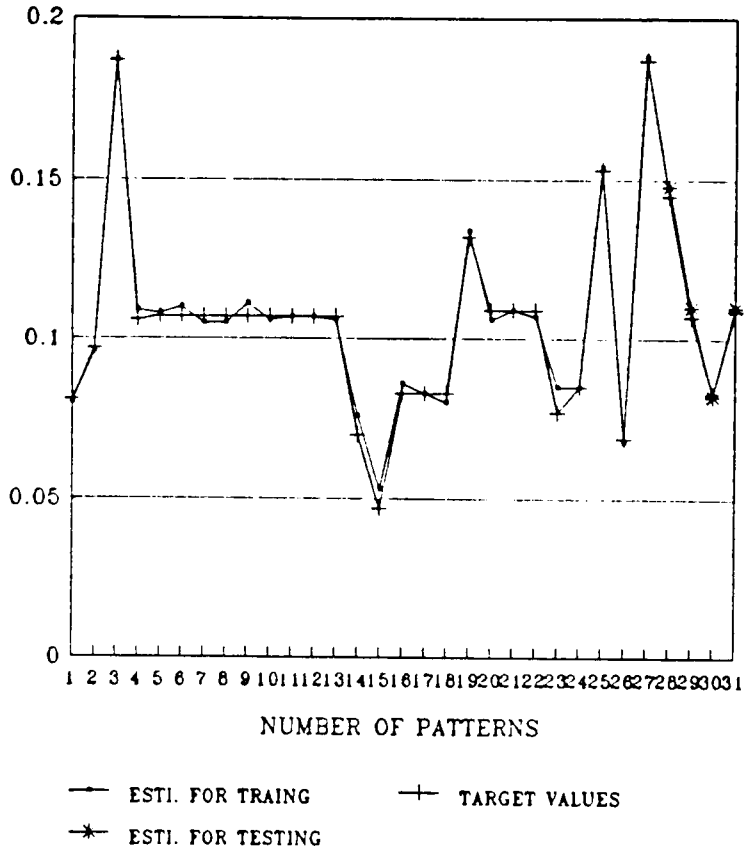


Figure 6.18. Network Estimation of Component X Using Derivative Data Set

Table 9. Summary of Chemical Composition Estimation (II)

Learning rate	Learning shape	Momental term	No. of iterations
0.7	0.8	0.9	3000
No. of layers	Input nodes	Hidden nodes	Output nodes
4	140	12+16	1
Chemical components	RMS for training set	RMS for testing set	Number of networks
x	0.0015	0.002	10
y	0.0021	0.005	10
z	0.0024	0.004	10

CHAPTER 7

APPLICATIONS TO POWER INDUSTRY AND NUCLEAR MEDICINE PROBLEMS

7.1 Lubrication Oil and Other Chemical Analysis in Power Plant Maintenance

The purpose of lubrication analysis is to monitor oil quality and the condition of the equipment lubricated. "These are determined from representative fluid samples obtained at key points in lubrication system flow paths [26]." Oil samples are taken and analyzed on a regular basis. The analysis results are reviewed and trended to predict potential equipment failure or reveal lubricant contamination and deterioration. The benefit of using a lubricant analysis may be summarized as follows: 1. To reduce costly multi-component failure and downtime by limiting the repairs to specific components, rather than major overhauls or entire unit replacement. 2. To utilize the database as a management tool.

Spectrographic analysis is one of the routine tests performed by predictive maintenance specialists in order to determine wear metals such as aluminum, silicon, copper, lead, and iron. Gamma, infrared (IR), and near infrared spectroscopy can be used for analyzing engine lubricants and lubricants used in gear boxes, air compressors, pumps, and motors. Lubricant analysis measures the inherent ability to reduce friction and wear in a machine. Some tests indicate the lubricant's ability to protect internal parts from corrosion.

Artificial neural networks may provide a new approach for lube oil analysis. Similar to the principle described in the previous chapters, a neural network can be utilized to give information concerning component conditions and lubricant quality which may be used in maintenance planning. A multi-layer feedforward neural network can be developed using selected intensities of spectra as the inputs, the component condition or lube oil quality recommendations as the outputs. The back-propagation algorithm may be employed to train the resultant network using a set of experimental data. When the network is trained, it can provide the information by presenting a set of intensities of a given spectrum. Because of the quick response feature of a trained neural network, this new method will enhance the capability for on-line surveillance and provide quick recommendations. There is often disagreement in the results generated by different vendors of lubrication oil analysis. Neural network application to this problem can also be used to standardize lubrication oil and other chemical analysis.

7.2 Chemistry Analysis in Nuclear Industry

Artificial neural network technology can be also applied to applications in the nuclear industry such as effluent gas analysis, boiler water chemistry analysis, and reactor coolant analysis. The neural network method can be used as a standard methodology for these problems.

In the nuclear power industry, water chemical analysis has played a significant role in minimizing corrosion, fouling, and radioactivity. Chemical control of reactor

coolant is one of the functions of the chemical and volume control system (CVCS) [27]. Water is an aggressive substance when in contact with structural materials at elevated temperatures. Water coolant chemistry and corrosion processes are important in the reliable operation of nuclear power plants. Because of this, the reliability of many systems at nuclear power plants depends greatly on water control [28]. There are several factors related to the water chemistry control, one of which is the co-ordination of lithium and boron concentration in the primary circuit to maintain the pH within a narrow range during reactor operation. In some PWRs where sub-cooled boiling occurs and enhanced corrosion of zircaloy cladding can be observed, hydrogen is used to control oxygen and to remove nitrogen. Therefore, oxygen control is another issue of chemical control of reactor coolant. Chemical shim and makeup, and reactor coolant purification are the other technologies used in water chemistry control during reactor operation.

Neural network can be used to analyze water chemistry through measurement information (such as spectroscopic data). The concentrations of chemical components as well as diagnostics about water condition can be estimated using a neural network model.

7.3 Quantitative Analysis of PET Using Back-propagation Neural Networks

The Positron Emission Tomography (PET) technique has been extensively used in the nuclear medical field. One popular application of this technique is to use a PET scanner to inspect organs such as the heart and the brain. One of the

important parameters for identification is the blood flow rate in the tissue and is used to test the potential disease in the heart and the brain. This was previously conducted using radioactive tracers and mathematical model identification. The artificial neural network computing method provides new capability for analyzing PET scan data [29].

A three layer feedforward neural network with back-propagation training algorithm was developed to estimate the blood flow from PET scan data. A system representation is described in Figure 7.1. The activity values obtained from dynamic PET scans for a selected part of the organ were used as input signals to the network. The data about the blood flow within the same part of the organ were simulated with a mathematical model and served as the target output for network training. Time-activity curves obtained with PET scans for eight sections of the organ are shown in Figure 7.2. These eight patterns and their blood flow data were used for network training, the remaining 24 patterns for different sections of the organ served as testing. Different number of hidden nodes are tested for obtaining better performance. The network with 8 hidden nodes was found to perform better in this application. Once trained, the blood flow neural networks estimates could be generated in real time. The average estimation error for the testing patterns was 0.005 [30]. Figure 7.3 shows the activity curve fitting result for a test pattern. The solid curve is original test pattern and the estimation result is shown in dashed curve. Table 10 presents the network structure and learning parameters. These results demonstrate the feasibility of using a neural network coupled with mathematical

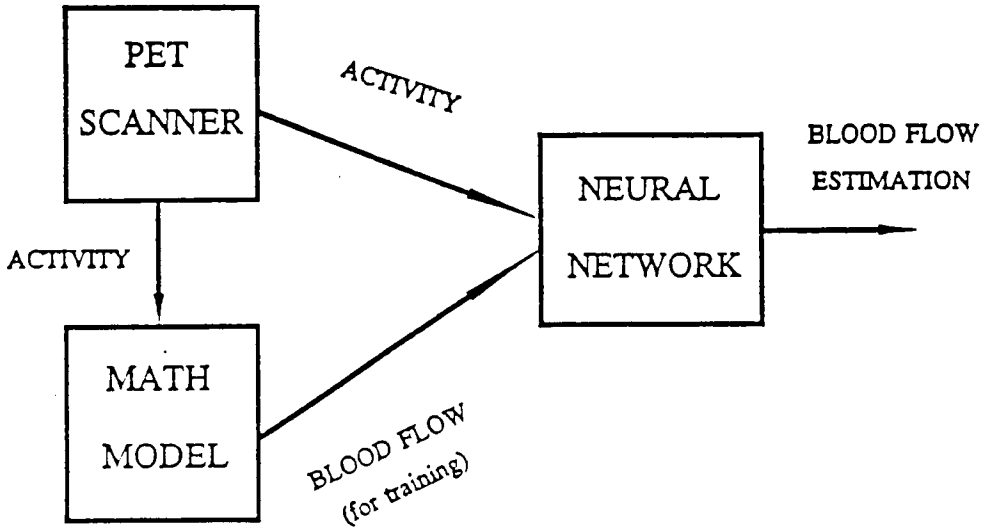


Figure 7.1. A Network System Representation for Blood Flow Estimation.

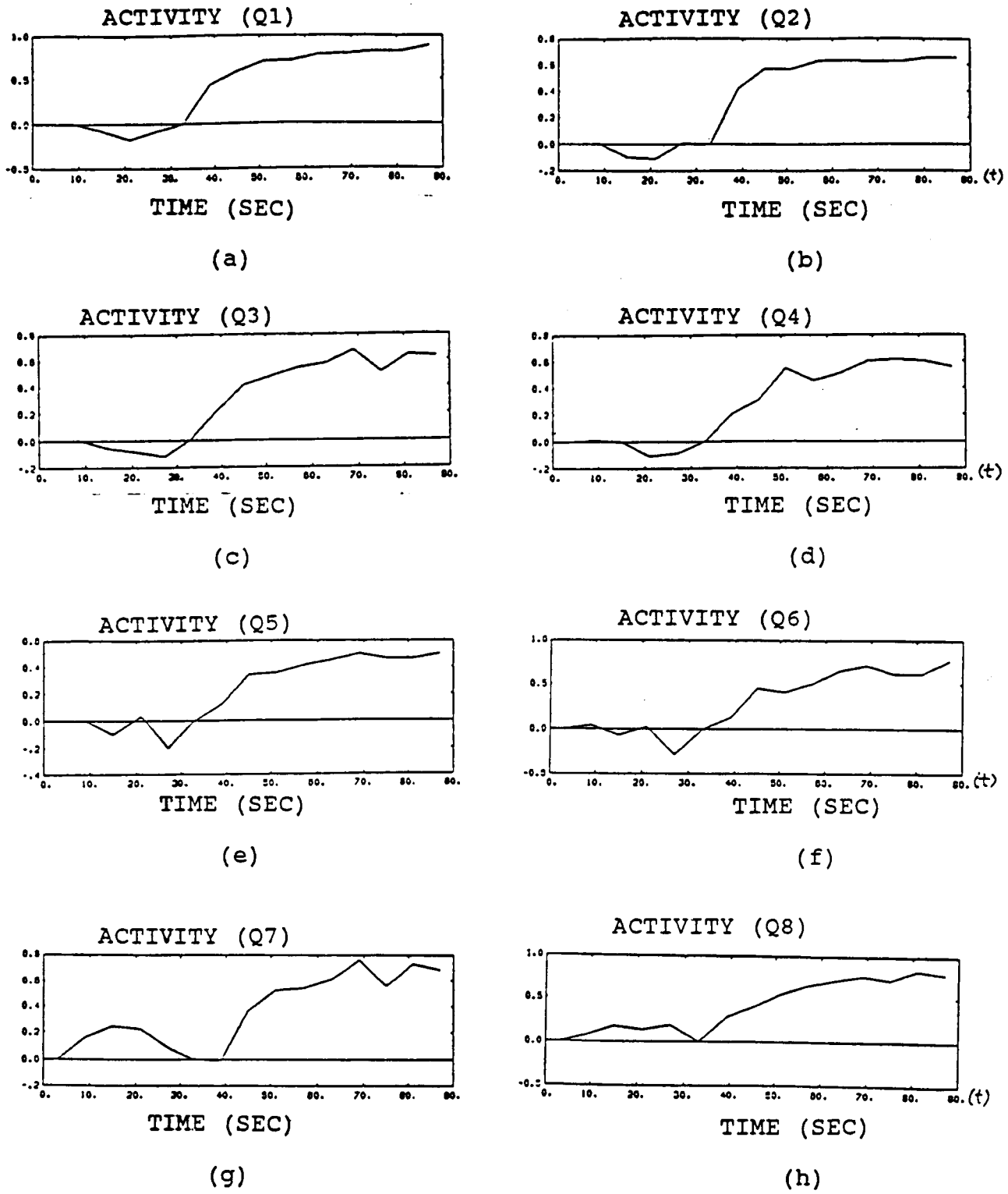


Figure 7.2. Time-Activity Curves for Network Training.

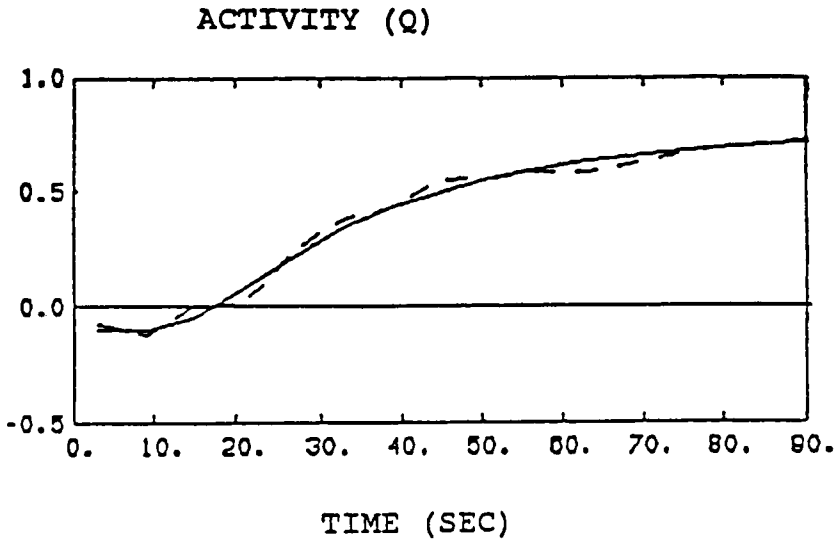


Figure 7.3. Time-Activity Curve Fitting After Identification.

Table 10. Summary of Blood Flow Estimation

Learning rate	Learning shape	Momental term	No. of iterations
0.8	0.9	0.9	1500
No. of layers	Input nodes	Hidden nodes	Output nodes
3	29	8	1
No. of trainng patterns	No. of testing patterns	Average error for training	Average error for testing
8	24	0.0027	0.005

model to estimate blood flow with dynamic PET scanning. Moreover, the use of neural networks, instead of mathematical models, reduces the computation time for determining the myocardial blood flow in a certain region of interest. This provides a method for generation of parametric blood flow images for medical diagnostics.

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE RESEARCH

8.1 Summary

Artificial neural networks trained with back-propagation algorithm have been applied to estimate the composition of chemical streams using spectroscopic data. Different approaches were evaluated to reduce the network training time and estimation error. In Chapter 4, the effect of network architectures on learning convergence were studied. The experimental results in Chapter 6 show that network performance is sensitive to training data organization, the network architecture, and the initial conditions. This supports the interpretation of the variation in network performance for the same training and testing data. The number of hidden layers and hidden nodes are the major factors which affect the network performance. The network with appropriate number of hidden nodes could be trained to achieve a high degree of estimation accuracy. Furthermore, the performance of trained networks on test data that were not contained in the training set demonstrates that these networks utilized general signal features to achieve accurate parameter estimation. In Chapter 5, the sensitivity of network estimation on the relationship between a portion of spectroscopic data and one or more chemical components was studied. The experimental results showed that certain inherent features can be extracted by

the trained networks. The effect of random noise in training patterns on network performance was also analyzed in this research. The method of combining Kohonen and back-propagation networks as a two stage network to estimate chemical compositions was given in Chapter 4. Because of the limitation of the given samples, no further result is presented in this study. The neural network approach can also be applied to the applications in power industry and nuclear medicine problems. This further demonstrates the feasibility of using a artificial neural network to estimate parameter variables for different applications.

8.2 Conclusions

The conclusions based on the study in this research are made mainly in two areas: the study of network structure and the study of network capability. These are summarized as follows:

1. The use of artificial neural networks to estimate concentration of chemical components through spectroscopic signatures provides a high degree of accuracy.

2. Multiple networks, each with single output node, perform better than one network estimating all the chemical components simultaneously.

3. The number of hidden nodes is one of the major factors affecting network performance. Setting the number of hidden nodes equal to the number of training patterns is found to perform well for this composition identification problem.

4. The estimation errors can be reduced by using an ensemble of networks.

5. In the present application the estimation error and training time can be reduced by using networks with two hidden layers.

6. Multi-layer perceptrons (MLP) are noise tolerant due to their intrinsic ability to generalize from taught examples to corrupted version of the original patterns. This is illustrated by the small error standard deviation in the estimated concentrations, generally less than 2% when the standard deviation is less than 3% of signal level.

7. The independent sensitivity analysis of the neural network has demonstrated the correspondences between spectral features and component fractions.

8. Network initial conditions and connection weights are also important factors affecting training time and estimation accuracy. Setting the initial connection weights to proper values is very important for learning convergence. For this purpose, a study of the statistical distribution of network connection weights was performed. However, no valuable clue was found in the current results.

9. The neural network method can be easily extended to process other spectroscopic data problems or similar applications such as lubrication oil analysis, effluent gas analysis, and water chemistry study, etc..

8.3 Recommendations for Future Research

Since the artificial neural networks is a new technology, many of their properties and capabilities are still under investigation. The lack of concrete mathematical proofs and some unsolved problems may raise doubts about the general capability of neural networks. During the study of chemometric data analysis, a large amount of numerical experiments were performed in order to study the feasibility and the advantages of neural network for this type of applications. General guidelines were developed for implementing a back-propagation network structure. Some recommendations for the future research are given below.

1. Application of neural networks approach to other similar problems for further verification of its capability, especially to problems which are difficult to solve by direct methodologies. The purpose of this study is to

find the general capabilities and properties of a back-propagation neural network by certain similar experiments. This study may also help us to address the guideline for selecting optimal network structures.

2. Further theoretical studies on neural network implementation and initial condition optimization. This study should focused on the theoretical basis for selecting optimal number of hidden layers and hidden nodes in each layer, the initial connection weights and learning parameters. Study of different network algorithms can be the subject for future research.

3. Combining expert system and neural networks. The expert system may function as a interface between the user and neural networks. It will provide the rules on selection of network structure, initial conditions, and method to preprocess the given data for network training and testing, as well as to analyze the results of estimation.

4. Combining Kohonen self-organizing and back-propagation neural networks as a two-stage network for processing other large amount of industrial data.

LIST OF REFERENCES

[1] A. A. Garrison, E. C. Muly, M. J. Roberts, and C. F. Moore, "On-line Chemical Composition Analyzer Development," University of Tennessee, Final Report, July 1989.

[2] J. G. Grasselli, M. K. Snavely, and B. J. Bulkin, "Chemical Applications of Raman Spectroscopy," Wiley-Interscience, New York, 1981.

[3] P. M. Fredericks etal, "Materials Characterization Using Factor Analysis of FT-IR Spectra. Part 2: Mathematical and Statistical Considerations," Applied Spectroscopy, Volume 39, Number 2, pp 311-315, 1985.

[4] D. E. Rumelhart and J. L. McClelland, "Parallel Distributed Processing; Explorations in the Microstructure of Cognition," vol. I and II, MIT Press, Cambridge, MA, 1986.

[5] J. Dayhoff, "Neural network Architectures," Van Nostrand Reinhold, N. Y., 1990.

[6] R. H. Silverman and A. S. Noetzel, "Image Processing and Pattern Recognition in Ultrasonograms by Back Propagation," Neural Networks, vol. 3, pp 593-663, 1990.

[7] Y-H. Pao, "Adaptive Pattern Recognition and Neural Networks," Reading, MA, Addison-Wesley, 1989.

[8] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme Recognition Using Time-Delay Neural Networks," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, pp 328-339, March 1989.

[9] R. P. Gorman and T. J. Sejnowski, "Learned Classification of Sonar Target Using a Massively-Parallel Network," IEEE Transaction on Acoustics, Speech, and Signal Processing," vol. 36, pp 1135-1140, 1988.

[10] B. R. Upadhyaya, E. Eryurek and G. Mathai, "Neural Networks for Sensor Validation and Plant Monitoring," Intl. Fast Reactor Safety Meeting, Utah, Vol. 3, pp 349-358, August 1990.

[11] D. J. Burr, "Experiments with a Connectionist Text Reader," Proceedings of the IEEE First Intl. Conf. on Neural Networks, Vol. 4, pp 717-724, San Diego, 1987.

[12] T. J. Sejnowski and C. R. Rosenberg, "Parallel Networks That Learn to Pronounce English Text," Complex Systems, vol. 1, pp 145-168, 1987.

[13] L. F. Miller, G. T. Smith, Y. Wu, and R. E. Uhrig, "Evaluation of Neural Networks for Parameter Identification from Positron Emission Tomography Scans," ANS Transactions page 5, Washington, DC, November 1990.

[14] J. R. Long, V. G. Gregoriou, and P. J. Gemperline, "Spectroscopic Calibration and Quantitation Using Artificial Neural Networks," Anal. Chemical, Vol. 67, pp 791-797, 1990.

[15] L. K. Hansen and P. Salamon, "Neural network Ensembles," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, No. 10, pp 993-1001, October 1990.

[16] F.-S. Tsung and G. W. Cottrell,"A Sequential Adder Using Recurrent Network," IJCNN Intl. Joint Conf. on Neural Networks, Vol. 2, pp 133-139, 1989.

[17] A. Maren, C. Harston, and R. Pap, "Handbook of Neural Computing Applications," Academic Press, Inc., 1990

[18] S. Y. Kung and J. N. Hwang, "An Algebraic Projection Analysis for Optimal Hidden Units Size and Learning Rates in Back-propagation Learning," IEEE Intl. Conf. on Neural Networks, pp 363-370, July, 1988.

[19] S-C. Huang and Y-F. Huang, "Bounds on the Number of Hidden Neurons in Multilayer Perceptrons," IEEE Transaction on Neural Networks, vol. 2, No.1, January, 1991.

[20] R. Hecht-Nielsen, "Theory of the Backpropagation Neural Network," Intl. Joint Conf. on Neural Networks (Washington), vol. I, pp 593-605, 1989.

[21] S. A. Harp, T. Samad, and A. Guha, "Designing Application-Specific Neural Networks Using the Genetic Algorithm," Advances in Neural Information Processing Systems II, pp 447-454, 1990.

[22] J. Hertz, A. Krogh, and R. G. Palmer, "Introductions to the Theory of Neural Computation," Addison-Wesley, Reading, MA, 1991.

[23] G. Cybenko, "Continuous Valued Neural Networks with Two Hidden Layers Are Sufficient," Technical Report, Tufts Univ. Medford, MA, 1988.

[24] A. Lapedes and R. Farber, "How Neural Nets Work," Neural Information Processing Systems, pp 442-456, N. Y. 1988.

[25] D. L. Chester, "Why Two Hidden Layers Are Better Than One," IJCNN-90, Washington, D. C., pp I-265-268, January 1990.

[26] "Predictive Maintenance Primer," EPRI NMAC, NP-7205, April 1991.

[27] "PWR System Manual," Tennessee Valley Authority, 1990.

[28] I. L. Rybalchenko, "Role of Water Chemistry in the Reliable Operation of Nuclear Power Plants," Proceedings of an International Symposium of Water Chemistry and Corrosion Problems of Nuclear Reactor Systems and Components, pp 241-250, International Atomic Energy Agency, Vienna, November, 1982.

[29] Y. Liu, "Quantitative Analysis of PET with Backpropagation Neural Networks," ANS Eastern Regional Student Conf. Transactions, Pennsylvania State University, March 1990.

[30] G. T. Smith, Y. Liu, and B. R. Upadhyaya, "Quantitative Myocardial Blood Flow Determined with Neural Network Analysis," 76th Scientific Assembly and Annual Meeting of Radiological Society of North American, Chicago, November 1990.

APPENDICES

APPENDIX A

GUIDELINES FOR THE VAX VERSION NEURAL NETWORK SOFTWARE

It is important to understand the whole procedure used for training a network. Normally, there are four major steps considered when designing a neural network software. The first step is the preprocessing of data used for network training and testing. This includes data compression, feature extraction, data normalization and training, and testing. The second step involves the design of network structure and choosing network learning parameters. The third step is training the network with selected data set. The finally step is testing the trained network using the data that are not included in training.

In the VAX version of the neural network software used in this study, the data preprocessing process begins by running a program named "read.for". It is a FORTRAN program specially developed for present applications, and can be used to change the original data file to a standard matrix format for network training. Each training pattern takes two rows. The first row presents the network input values, the second row consists of target output values for a back-propagation network training. A data compression process is also performed in this program. Moreover, this program can transfer the target outputs in the original file from percentage to the range [0 - 1].

The next step is to run the normalization program called "norm.for". This program is mainly used for two purposes: 1. To calculate the maximum and minimum values among input vectors. 2. To normalize the input vectors according to the following equation:

$$X_{new}=0.8*\frac{X_{old}-X_{min}}{X_{max}-X_{min}}+0.1$$

Finally, this program generates a standard normalized data file for the back-propagation neural network training.

It is necessary to create two different sets of files for training and testing a network. A program called "gene.for" was designed for this task. This program separates the entire data set into a training set and testing set as desired by the author. For example, if the total data set included 30 patterns, one can select 22 of these patterns for network training and 8 patterns for network testing. Note that the number of training and testing patterns may vary from one application to another, as well as the choice of these patterns.

There is another file named "mul_sing.for" which changes a multiple target output to a single output in training data files. Sometimes, it is desirable to train a network with just single target output (this of course depends on the application). With this program, the training data which contains more than one target outputs will be separated into several training files. However, the input values of the new files remain the same as the original file. For instance, if the original training file contains patterns with 190 inputs points and five target outputs, the "mul_sing" program will create five files, each containing patterns 190 input points and single output.

When the data files are ready for network training, the following step is used to select the structure and learning parameters of the network. This is performed by editing a file named "bpn_mode.dat". A typical format of this file is demonstrated

in Figure A.1. The number of network layers, the number of nodes in each layer, the number of training patterns, the learning file name and other parameters are required to specify in this data file. The "bpn.exe" is a program used for training the network. After running this program, a control panel (Figure A.2) is displayed on the screen which indicates the information about network training procedure, such as training rate and change shape, and the number of training iteration. Sometimes, it is necessary to restart the training process. The decision of restarting a training process can be made when the number of best iteration is far away from the number of training iteration which are indicated on the screen. Also, the behavior of local error of training (Figure A.3) can be used to judge the performance of training. If the error behavior is not desirable or it is not converging, one may try to vary the network connection weights by selecting "nudge weights, jog weights, or kick weights" instead of restarting training process. When the network training is successful (according to the value of local error), one may save the network by selecting "save as" in the control panel and then quit the program.

The last step in the neural network learning procedure is to test the trained network. "bpn_test1.for" is the file used for this purpose. One may use either training data or testing data to test the network and then make comparison of the results for these two cases.

```
++++++++++++++++++++ PARAMETER DEFINITION FILE ++++++++++++++++++++++++++
***** DO NOT CHANGE THE STARTING FIELDS OF THE SPECIFICATIONS ***********
-------------------- NETWORK SPECIFICATIONS ----------------------------
 NUMBER OF LAYERS                    :4
 NODES/LAYER(STARTING AT I/P):-
 LAYER 1 :190
 LAYER 2 :8
 LAYER 3 :14
 LAYER 4 :1
 LAYER 5 :0
------------------- TRAINING  PARAMETERS -------------------------------
   NUMBER OF TRAINING PATTERNS   :22
 NUMBER OF TRAINING ITERATIONS   :10000
          LEARNING COEFFICIENT   :.7
                MOMENTUM TERM    :.9
          INPUT PATTERN LENGTH   :190
         OUTPUT PATTERN LENGTH   :1
   FREQUENCY OF DISPLAY UPDATE   :300
  TYPE OF DISPLAY [1(1P)/2(P)]   :1
  TRANSFER FUNCTION SHAPE[1-5]   :1.0
           DATA SCALING FACTOR   :1.0
WAIT BEFORE STARTING(0=YES/1=NO) :0
```

Figure A.1. A Typical Format of BPN_MODE.DAT

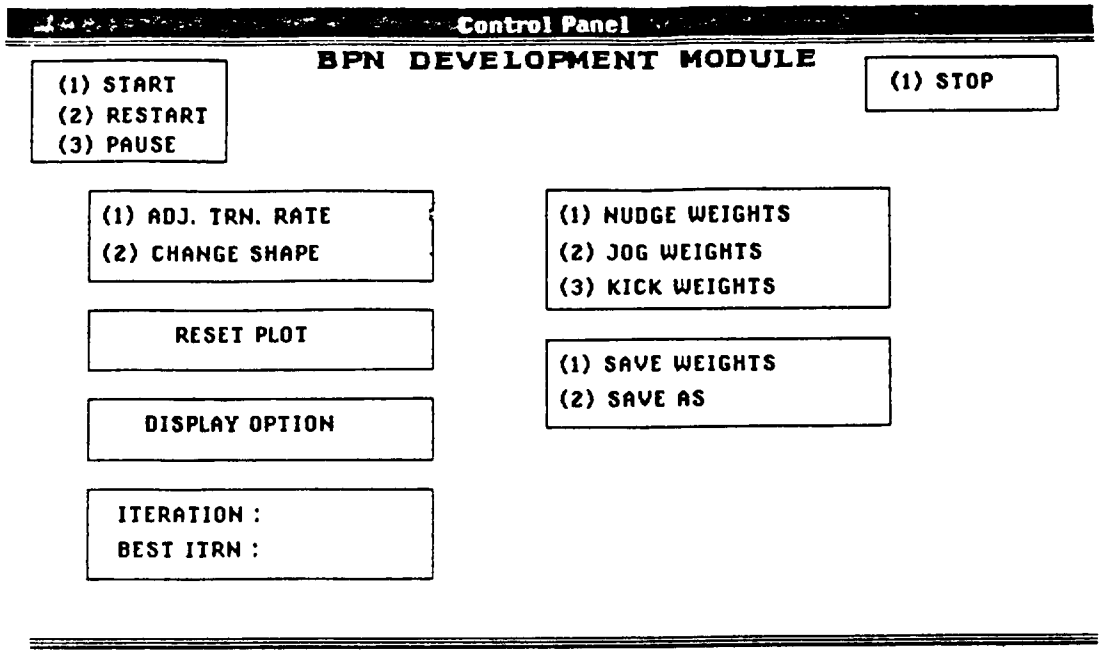


Figure A.2. The Control Panel (monitor) of Network Training

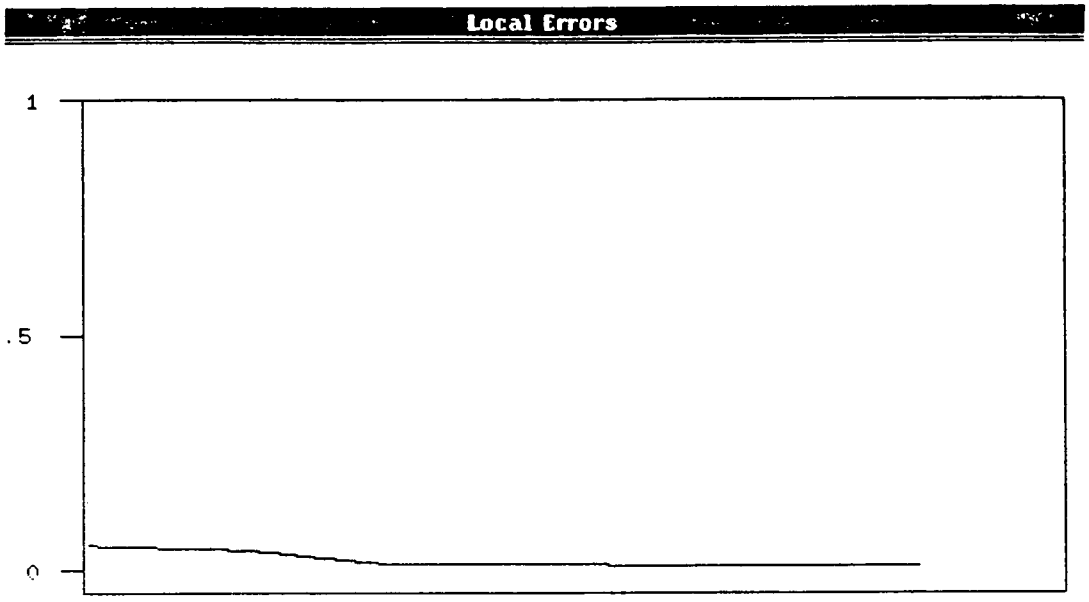


Figure A.3. A Example of Local Error During Training

APPENDIX B

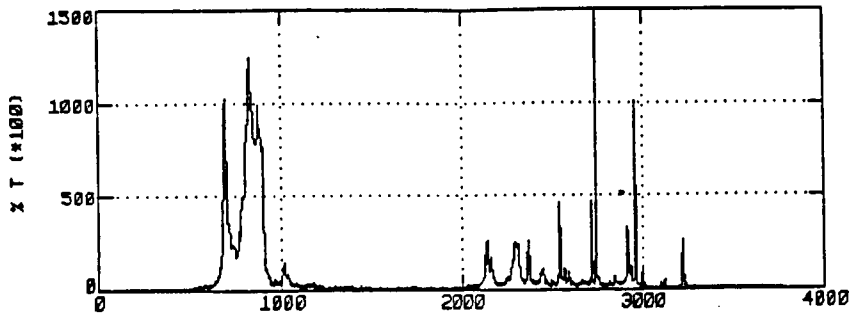
RAMAN SPECTROSCOPIC DATA USED IN THIS RESEARCH

DESCRIPTION OF RAMAN SPECTROSCOPIC DATA

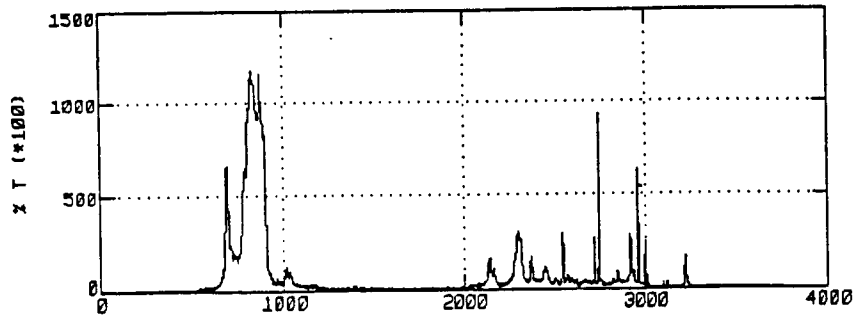
Figure no.	n-hexane	iso-octane	toluene	p-xylene	decane
B.1	10.381	9.983	45.771	12.949	20.915
B.2	14.024	21.943	29.942	11.089	23.003
B.3	22.007	25.866	37.900	13.209	1.019
B.4	11.909	27.479	44.411	13.264	2.938
B.5	13.791	21.624	37.384	6.864	20.337
B.6	14.016	25.974	49.950	5.022	5.038
B.7	18.070	29.897	30.023	5.020	16.990
B.8	17.957	21.947	33.919	7.023	19.154
B.9	9.935	17.853	42.450	12.894	16.868
B.10	14.194	29.809	33.859	14.871	7.266
B.11	22.976	19.073	43.968	12.990	0.993
B.12	29.839	17.853	37.736	13.573	0.999
B.13	17.983	18.016	30.021	10.984	22.996
B.14	29.772	18.093	34.102	11.077	6.956
B.15	25.881	29.862	30.180	7.092	6.986

DESCRIPTION OF RAMAN SPECTROSCOPIC DATA
(continued)

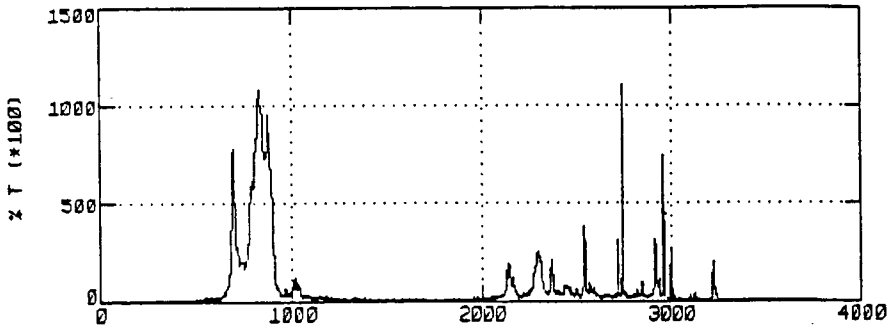
Figure no.	n-hexane	iso-octane	toluene	p-xylene	decane
B.16	25.964	14.018	33.885	5.000	21.133
B.17	21.673	18.916	41.700	5.110	12.601
B.18	9.858	25.937	33.983	9.025	21.196
B.19	21.211	24.807	47.758	3.941	2.283
B.20	14.093	9.920	34.215	8.835	32.937
B.21	10.033	25.646	37.818	5.097	21.405
B.22	12.949	24.766	48.365	10.558	3.361
B.23	17.728	13.918	37.570	9.809	20.975
B.24	9.920	18.260	49.666	7.237	14.917
B.25	23.715	23.777	45.790	4.023	2.696
B.26	29.814	18.177	29.959	12.722	9.327
B.27	10.078	14.207	33.461	15.282	26.972
B.28	18.260	14.007	38.155	4.909	24.669
B.29	15.827	28.184	42.569	12.394	1.025
B.30	13.923	9.975	41.833	11.312	22.956


1500
1000
500
0
% T (*100)
0
1000
2000
3000
4000

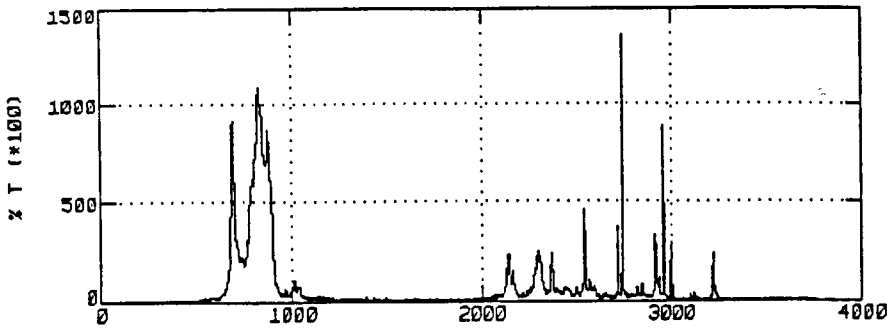
B.1


1500
1000
500
0
% T (*100)
0
1000
2000
3000
4000

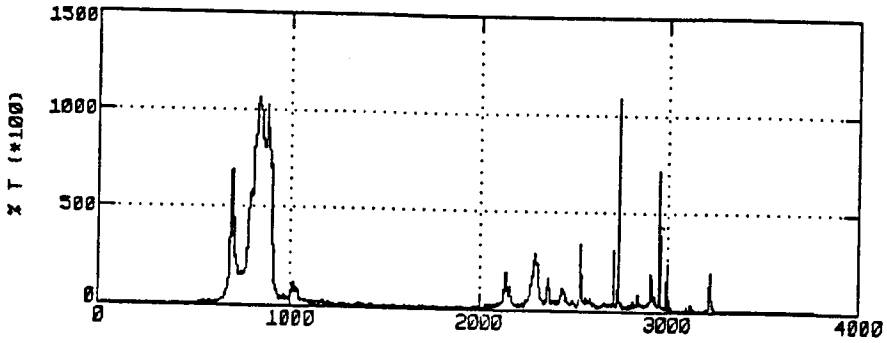
B.2


1500
1000
500
0
% T (*100)
0
1000
2000
3000
4000

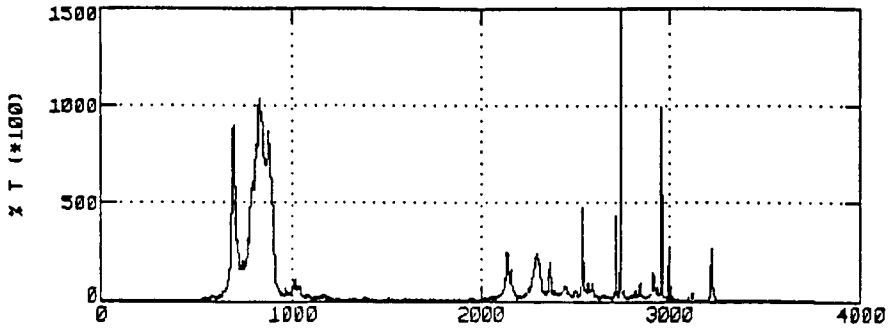
B.3


1500
1000
500
0
% T (*100)
0
1000
2000
3000
4000

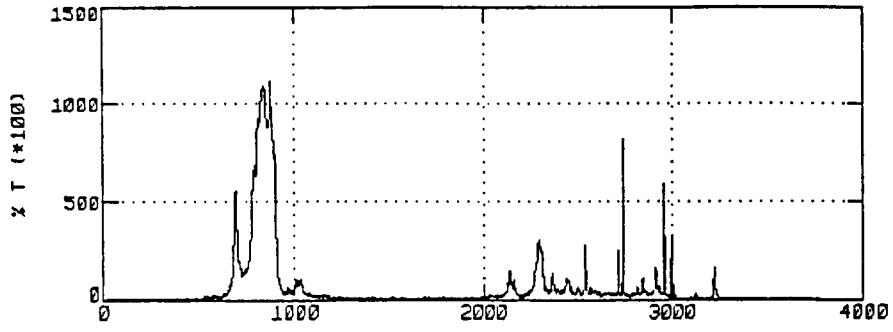
B.4


1500
1000
500
0
% T (*100)
0
1000
2000
3000
4000

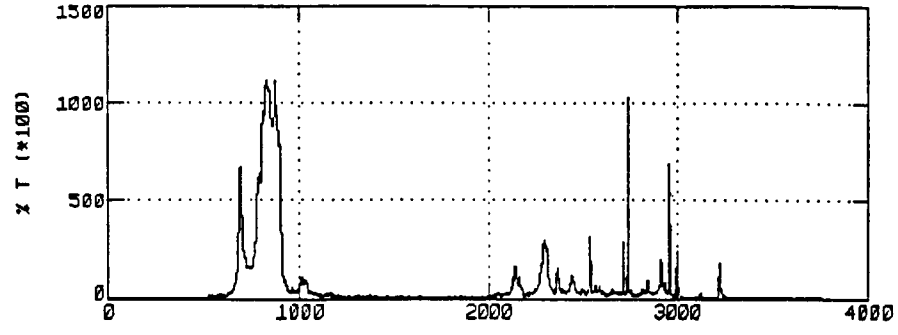
B.5


1500
1000
500
0
% T (*100)
0
1000
2000
3000
4000

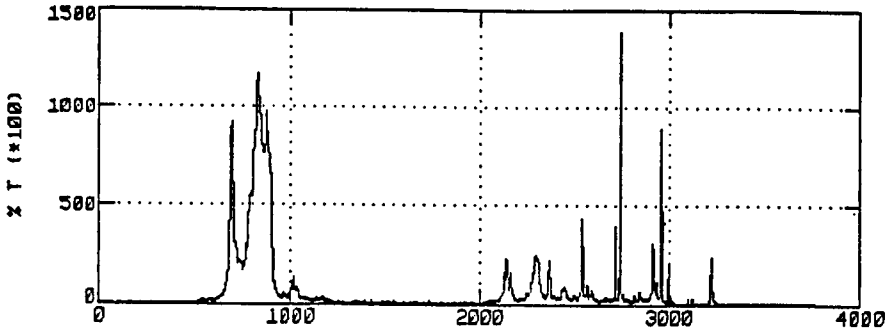
B.6


1500
1000
500
0
% T (*100)
0
1000
2000
3000
4000

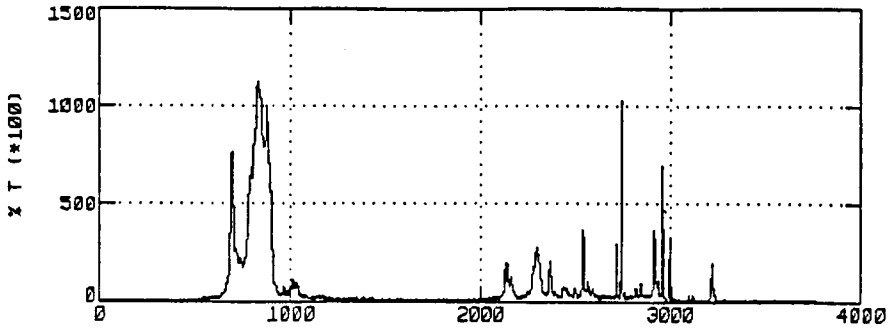
B.7


1500
1000
500
0
% T (*100)
0
1000
2000
3000
4000

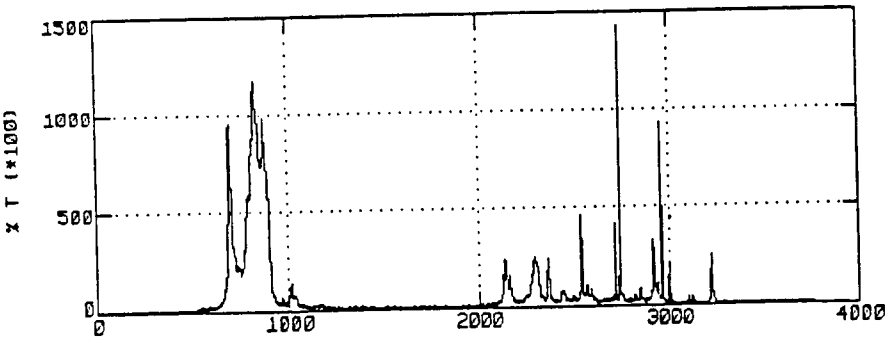
B.8



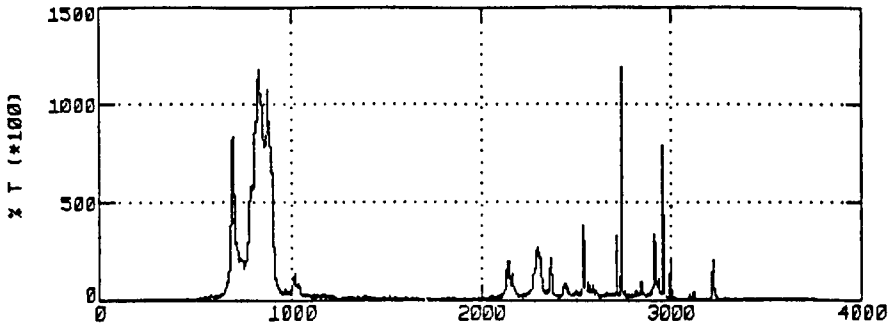
B.9



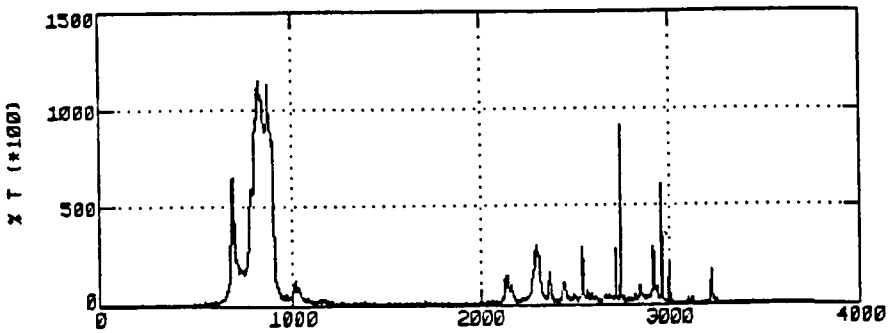
B.10



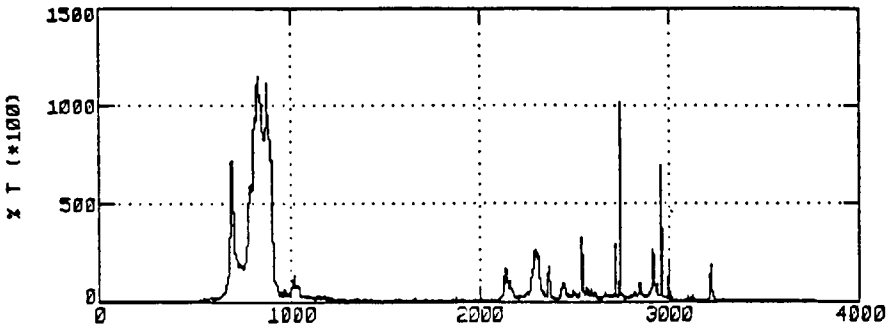
B.11



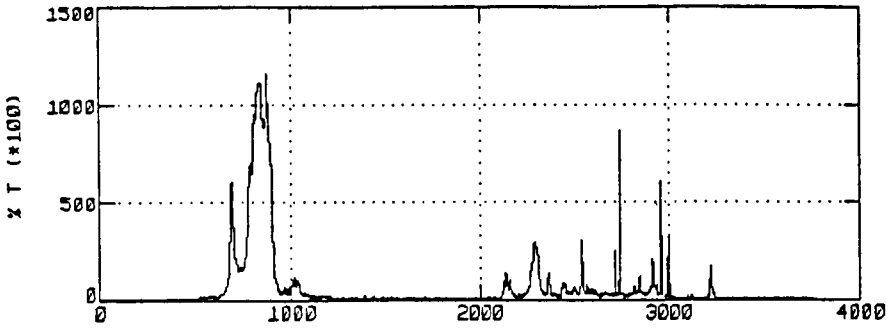
B.12



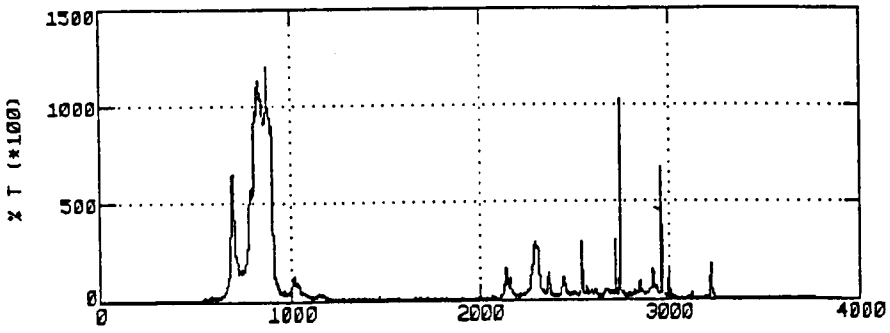
B.13



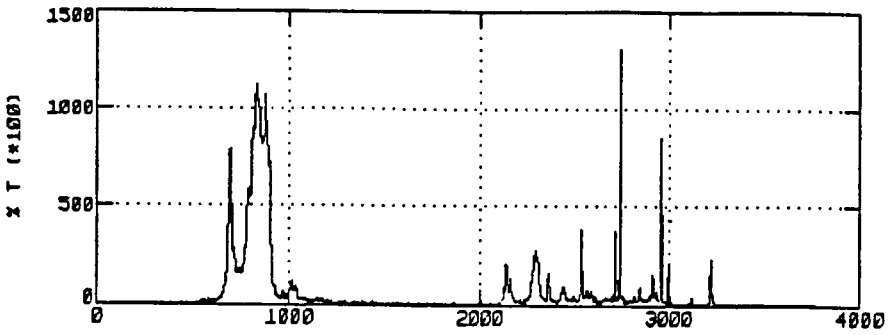
B.14



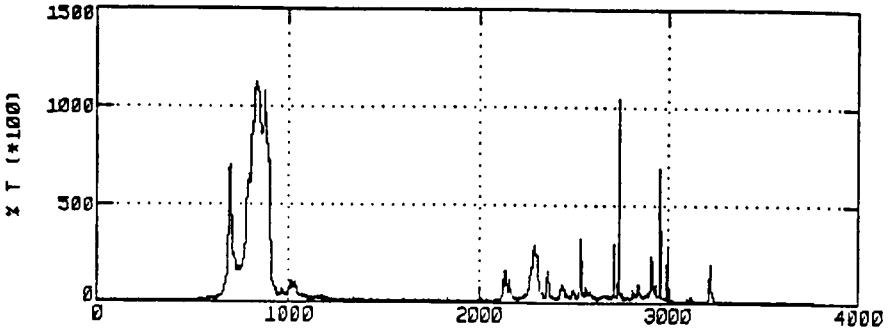
B.15



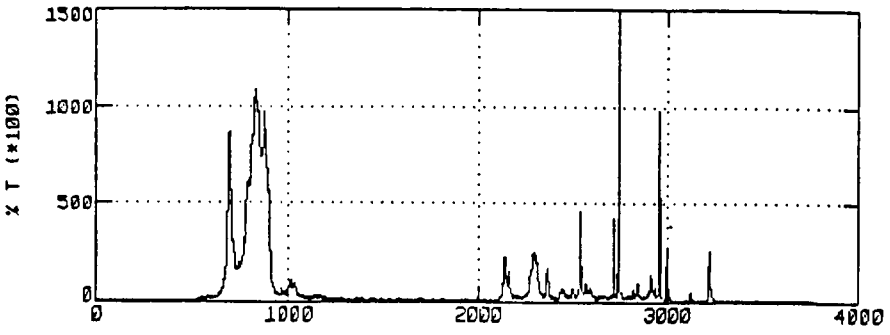
B.16



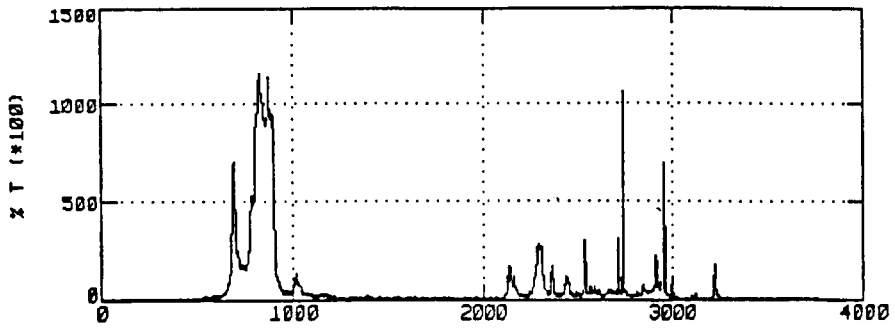
B.17



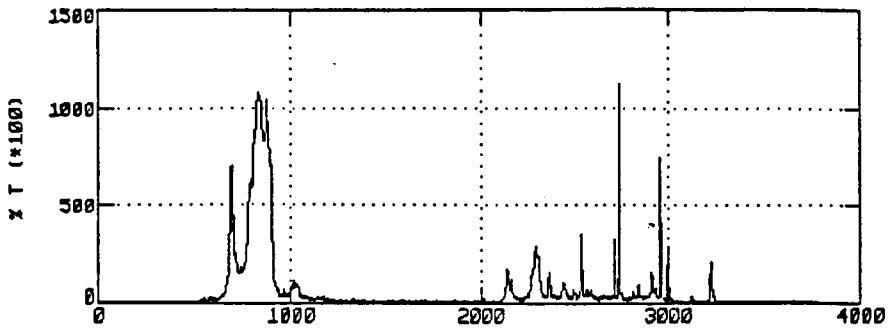
B.18



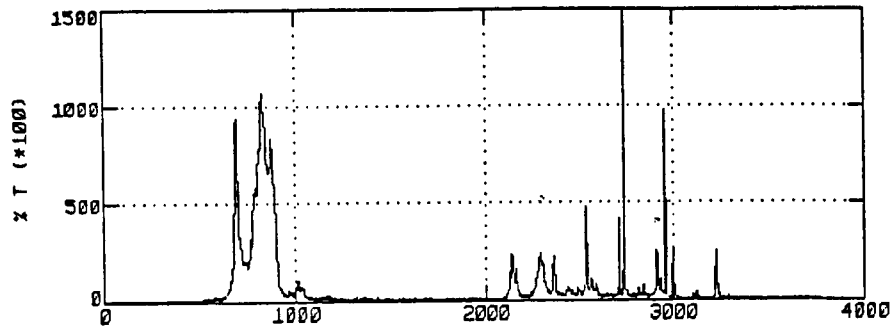
B.19



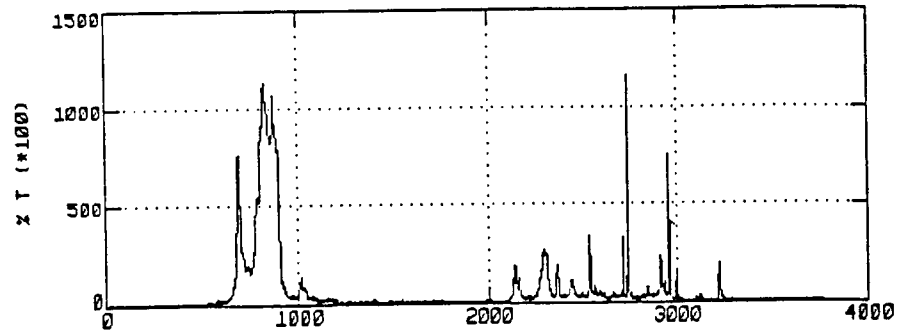
B.20



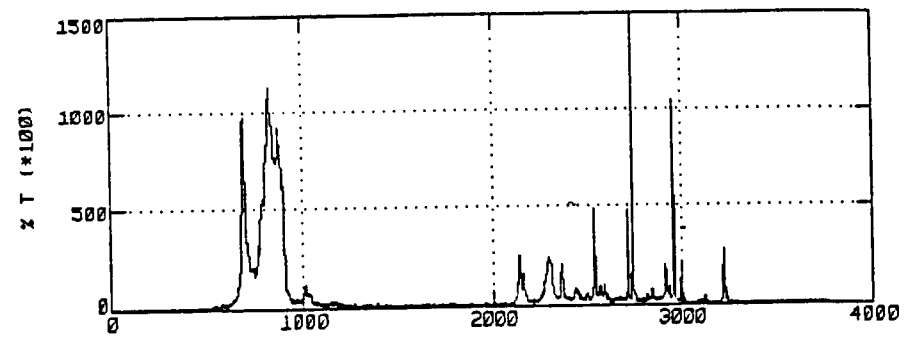
B.21



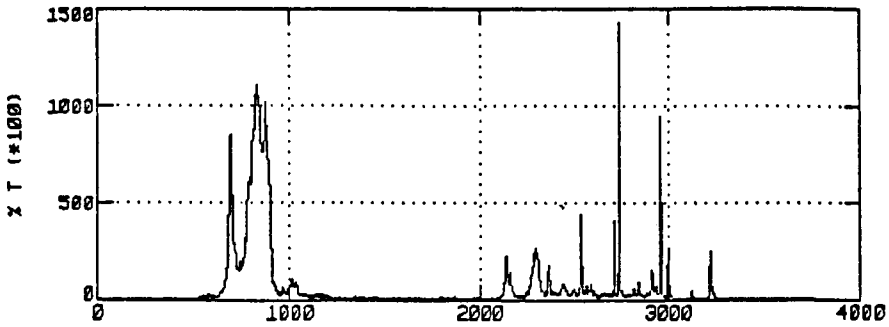
B.22



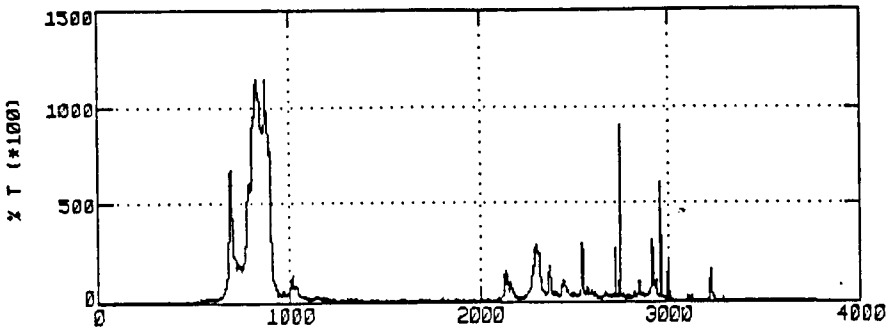
B.23



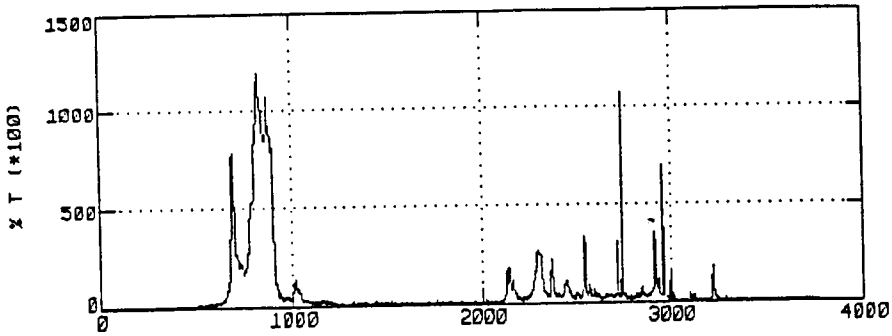
B.24



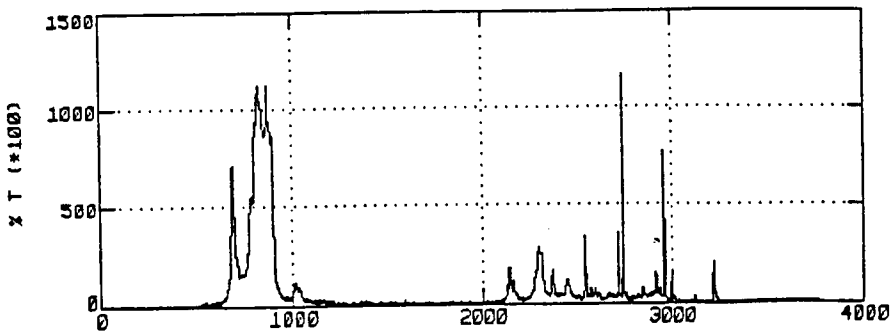
B.25



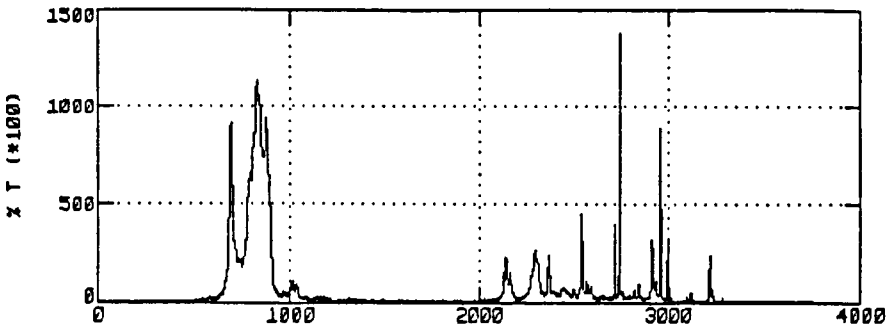
B.26



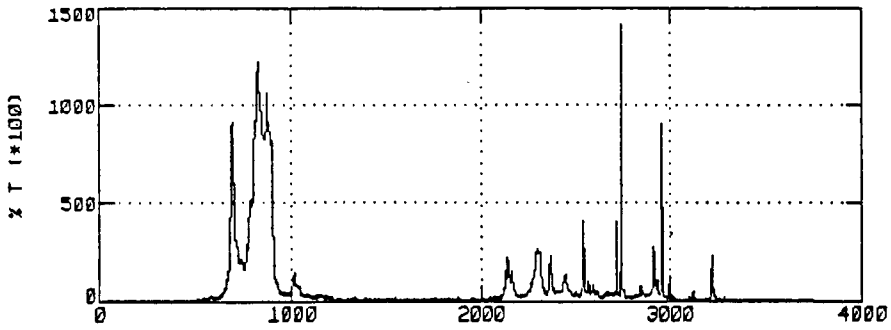
B.27



B.28


1500
1000
500
0
% T (*100)
0
1000
2000
3000
4000

B.29


1500
1000
500
0
% T (*100)
0
1000
2000
3000
4000

B.30

APPENDIX C

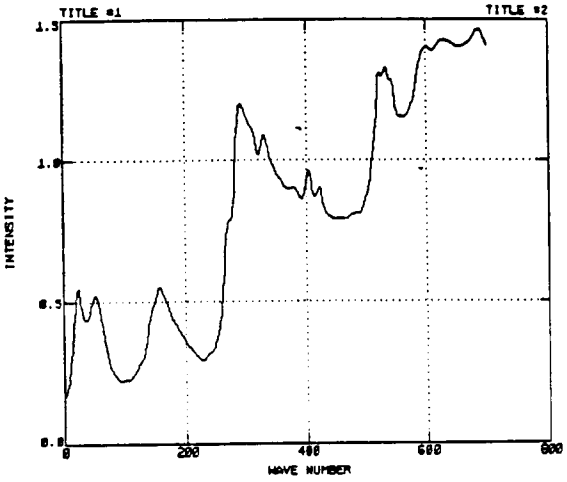
NEAR INFRARED SPECTROSCOPIC DATA USED IN THE RESEARCH

DESCRIPTION OF NEAR INFRARED SPECTROSCOPIC DATA

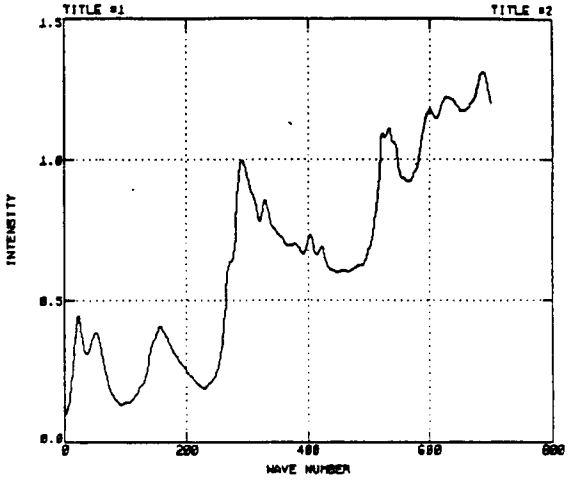
Figure no.	Component X	Component Y	Component Z
C.1	14.5	22.7	62.8
C.2	8.09	23.5	68.4
C.3	9.67	19.0	71.4
C.4	18.7	19.5	61.8
C.5	10.6	24.2	65.3
C.6	10.7	21.9	67.4
C.7	10.7	21.9	67.4
C.8	10.7	21.9	67.4
C.9	10.7	21.9	67.4
C.10	10.7	21.9	67.4
C.11	10.7	21.9	67.4
C.12	10.7	21.7	67.4
C.13	10.7	21.7	67.4
C.14	10.7	21.7	67.4
C.15	10.7	21.7	67.4
C.16	6.96	18.0	75.0

DESCRIPTION OF NEAR INFRARED SPECTROSCOPIC DATA
(continued)

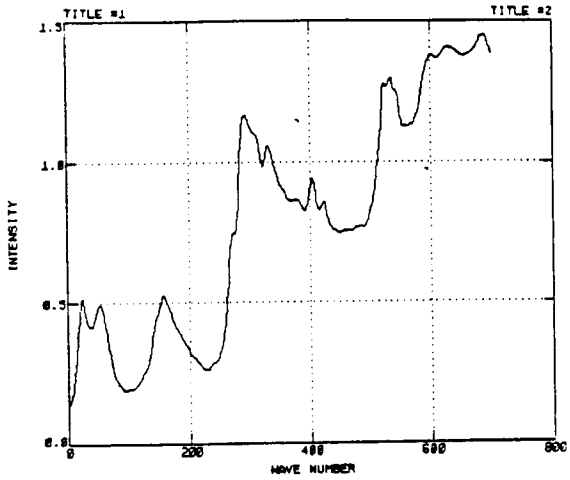
Figure no.	Component X	Component Y	Component Z
C.17	4.75	24.4	70.9
C.18	8.29	25.6	69.1
C.19	8.29	2.26	69.1
C.20	8.29	2.26	69.1
C.21	8.29	2.26	69.1
C.22	13.2	20.9	65.9
C.23	10.9	22.8	66.3
C.24	10.9	22.8	66.3
C.25	10.9	22.8	66.3
C.26	10.9	22.8	66.3
C.27	7.66	18.1	84.2
C.28	8.53	20.2	71.3
C.29	15.3	20.0	64.8
C.30	6.88	27.2	65.9
C.31	18.7	18.7	62.4


TITLE #1
TITLE #2
INTENSITY
WAVE NUMBER

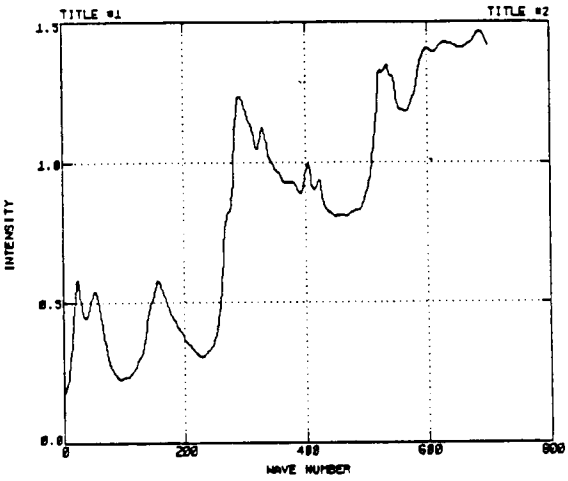
C.1


TITLE #1
TITLE #2
INTENSITY
WAVE NUMBER

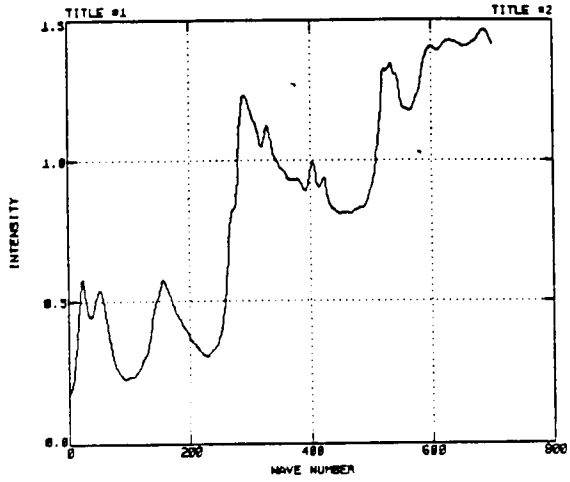
C.2


TITLE #1
TITLE #2
INTENSITY
WAVE NUMBER

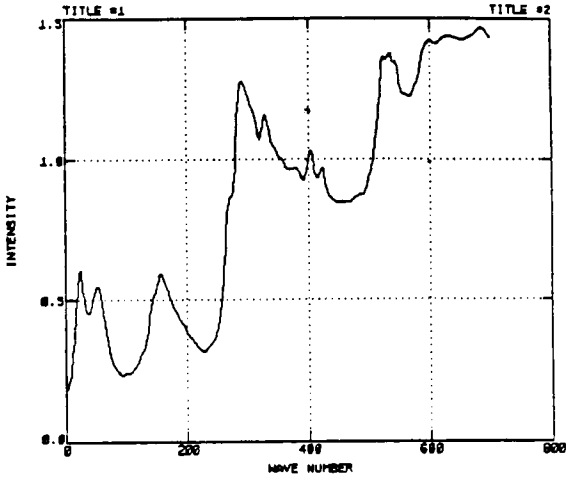
C.3


TITLE #1
TITLE #2
INTENSITY
WAVE NUMBER

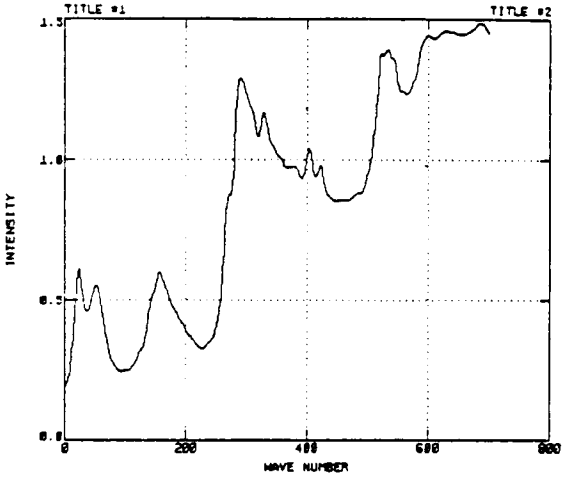
C.4


TITLE #1
TITLE #2
INTENSITY
WAVE NUMBER

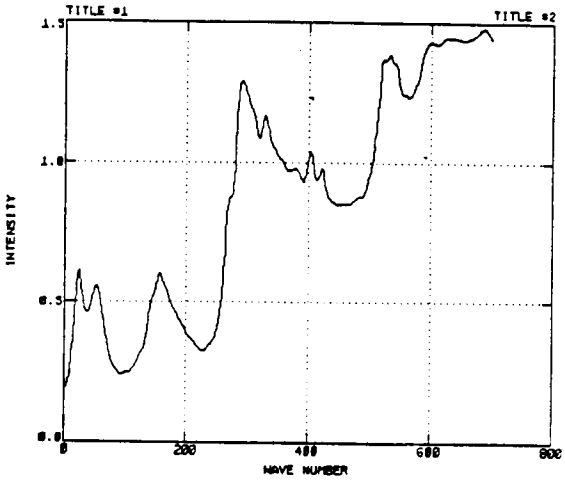
C.5


TITLE #1
TITLE #2
INTENSITY
WAVE NUMBER

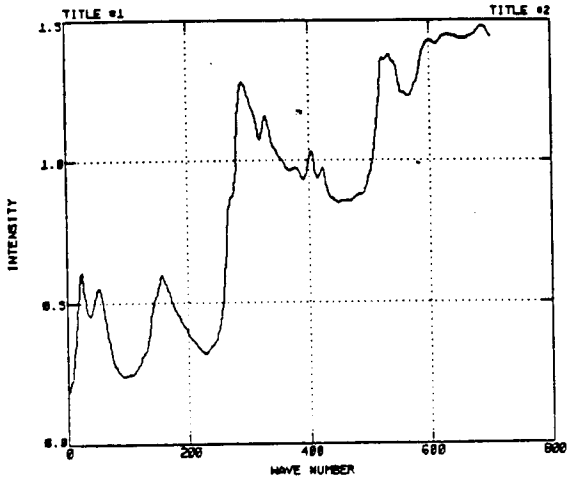
C.6


TITLE #1
TITLE #2
INTENSITY
WAVE NUMBER

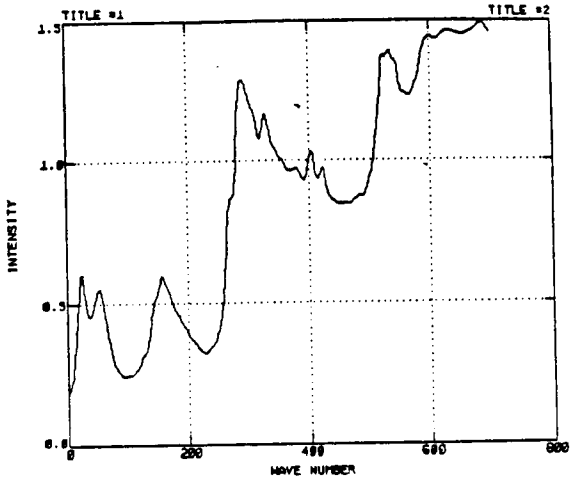
C.7


TITLE #1
TITLE #2
INTENSITY
WAVE NUMBER

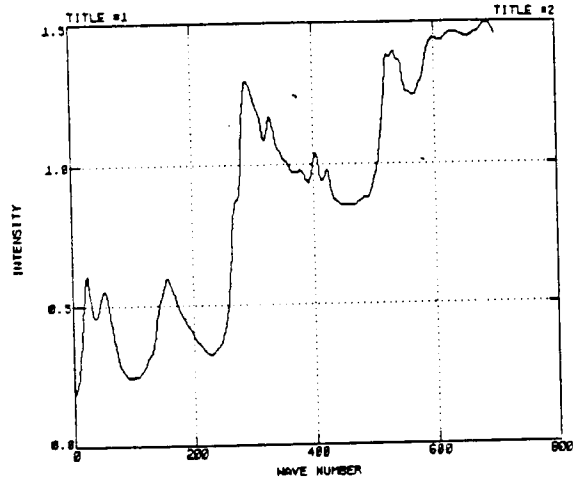
C.8



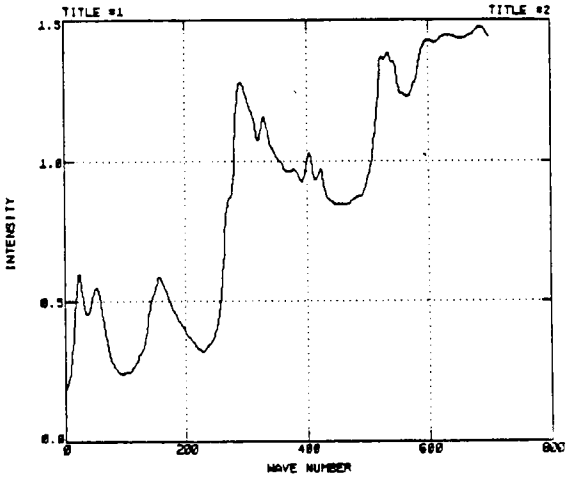
C.9



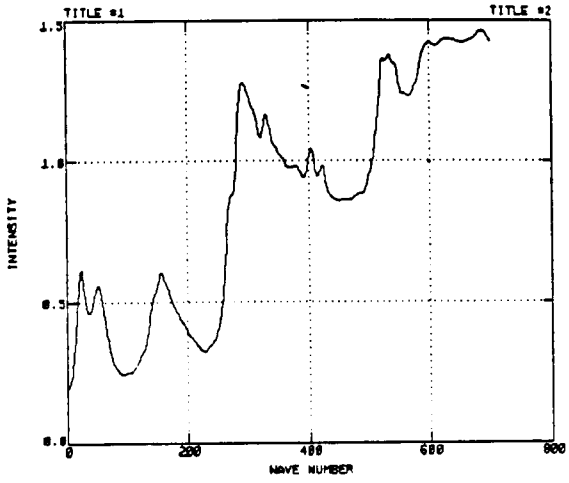
C.10



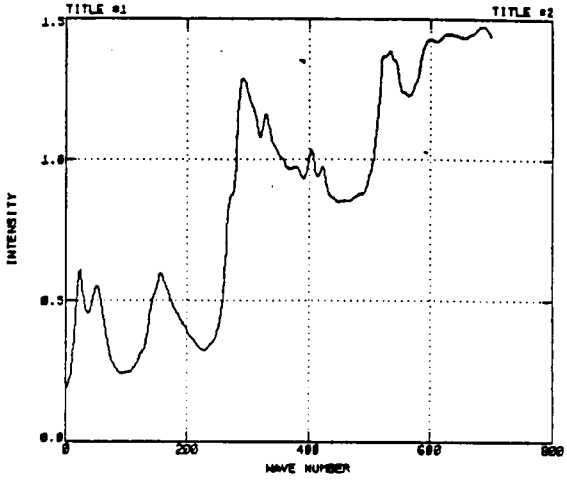
C.11



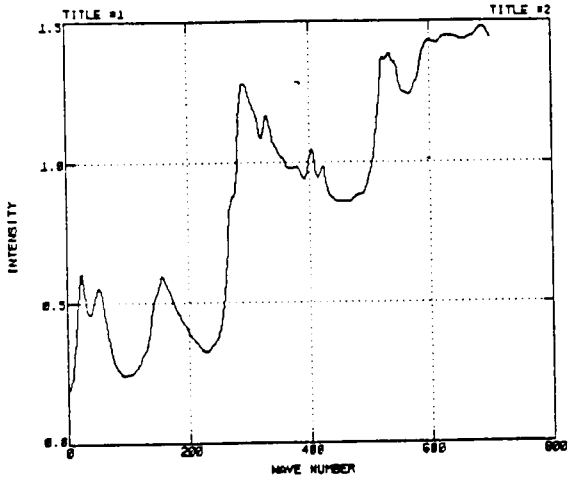
C.12



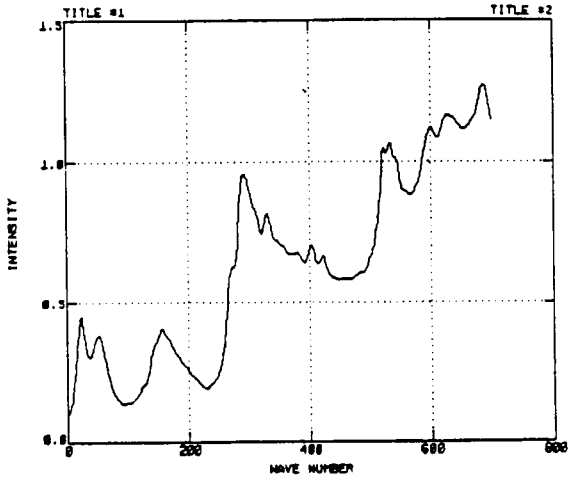
C.13



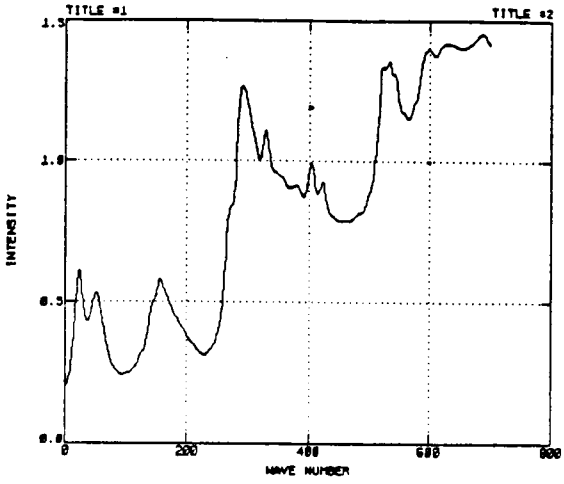
C.14



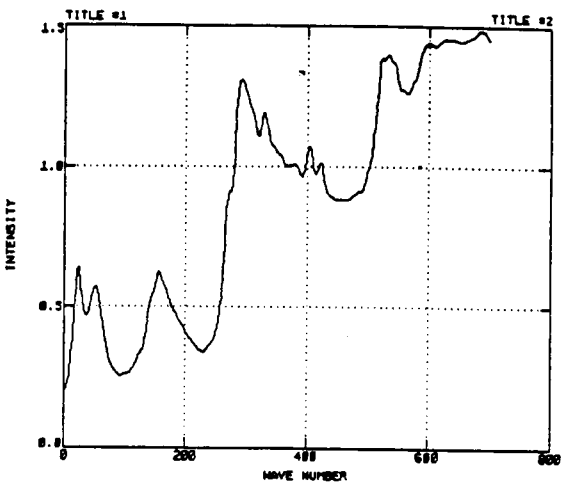
C.15



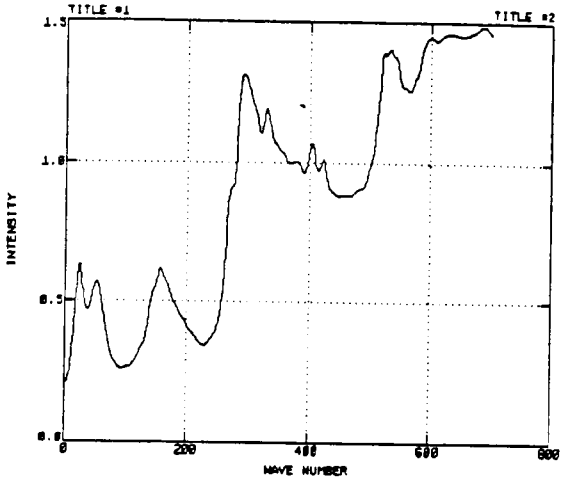
C.16



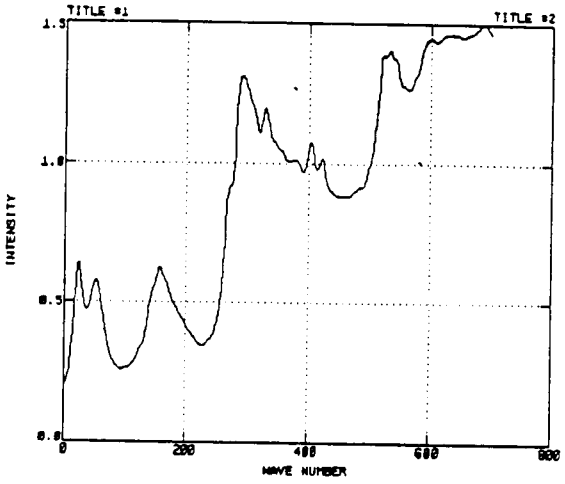
C.17



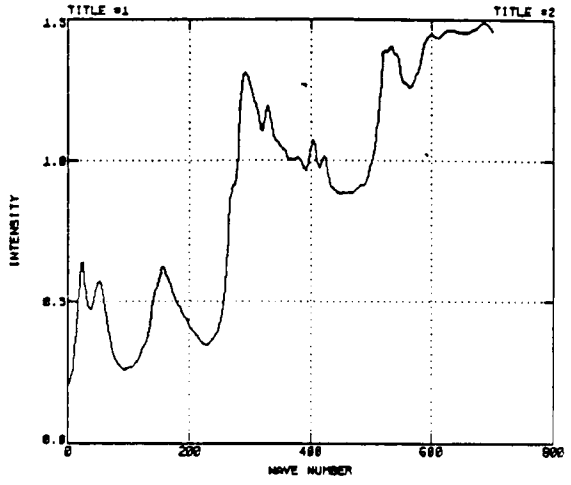
C.18



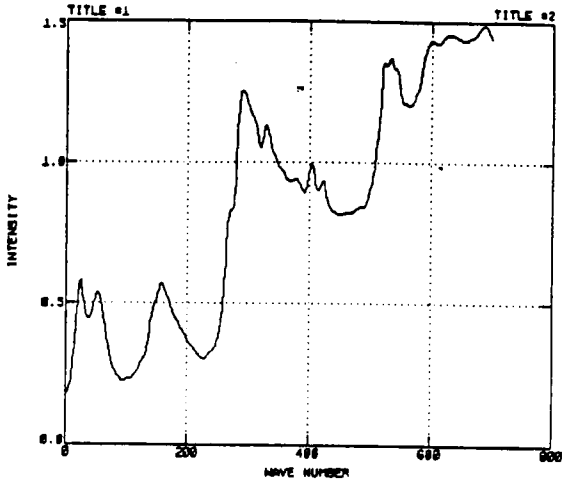
C.19



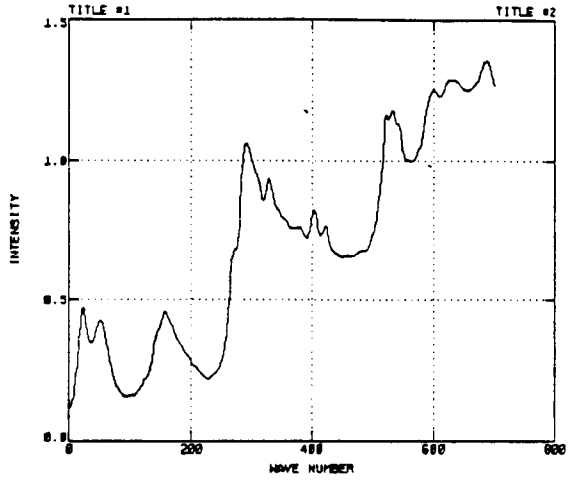
C.20



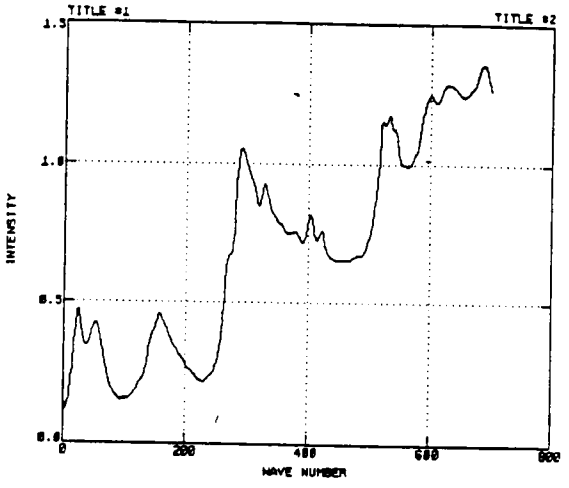
C.21



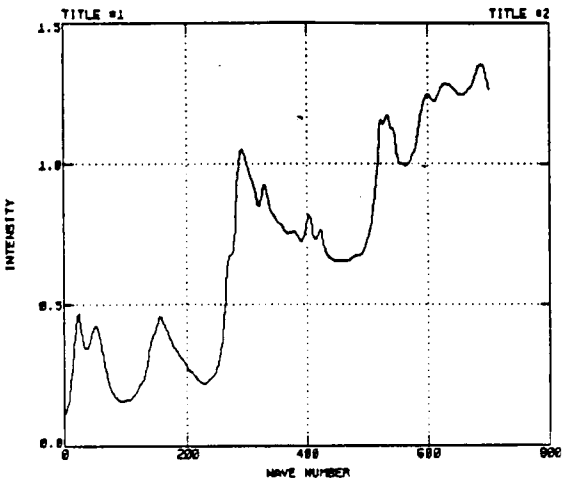
C.22



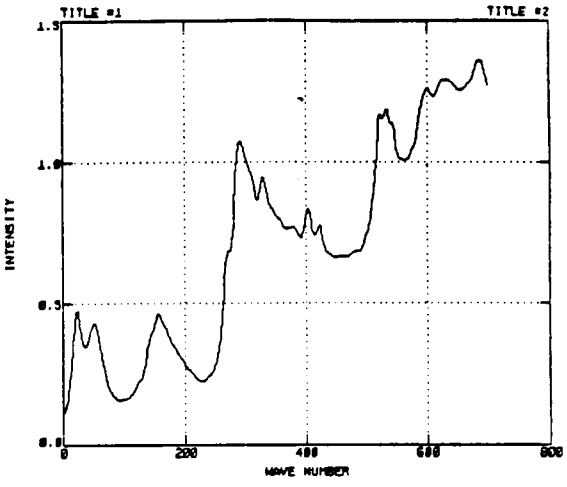
C.23



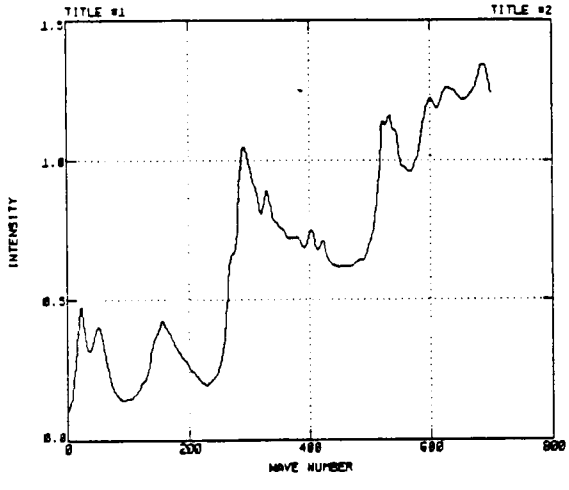
C.24



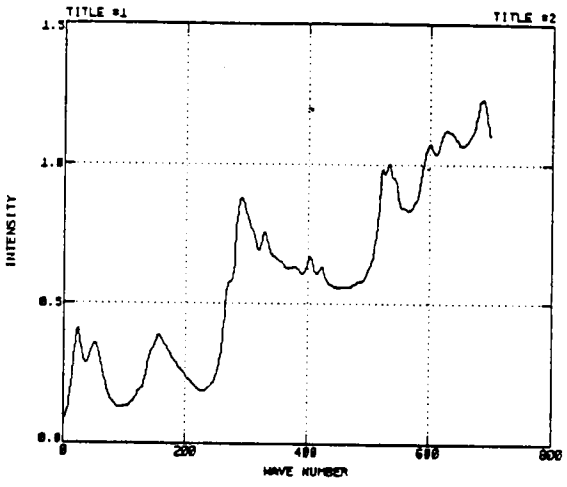
C.25



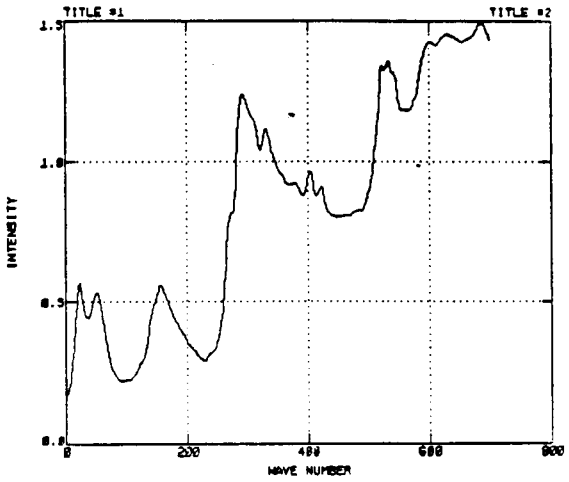
C.26



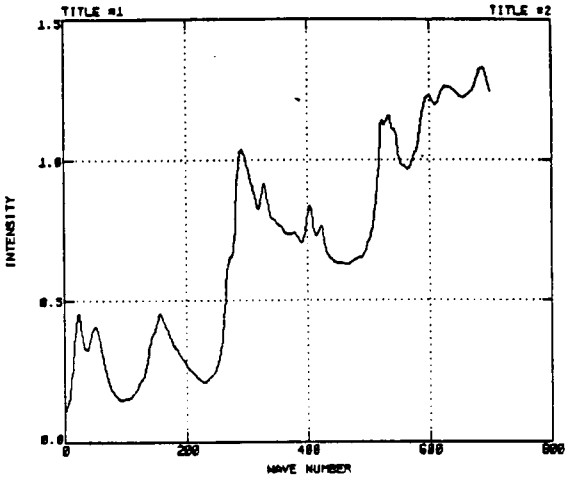
C.27



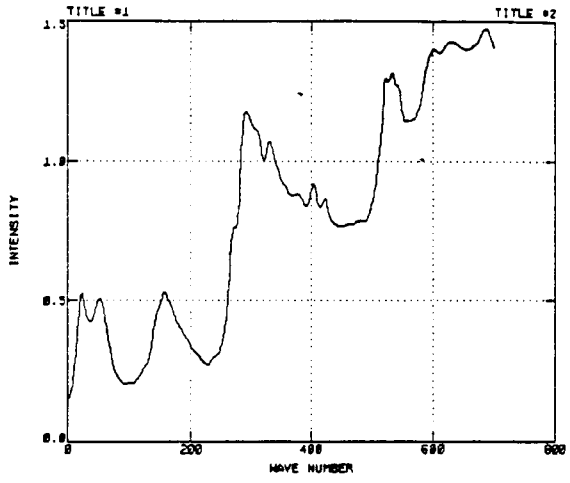
C.28


TITLE #1
TITLE #2
INTENSITY
WAVE NUMBER

C.29


TITLE #1
TITLE #2
INTENSITY
WAVE NUMBER

C.30


TITLE #1
TITLE #2
INTENSITY
WAVE NUMBER

C.31

VITA

Ying Liu was born in Beijing, the people's Republic of China on September 28, 1962. She graduated from Xiangli High School in Jingshi, Hunan in July 1979. Thereafter, She enrolled in Heilongjiang Commercial Institute where she received a Bachelor of Science degree in Electrical Engineering. In 1987, she became a graduate student at China Institute of Mining, Beijing, and studied computer simulation.

She received a graduate teaching assistantship at The University of Tennessee, Knoxville in 1989 and began the study of Nuclear Engineering towards a Master of Science degree.