Neural Network Analysis of Chemical Compounds in Nonrebreathing Fisher-344 Rat Breath

THESIS

Robert E. Sackett Jr.
Captain, USAF

AFIT/GEE/ENG/95D-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GEE/ENG/95D-02

Neural Network Analysis of Chemical Compounds in Nonrebreathing Fisher-344 Rat Breath

THESIS

Robert E. Sackett Jr.
Captain, USAF

AFIT/GEE/ENG/95D-02

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U. S. Government.

AFIT/GEE/ENG/95D-02

Neural Network Analysis of Chemical Compounds in Nonrebreathing Fisher-344 Rat Breath

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Engineering and Environmental Management

Robert E. Sackett Jr., B.S. Electrical Engineering

Captain, USAF

December 1995

Acknowledgements

First of all, I would like to thank God for his grace and seeing me through this project. I would like to thank Dr. Steven K. Rogers for not only his excellent passing on Tuesdays and Thursdays, but also for his totally superior instruction and guidance. He is unequivocally the best instructor I've ever had the privilege to know. For excellent advice on analyzing the rat breath data, I thank Dr. Marty Desimio. Thanks to the other members of my committee, Dr. Charles Bleckmann, Dr. Matt Kabrisky, and Dr. Dennis Ruck. Thanks to my Hawkeye lab partners, Dave Schuchardt and Steve Pellissier, for hooking me up daily on countless problems. Thanks to Lem Myers, Curtis Martin, and Bill Polakowski for all of their help and excellent code. A special thanks to LCol Lyon for making this all possible.

Most of all, I would like to thank the two people most precious to me: my beautiful wife, Lisa, and new baby girl, Lauren. You absolutely mean the world to me and you deserve much more recognition than just an acknowledgement in this thesis for your continuous support and constant love during the last 18 months. Thank you, Lisa!

Robert E. Sackett Jr.

Table of Contents

List of Figures

List of Tables

Abstract

This research applies statistical and artificial neural network analysis to data obtained from measurement of organic compounds in the breath of a Fisher-344 rat. The Research Triangle Institute (RTI) developed a breath collection system for use with rats in order to collect and determine volatile organic compounds (VOCs) exhaled. The RTI study tested the hypothesis that VOCs, including endogenous compounds, in breath can serve as markers to exposure to various chemical compounds such as drugs, pesticides, or carcinogens normally foreign to living organisms. From a comparative analysis of chromatograms, it was concluded that the administration of carbon tetrachloride dramatically altered the VOCs measured in breath; both the compounds detected and their amounts were greatly impacted using the data supplied by RTI. This research will show that neural network analysis and classification can be used to discriminate between exposure to carbon tetrachloride versus no exposure and find the chemical compounds in rat breath that best discriminate between a dosage of carbon tetrachloride and either a vehicle control or no dose at all. For the data set analyzed, 100 percent classification accuracy was achieved in classifying two cases of exposure versus no exposure. The top three marker compounds were identified for each of three classification cases. The results obtained show that neural networks can be effectively used to analyze complex chromatographic data.

Neural Network Analysis of Chemical Compounds in Nonrebreathing Fisher-344 Rat Breath

I. Introduction

1.1 Background

Pattern recognition principles/techniques can be used to analyze a multitude of environmental problems. Applications include classifying bacterial species from mass spectrometry (8), identifying phytoplankton from flow cytometry (5), and identifying different classes of jet fuel from gas chromatography (11). The specific environmental problem to addressed in this thesis has been posed and studied by Dr. James H. Raymer of the Research Triangle Institute (RTI) (16).

RTI developed a breath collection system for use with rats in order to collect volatile organic compounds (VOCs) in their breath (16). The VOCs were analyzed by RTI using thermal desorption/gas chromatography with flame ionization or mass spectrometric detection for three cases: after a specific dosage level of carbon tetrachloride had been injected, after a vehicle control (VC) dose had been injected, and after no dosage had been administered. RTI's study tested the hypothesis that VOCs in breath can serve as markers to exposure to various chemical compounds normally foreign to living organisms such as drugs, pesticides, or carcinogens. From a qualitative analysis of the chromatograms for each of the three cases discussed above, RTI concluded that the administration of carbon tetrachloride dramatically altered the VOCs measured in breath and the concentration of a large variety of compounds was elevated.

From the data supplied by RTI, this thesis will show that neural network analysis and classification can be used to find the compounds in breath that best discriminate between a

dosage of carbon tetrachloride and either a VC dose or no dose at all. Figure 1 provides an illustrative overview of the research performed in this thesis.

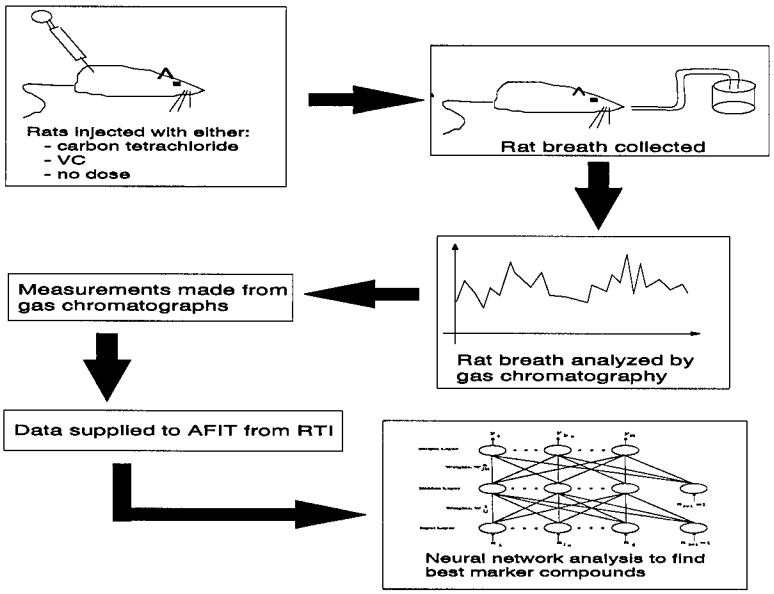

Figure 1. Overview of Research

1.2 Problem Statement

Investigate the statistical and neural network processing of rat breath data to determine the Bayes accuracy for classification of a particular dosage condition and feature saliency of chemical compounds in discriminating a dosage condition. Find the compounds in breath that best discriminate between a dosage of carbon tetrachloride and either a VC dose or no dose at all.

1.3 Research Objectives

Determine how difficult it is to classify a specific dosage condition using the rat breath data, i.e. what is the estimated Bayes error rate? Determine which chemical compounds in the rat breath provided the best discrimination between dosage conditions (none, VC, and carbon tetrachloride).

1.4 Scope

This research first investigates the techniques used to bound the Bayes error rate for a specific data set. Statistical techniques are employed to bound the Bayes error rate for rat breath for each type of classification. Once the Bayes error bound is found, it is used to get insight into the bounds that an artificial neural network (ANN) should reach and whether the current feature set is acceptable. The ANN classification is performed on the dosage condition and is analyzed in a pairwise fashion for three cases. The three cases of classification are 1) a carbon tetrachloride dose is classified with a VC dose, 2) a carbon tetrachloride dose is classified versus a no dose and 3) a VC dose is classified with a no dose. Forward sequential selection techniques and a feature saliency metric will be used to provide insight into which chemical compounds found in rat breath best contribute to the discrimination between dosage conditions.

1.5 Approach

The approach taken in this thesis is composed of four steps. The first step is to implement the techniques of bounding the Bayes error rate presented by Fukunaga and Hummels (7) and Martin (12). The second step of the approach is to train and test a neural network in classification of dosage levels based on the obtained Bayes error bound. The third step is to use forward sequential selection techniques and neural network classification to determine which chemical compounds best discriminate between dosage levels. The fourth step is to utilize a feature saliency metric to validate the results obtained using the forward sequential selection techniques.

1.6 Overview of Thesis

Chapter II provides a background of the artificial neural network used and the techniques associated with bounding the Bayes error and feature saliency. Chapter III describes the rat breath data, the methodology of the experimentation, and presents the results as each individual method is presented. Chapter IV provides a summary of the results and presents

the conclusions of this research. Appendix A presents derivations of learning laws for the Multilayer Perceptron and Appendix B presents techniques to increase the convergence of the gradient descent search of the MLP. Appendix C provides a derivation of the Ruck saliency metric using the notation presented in Chapter II. Appendix D provides a legend of the chemical compounds abbreviated in Chapter III. Appendix E provides the code to compute the Ruck saliency metric.

II. Theory

2.1 Chapter Overview

In this chapter, the relevant theory utilized in this thesis will be presented. Specifically, the topics to be presented include the multilayer perceptron, Bayes decision theory, Bayes error rate bounding, and feature selection.

2.2 Introduction to the Multilayer Perceptron

There are several options to be considered when faced with a pattern recognition problem. One option to be considered is whether to use a statistical pattern recognition scheme or an artificial neural network. An artificial neural network, specifically the Multilayer Perceptron (MLP), is considered because of its diversity and ability to classify data that is not linearly separable (17). To illustrate example data that are not linearly separable, consider data in the form of the binary logic operator, exclusive OR (XOR). The XOR data can be viewed from a geometric point of view as in Figure 2. Assume that data would fall into either class 0 or class 1 and that the two input features are labeled X_1 and X_2.

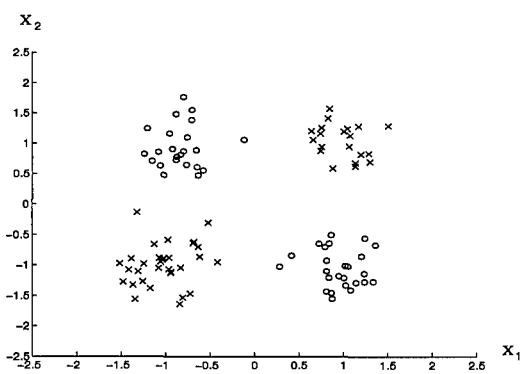

Figure 2. XOR Data

It is obvious from analyzing the plot of the XOR data that there is not a line that will separate class 0 from class 1. Hence, the XOR data are not linearly separable. The MLP has

no problem classifying the XOR data (4). The architecture and function of the MLP covered in the next section provides insight into why an MLP can easily classify XOR data.

2.3 Architecture and Function of the MLP

The architecture of the multilayer perceptron (MLP) is shown in Figure 3 (18).

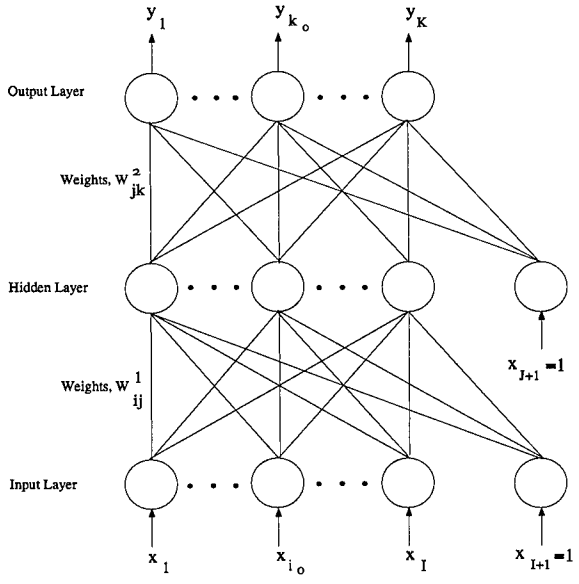

Figure 3. Multilayer Perceptron

As can be seen in Figure 3, each input is weighted and then the weighted inputs are summed at the nodes in the hidden layer and bias term X_{I+1} is added. The advantage of adding the bias term is that the hyperplanes constructing the decision surface are not restricted to pass through the origin. The resulting sum is then run through a nonlinear transformation. See Figure 4 to analyze a hidden layer single node. Note that $X = [X_0, X_1, ..., X_{N-1}, 1]$ and $W = [W_0, W_1, ..., W_N]$. The transformation is usually either linear or sigmoidal. An example of a sigmoidal transformation is shown in Figure 5.

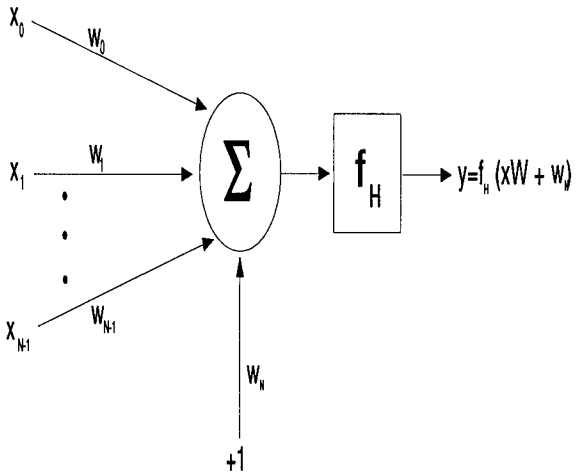

Figure 4. Node Structure

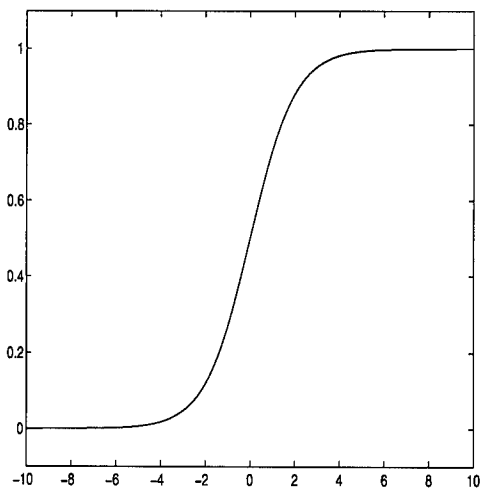

Figure 5. Sigmoid Function

The procedure at the hidden layer is then repeated at the output layer with a different set of weights. For the MLP to be trained to classify, a learning law for each set of weights must be found that is dependent on the nonlinear transformation. The error is defined below:

$$E = \frac{1}{2}\sum_{k=1}^{K}(d_k - y_k)^2$$

Where, d_k is the desired output, k is the number of outputs, and y_k is the actual output. The weights must be updated using one of many existing training rules. A popular technique is the backward error propagation technique, or backprop (17). The generalized learning law for backprop is shown below:

$$W^+ = W^- - \eta\frac{\partial E}{\partial W}$$

Where, W^+ is the updated weight, W^- is the old weight, and η is a constant. Notice that this technique is based on gradient descent in the weight space over an error surface that is created by a sum of the squared error at each output node (19).

Several different combinations of transformations at both the hidden and output layer can be made. For instance, a MLP could have a linear transformation at the output layer and a sigmoid transformation at the hidden layer (1). Depending on the combination, the derivation of the learning laws for each layer will differ. Four combinations of transformations at both layers are considered and derived in Appendix A.

2.4 Introduction to Bounding the Bayes Error Rate

In real-world problems of pattern recognition, accuracy of classification is generally used as a measuring stick to determine how well a particular system performs. An element of error always exists in classification unless the problem is trivial and 100 percent classification accuracy is always achieved. To achieve a minimum probability of error, a classifier must be designed to have an error rate that matches the minimum achievable average error which is

the Bayes error, or Bayes error rate. This chapter focuses on the necessary theory to explain Bayes error rate and bounding it using a multilayer perceptron.

2.5 *Bayes Decision Theory*

"Bayes decision theory is a fundamental statistical approach to the problem of pattern classification (3)." A rigorous presentation of Bayes decision theory is presented by Duda and Hart and the reader is encouraged to explore their presentation (3). To give the reader a clear overview of Bayes decision theory, a two-class problem will be considered (3).

Suppose that a fish packing plant wanted to automate its operations of packing sea bass and salmon. The two types of fish will be sorted on a conveyor belt by an optical scanner. After processing the images, the information (or features) that discriminate between a sea bass and salmon will be extracted. Assume that the salmon is lighter in color than the sea bass. Therefore, brightness is used as a feature and the classification of the two types of fish will be based solely on brightness. Further assume that the brightness data points for the sea bass and salmon are as shown in Figure 6.

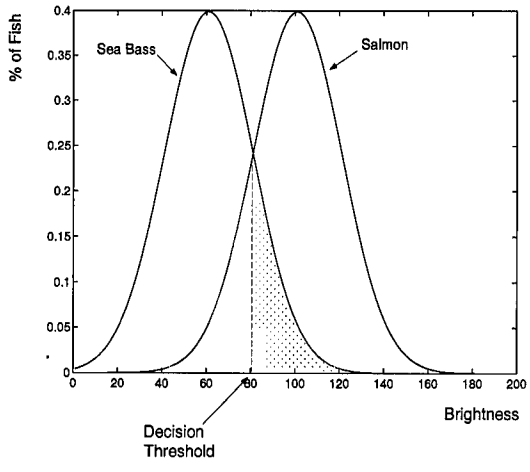

Figure 6. Lightness Distributions of the Sea Bass and Salmon

From Figure 6, it can be seen that most of the salmon are indeed lighter than the sea bass. Also note that there is no way to partition the feature space into two absolutely distinct regions

(one corresponding to a sea bass and the other to a salmon). This plot suggest the following rule for classifying the fish based on the lightness feature: Classify the fish as salmon if its feature vector falls above the decision threshold, and as a sea bass otherwise. From Figure 6, when brightness measurement equals 100, a great percentage of salmon will be classified as salmon, but some sea bass could also be classified as salmon at this brightness level. So, some probability of error exists with every decision. The Bayes error is the shaded area under the curve on either side of the decision threshold. The probability of error is further discussed in Section 2.5.2 and by Duda and Hart (3).

2.5.1 Bayes Rule. To understand Bayes Rule, Table 1 shows the variables used and provides a brief description of each (3).

Table 1. Bayes Rule Variables

Variable	Description
ω_i	state of nature or class (it is a random variable); i = 1,2,...
P(ω_i)	a priori probability of class ω_i
x	a measurement or feature (random variable whose distribution depends on ω_i)
p(x)	probability density function (pdf) for x
p($x \mid \omega_i$)	state-conditional probability density function for x
P($\omega_i \mid x$)	a posteriori probability

The goal is to find P($\omega_i \mid x$) or, in a pattern recognition sense, to make a class decision based on a measurement, x. Bayes Rule provides a way to find P($\omega_i \mid x$) as shown below.

$$\mathrm{P}(\omega_i \mid x) = \frac{\mathrm{p}(x|\omega_i)\mathrm{P}(\omega_i)}{\mathrm{p}(x)}$$

Observation of the a posteriori probabilities provided by Bayes Rule provides the basis for calculating the probability of error associated with choosing a particular state of nature.

2.5.2 Probability of Error. For a two-class problem as presented in Section 2.5, if the a posteriori probability for the salmon was greater than that of the sea bass for a given measurement, we would be inclined to choose the salmon as the true state of nature. With this decision as in all pattern recognition classification, some probability of error is associated with

the choice. For a particular measurement, x, the probability of error associated with choosing the wrong state of nature is illustrated by the following rule:

$$\mathrm{P}(error \mid x) = \begin{cases} \mathrm{P}(\omega_1 \mid x), & \text{if } \omega_2 \text{ is chosen;} \\ \mathrm{P}(\omega_2 \mid x), & \text{if } \omega_1 \text{ is chosen.} \end{cases}$$

Define an optimal classifier as one that minimizes the probability of classification error. In order to achieve this minimum, the classifier must choose ω_1 if $\mathrm{P}(\omega_1 \mid x)$ is greater than $\mathrm{P}(\omega_2 \mid x)$. Since all values of x must be considered, the average probability of error can be computed from the following math.

$$\mathrm{P}(error) = \int_{-\infty}^{+\infty} \mathrm{P}(error \mid x)\mathrm{p}(x)dx$$

Note that if for every value of x, $\mathrm{P}(error \mid x)$ is as small as possible, the integral will be as small as possible and the average probability of error will be minimized. From this analysis, Bayes decision rule is born for minimizing the probability of error.

Decide ω_1 if $\mathrm{P}(\omega_1 \mid x) > \mathrm{P}(\omega_2 \mid x)$: otherwise decide ω_2.

Bayes decision rule can be rewritten as:

Decide ω_1 if $\mathrm{p}(x \mid \omega_1)\mathrm{P}(\omega_1) > \mathrm{p}(x \mid \omega_2)\mathrm{P}(\omega_2)$: otherwise decide ω_2

(if $\mathrm{p}(x)$ is treated as a scale factor and just eliminated from the math).

Since the Bayes error rate minimizes the probability of error, a classifier that approaches or matches the Bayes error rate is highly desirable to achieve maximum classification accuracy. In most real-world problems, neither the a priori probability nor conditional pdf for each class is known, so it is impossible to analytically determine the Bayes error rate. There are several techniques to place a bound around the Bayes error for a given set of data. The Bayes error bound can be used to gauge how well a particular classifier performs. If a classifier's error rate falls within the computed Bayes error bound, then the classifier should be considered to have performed well because achieving the Bayes error rate is the best any classifier can be expected to achieve on average.

2.6 *Bounding the Bayes Error Rate*

There are several techniques to bound the Bayes error rate. The method considered here uses two different types of data manipulation, namely the resubstitution and leave-one-out methods. In most pattern recognition problems, only a finite set of data exists and that finite set must be used to not only design a classifier, but also test the classifier. By using both the resubstitution and leave-one-out methods a bound on the Bayes error rate can be found.

2.6.1 Resubstitution Method. In the resubstitution method, the entire finite set of data is used to design the classifier. If N is the complete set of data, N_D the design set of data, and N_T the test set of data, then the following math represents how the data are utilized in the resubstitution method.

$$N = N_D = N_T$$

The estimate of the probability of error is found by finding the proportion of samples that are misclassified in the test set.

2.6.2 Leave-One-Out Method. In the leave-one-out method, every sample of the finite set of data is used to design a classifier except one which is held out to test the classifier. This procedure of leaving one sample of the data set out for testing is repeated until all samples of the data set have been used for testing. Using the notation in Section 2.6.1, the following equations represent how the data is utilized in the leave-one-out method.

$$N_D = N - 1 \ \ \& \ \ N_T = 1$$

The estimate of the probability of error is found by finding the proportion of samples that are misclassified in all of the test sets. Lachenbruch first published the leave-one-out method in 1967 (9).

2.6.3 Resubstitution and Leave-One-Out Bounds. Each method returns an estimate of the probability of error as a function of the number of neighbors in a k-nearest neighbor

density estimator, the size of the window of a parzen window density estimator, or the number of hidden nodes in a multilayer perceptron (12) (13). For both the resubstitution and leave-one-out methods, an estimate of the probability of error is found over a range of parameters to produce a curve for each method as seen in Figure 7.

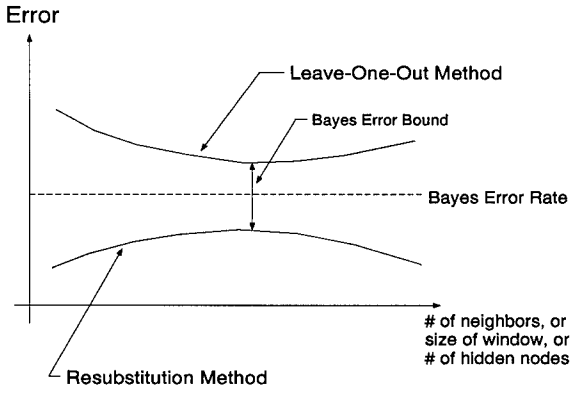

Figure 7. Bayes Error Bound

Using the resubstitution and leave-one-out methods on the same set of data provides upper and lower bounds on the Bayes error rate (6) as illustrated in Figure 7. The bound is actually found by first finding the minimum of the leave-one-out curve and then finding the corresponding value of the resubstitution curve. Since generating even one curve for either method is a random process, a curve for each method should be found several times and then the average curve for each method should ultimately be used. The resubstitution method returns an estimate of the error rate which is generally optimistic. It is considered optimistic because the error rate is usually lower than the Bayes error rate, but cannot be achieved by a classifier when it is presented with new samples outside of the finite set of data used (6). On the other hand, the leave-one-out method returns an estimate of the error rate which could be considered pessimistic because the error rate is usually higher than the Bayes error rate.

2.7 *Introduction to Feature Selection*

When analyzing the classification results of an MLP, the following question may arise, "which features presented as input were most important in determining the outcome?" For example, in the business world neural networks have been used to classify individuals as good or bad loan risks based on hundreds of factors (or features) such as age, income, debts, etc (18). The neural network may perform superbly in classifying these loan applicants but the financial institution is required by law to inform all those denied a loan the reason for denial. For this purpose the financial institution must know which input features were most important in classifying the loan applicant as a bad loan risk. Feature selection techniques can be applied to this problem in order to determine the most important features that contributed to classifying the individual as a bad loan risk.

Feature selection is the process by which a large set of candidate features is reduced to a smaller set while the techniques used in feature selection are aimed at partitioning the feature set into the important or *salient* features and the unimportant features (22). Although there are several approaches to neural network feature selection, all techniques fall into three general categories (22). The first class of techniques involves a search for relevant feature subsets, the second class uses saliency metrics to rank individual features, and the third class is concerned with screening irrelevant features. An excellent presentation of these techniques is presented by Steppe and the reader is encouraged to explore her presentation (21).

In this thesis, the first and second classes of techniques will be explored and used. Before any analysis of the data is performed, Fisher's discriminant is used to initially screen the data (14). Section 2.7.1 will present Fisher's discriminant. Section 2.7.2 will present the first class of feature selection techniques, forward sequential selection, and Section 2.7.3 will present the second class of techniques, saliency metrics.

2.7.1 Fisher's Discriminant. As stated in the last section, Fisher's discriminant is initially employed to screen the data. For each of the three two-class problems analyzed in this thesis, Fisher's discriminant is used to compare the data of one class versus the other. Fisher's discriminant, f, is defined in Equation 1 (14).

$$f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{1}$$

Note that μ is the mean and σ^2 is the variance.

2.7.2 Forward Sequential Selection. Steppe labels this technique as forward sequential selection (21) while it is also referred to as an add-on procedure (14). Each feature under consideration is analyzed individually and the feature which produces the best classification accuracy is used as the nucleus for the next set. Then, all pairs of features comprising the nucleus and one other feature are analyzed. The feature whose addition to the nucleus results in the best classification accuracy is incorporated into the nucleus. This process is repeated each time adding the one feature whose addition results in the best classification accuracy until all features have been considered or until the desired level of performance has been achieved.

2.7.3 Saliency Metrics. Although saliency metrics have been proposed by Ruck, Priddy, and Tarr (19) (15) (23), Steppe demonstrated the equivalence of these metrics (21). In this thesis, only the Ruck saliency metric will be presented and used (19). The Ruck saliency metric derives an expression for the derivative of an output with respect to a given input and then uses this expression to measure the sensitivity of an MLP to each input feature. The derivation using the notation presented in Section 2.3 for the MLP is shown in Appendix C and only the highlights are shown here. Note again that superscripts always represent a layer index and not a quantity raised to a power.

The definition of activation is the weighted sum of the input values plus the threshold as shown below for the kth node of the output layer, a_k^o.

$$a_k^o = \sum_{j=1}^{J+1} W_{jk}^2 x_j^2$$

(Note: The output of the hidden nodes is defined as x_j^2 and W_{jk}^2 is the weight from the jth node on the hidden layer to the kth output node.)

The output of the MLP is y_k and using a sigmoidal transformation it is shown below.

$$y_k = f_H(a_k^o) = \frac{1}{1+e^{-a_k^o}}$$

To compute the Ruck saliency metric, the derivative of the output with respect to the input must be found as shown below.

$$\frac{\partial y_k}{\partial x_i} = \frac{\partial f_H(a_k^o)}{\partial x_i} = y_k(1-y_k)\frac{\partial a_k^o}{\partial x_i}$$

The resulting saliency metric, Λ_i, measures the usefulness of each input feature for determination of the correct output class.

$$\Lambda_i = \sum_{k=1}^{K} \mid \frac{\partial y_k}{\partial x_i} \mid = \sum_{k=1}^{K} \mid \delta_k^o \sum_{j=1}^{J+1} W_{jk}^2 \delta_j^1 W_{ij}^1 \mid$$

2.8 Summary

This review has detailed the necessary theory to analyze the Fisher-344 rat breath data with neural networks. The next chapter describes how the data were processed and presents the results of each individual method as each method is presented.

III. Methods & Results

3.1 Introduction

In this chapter, the methods used to analyze the Fisher-344 rat breath data are outlined. First, the initial analysis and manipulation of the data are discussed. An overview of the methods employed in this thesis is shown in figure 8. As each individual method is presented throughout this chapter, the results obtained using that method will be shown directly after.

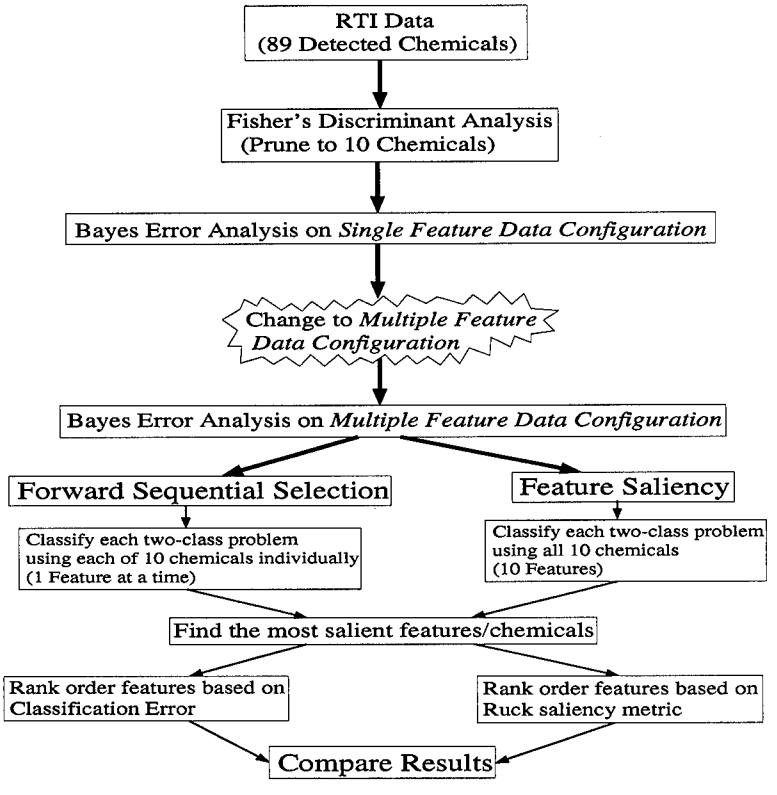

Figure 8. Methods Overview

3.2 *Data Manipulation*

3.2.1 Correlation Analysis. The data provided by RTI were obtained through gas chromatography/mass spectrometry (GC/MS) (16). The actual chromatographs of the rat breath were not analyzed in this research. Two measurements were provided for each dose as shown in the sample data entry line in Table 2.

Table 2. Sample Data Entry Line

Observation	*Compound*	*Peak Number*	*Drug*	*Level*	*Rat Wt.*	*M1*	*M2*
428	1,2-Dichloroethane	20.4	CCl4, 96-98h	Hi 1	0.728	2.66321E-05	3.05602E-07

The observation column provides the tracking number of the specific measurements, M1 and M2. For an observation, the varying input factors are drug, level, and rat weight and the varying output factors are compound, peak number, M1, and M2. The compound column identifies the chemical compound detected using GC/MS and denotes a specific pattern classification feature for the purposes of this research. The peak number identifies the specific peak associated with the chemical compound on the gas chromatograph. The peak number is not used in this research since the actual chromatographs were not analyzed. The drug column identifies one of three doses: 1) carbon tetrachloride, 2) vehicle control (VC), or 3) no dose which are denoted as dosage conditions in this thesis. VC is simply a saline solution. Also, the drug column denotes the time of the measurements with respect to the injection time which was ignored in this research. The level column refers to the level of the carbon tetrachloride dose. A carbon tetrachloride dose has three dosage levels 1) low 2) medium and 3) high, but only the high dose level was used in this research. The rat weight column provided the mass of the rat used for each specific observation but was not pertinent for this research.

In a pattern recognition sense, each dosage condition represented a separate class. For instance, using this data a two-class problem can be created by assigning no dose as one class and assigning another dosage condition, such as a carbon tetrachloride dose, as the other class. M1 and M2 are naturally chosen as features since they are the only relevant measurements provided. M1 is the measurement of a μ mole of compound per 100 grams of rat mass

per minute and M2 is a μ mole of compound per μ mole of carbon dioxide produced. The relationship of M1 to M2 was tested by computing the sample correlation coefficient as defined below (2).

Let S_{xy} be the sample covariance. Then, given n pairs of observations (x_1, y_1), (x_2, y_2), ... , (x_n, y_n),

$$S_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})$$

$\overline{x}$ and $\overline{y}$ are the sample means. Let r be defined as the sample correlation coefficient.

$$\text{Then,} \quad r = \frac{S_{xy}}{\sqrt{\sum(x_i-\overline{x})^2}\sqrt{\sum(y_i-\overline{y})^2}}$$

A reasonable rule of thumb to say that two variables are correlated is if $0.8 \leq r \leq 1.0$ denotes a strong correlation (2). A scatter plot of the data plotting one variable against another can also provide insight into the correlation as shown in Figure 9 (2). The computed sample correlation coefficient for M1 and M2 was r = 0.9932. A scatter plot of the data is shown in Figure 10. It is obvious that M1 and M2 are strongly correlated and, therefore, only one of the variables, M1, need be used as a feature since using both would be redundant.

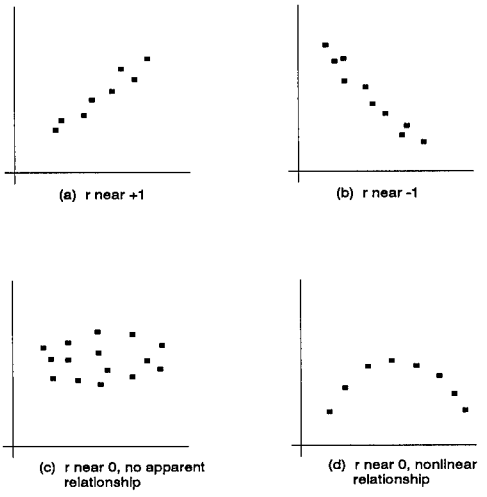

Figure 9. Scatter Plots for Different Values of r

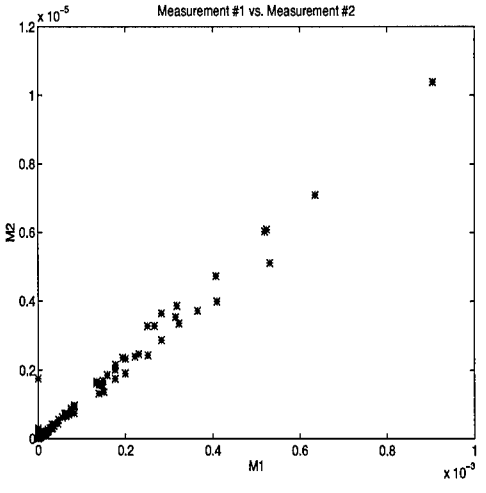

Figure 10. Scatter Plots for M1 and M2, $r = 0.9932$

3.2.2 Data Configuration. In the rat breath data, over 80 different chemical compounds were detected over the range of dosage conditions. Considering if only M1 is used, the question arises as to how to configure the data in order to be able to tell which chemical compounds provide the best discrimination between any two of the three dosage conditions. Two possible ways to configure the data are 1) have a single entry feature vector irrespective of the detected chemical compound as shown in Equation 2 or 2) treat M1 and each corresponding chemical compound as a separate feature as shown in Equation 3. Remember from Section 3.2.1 that for a two-class problem, class 0 could be assigned to all measurements from observations of no doses and class 1 could be assigned to all measurements from observations of carbon tetrachloride doses.

$$X1 = \begin{bmatrix} \text{Class} & \text{Feature} \\ 0 & \text{M1} \\ 0 & \text{M1} \\ \vdots & \vdots \\ 1 & \text{M1} \\ 1 & \text{M1} \end{bmatrix} \tag{2}$$

$$X2 = \begin{bmatrix} \text{Class} & \text{Feature 1} & \text{Feature 2} & \cdots & \text{Feature N} \\ 0 & \text{M1(Acetone)} & \text{M1(Benzene)} & \cdots & \text{M1(Nth Compound)} \\ 0 & \text{M1(Acetone)} & \text{M1(Benzene)} & \cdots & \text{M1(Nth Compound)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \text{M1(Acetone)} & \text{M1(Benzene)} & \cdots & \text{M1(Nth Compound)} \\ 1 & \text{M1(Acetone)} & \text{M1(Benzene)} & \cdots & \text{M1(Nth Compound)} \end{bmatrix} \tag{3}$$

Hereafter, the data configuration in Equation 2 is labeled the single feature configuration and the data configuration in Equation 3 is labeled the multiple feature configuration. Note that for the multiple feature configuration the number of chemical compounds used as features can

be varied from 1 to N where N is the total number of chemical compounds detected. Further note in the multiple feature configuration that each row represents multiple M1 measurements incorporating several observations. The effectiveness of each configuration is analyzed in Section 3.4.

3.3 Fisher's Discriminant

Although 89 chemical compounds were detected in the rat breath, M1 of just 10 compounds were considered in the two different data configurations for three two-class classification problems 1) a VC dose versus a carbon tetrachloride dose 2) a no dose versus a carbon tetrachloride dose and 3) a no dose versus a VC dose. In an effort to parse the original 89 chemicals down to 10 chemicals for each of the 3 dosage conditions, Fisher's discriminant was employed (14).

For example, in classification Case 1 Fisher's discriminant is calculated for each of the 89 compounds comparing only carbon tetrachloride dose data points to no dose data points. Then, the ten compounds with the highest f are considered for the parsed feature set. This process is then repeated for the other two classification cases.

Fisher's discriminant was computed for all 89 chemical compounds for each of the three cases of classification. The results of computing Fisher's discriminant for each classification case are shown in Tables 3, 4, and 5. Only the top ten chemicals and chloroacetone are shown. Chloroacetone was added in the analysis in the first two classification cases because it was specifically singled out in the RTI study (16) but did not make the top ten list. A key is provided in Appendix D which shows the full name of the chemical compounds abbreviated throughout this section. Chemicals are ranked in descending order of Fisher's discriminant values in Tables 3, 4, and 5.

Table 3. Fisher's Discriminant for VC vs. Carbon Tetrachloride

Compound	chlorobenz	unkflo	tetra	2butanal	c7h12	sat2	2pent	npent	1butanol	2hex	chloroace
f	5.2847	3.7934	2.9975	2.7402	2.2843	1.8227	1.7428	1.5299	1.4946	1.4913	1.2032

Table 4. Fisher's Discriminant for No Dose vs. Carbon Tetrachloride

Compound	chlorobenz	unkflo	12di	2butanal	tetra	2pent	2hex	npent	methmeth	1butanol	chloroace
f	7.3392	5.3348	4.1911	3.9360	2.9261	2.7052	1.8232	1.5827	1.5706	1.4747	1.3810

Table 5. Fisher's Discriminant for No Dose vs. VC

Compound	pchloro	ethace	odi	2hex	chloroace	phenol	2methprop	2methfur	benzonitrile	2octbenz
f	3.9378	2.9223	2.8823	1.6534	1.5715	1.5329	1.4299	1.3303	1.1898	1.0508

Note that nine of the eleven compounds in the VC versus carbon tetrachloride classification case are found in the no dose versus carbon tetrachloride case. This result is not surprising given the fact that the only difference between a VC and no dose is that a VC consists of a saline solution. From the top ten Fisher discriminant chemical compounds, the Bayes error analysis and feature selection techniques can now be employed for all three classification cases (refer to Figure 8).

3.4 Bayes Error Estimation

The Bayes error estimation using Parzen/k-nn techniques was employed using MATLAB® code (12). The reader is encouraged to explore Martin's programs to gain insight into this methodology. The Bayes error estimation using MLP techniques was accomplished using the software LNKnet (10). When the data manipulation technique of resubstitution was employed, 500 epochs were used to train the MLP. With the leave-one-out method, 50 training epochs were employed in each experiment. The number of hidden nodes was varied for both data manipulation methods as discussed in Chapter II.

3.4.1 Bayes Error Estimation of Single Feature Data Configuration. The data were initially put into the single feature configuration and the Bayes error was estimated. Figure 11 shows the computed bounds using a Parzen window density estimator. Analysis of Figure 11 shows that the estimated Bayes error bound is [22,60]. After realizing that the best error that could be achieved was between 22 and 60 percent, the single feature data configuration was abandoned and the multiple feature data configuration was adopted. Sections 3.4.2.1 to 3.4.2.3 will show that the multiple feature configuration can achieve much lower error.

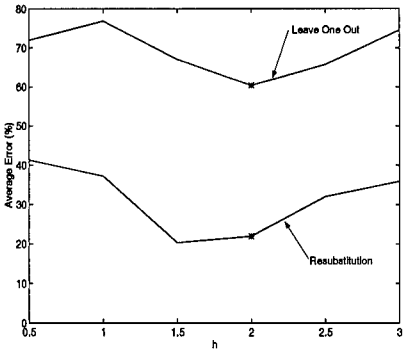

Figure 11. Bayes Error Bounds for VC vs. Carbon Tetrachloride (Single Feature)

3.4.2 Bayes Error Estimation of Multiple Feature Data Configuration. For each of the three dosage conditions, the Bayes error is bounded using a Parzen window density estimator and an MLP.

3.4.2.1 VC vs. Carbon Tetrachloride. With the data in the multiple feature configuration, an estimate of the Bayes error was computed. Figure 12 shows the computed bounds using a Parzen window density estimator. Figure 13 shows the computed bounds using an MLP. Analysis of Figure 12 and 13 shows that the estimated Bayes error bound is (0,6). Note the consistencies of the computed bounds between the Parzen window density estimator and MLP.

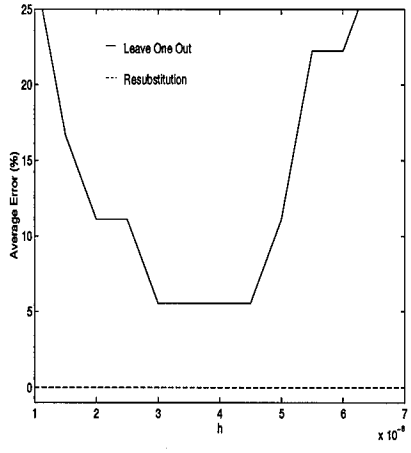

Figure 12. Bayes Error Bounds for VC vs. Carbon Tetrachloride (Parzen)

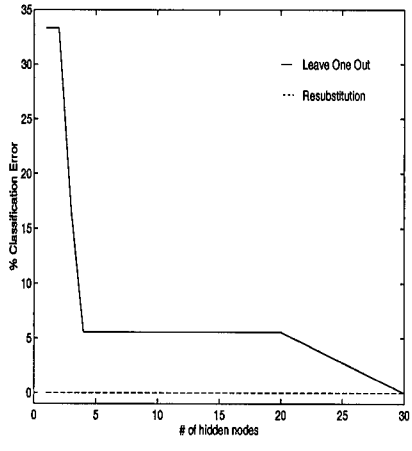

Figure 13. Bayes Error Bounds for VC vs. Carbon Tetrachloride (MLP)

3.4.2.2 No Dose vs. Carbon Tetrachloride. With the data in the multiple feature configuration, an estimate of the Bayes error was computed. Figure 14 shows the computed bounds using a Parzen window density estimator. Figure 15 shows the computed bounds using an MLP. Analysis of Figure 14 shows that the estimated Bayes error bound is (0,3.5) and Figure 15 illustrates a consistent bound shown on the last step before the MLP memorizes.

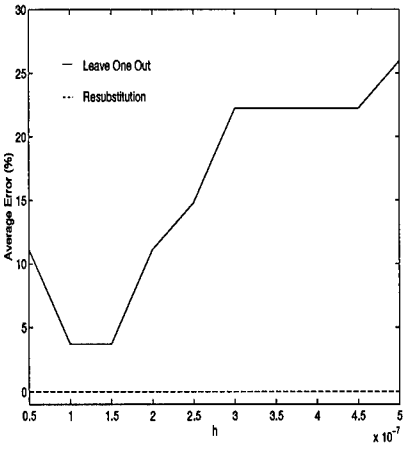

Figure 14. Bayes Error Bounds for No Dose vs. Carbon Tetrachloride (Parzen)

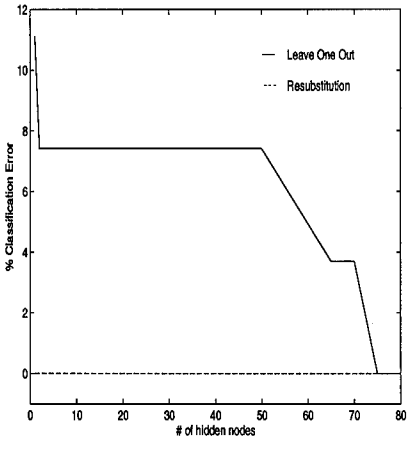

Figure 15. Bayes Error Bounds for No Dose vs. Carbon Tetrachloride (MLP)

3.4.2.3 No Dose vs. VC. With the data in the multiple feature configuration, an estimate of the Bayes error was computed. Figure 16 shows the computed bounds using a Parzen window density estimator. Figure 17 shows the computed bounds using an MLP. Analysis of Figure 16 and 17 consistently shows that the estimated Bayes error bound is (0,14).

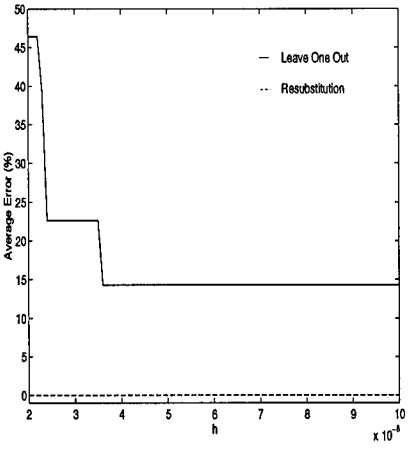

Figure 16. Bayes Error Bounds for No Dose vs. VC (Parzen)

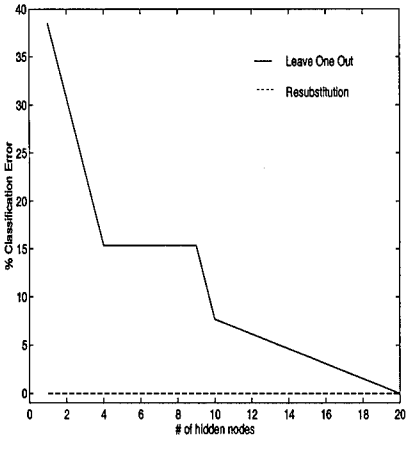

Figure 17. Bayes Error Bounds for No Dose vs. VC (MLP)

3.5 *Feature Selection*

Two methods are employed to find which chemical compounds found in the rat breath best contribute to the discrimination between dosage conditions. The first method is a forward sequential selection technique (14) (21) and the second method involves computing a saliency metric (19).

3.5.1 Forward Sequential Selection. For each dosage condition, the forward sequential technique is employed by classifying each two-class problem using each of the 10 chemical compounds individually. Each compound can be viewed as an individual feature as is done in the multiple feature configuration. Each feature/compound is classified in the initial step of the forward sequential selection technique. If conditions warrant a second or subsequent steps as explained in Section 2.7.2, then that step is performed.

3.5.1.1 VC vs. Carbon Tetrachloride. Table 6 shows the first step results in the forward sequential selection technique. Analysis of Table 6 shows that 2hex achieved the lowest classification error and is, therefore, used as the first feature of the nucleus.

Table 6. First Step Classification Error for VC vs. Carbon Tetrachloride

Compound	chlorobenz	unkflo	tetra	2butanal	c7h12	sat2	2pent	npent	1butanol	2hex	chloroace
% Error	6.25	6.67	12.5	7.69	10.0	61.54	7.41	11.11	6.25	0.00	20.0

To illustrate the effectiveness of the 2hex feature, Table 7 shows the second step results in the forward sequential selection technique where the compounds listed are classified with 2hex in the multiple feature configuration.

Table 7. Second Step Classification Error for VC vs. Carbon Tetrachloride

Compound and 2hex	chlorobenz	unkflo	tetra	2butanal	c7h12	sat2	2pent	npent	1butanol	chloroace
% Error	0.00	6.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Analysis of Tables 6 and 7 shows that when 2hex is used in conjunction with all other compounds except unkflo, the classification error is decreased.

3.5.1.2 No Dose vs. Carbon Tetrachloride. Table 8 shows the first step results in the forward sequential selection technique. Analysis of Table 8 shows that 2hex achieved the lowest classification error and is, therefore, used as the first feature of the nucleus.

Table 8. First Step Classification Error for No Dose vs. Carbon Tetrachloride

Compound	chlorobenz	unkflo	12di	2butanal	tetra	2pent	2hex	npent	methmeth	1butanol	chloroace
% Error	8.33	15.38	14.29	14.29	7.69	16.67	0.00	15.38	15.38	10.00	36.36

To illustrate the effectiveness of the 2hex feature, Table 9 shows the second step results in the forward sequential selection technique where the compounds listed are classified with 2hex in the multiple feature configuration.

Table 9. Second Step Classification Error for No Dose vs. Carbon Tetrachloride

Compound	chlorobenz	unkflo	12di	2butanal	tetra	2pent	npent	methmeth	1butanol	chloroace
% Error	8.33	15.38	7.69	0.00	0.00	5.56	0.00	0.00	7.69	0.00

Analysis of Tables 8 and 9 shows that 2hex is used again as the first feature of the nucleus as it was used for the VC versus carbon tetrachloride classification case. Again, when 2hex is used in conjunction with all other compounds except chlorobenz and unkflo, the classification error is decreased.

3.5.1.3 No Dose vs. VC. Table 10 shows the first step results in the forward sequential selection technique. Analysis of Table 10 shows that ethace achieved the lowest classification error and is, therefore, used as the first feature of the nucleus.

Table 10. First Step Classification Error for No Dose vs. VC

Compound	pchloro	ethace	odi	2hex	chloroace	phenol	2methprop	2methfur	benzonitrile	2octbenz
% Error	30.00	11.11	40.00	22.22	60.00	20.00	20.00	57.14	33.33	57.14

Analysis of Table 11 reveals that no feature/compound achieves zero classification error as happened in the first two classification cases. So, a second step in the forward sequential selection technique is necessary to further analyze the features. Table 11 shows the second step results in the forward sequential selection technique where the compounds listed are classified with ethace in the multiple feature configuration.

Table 11. Second Step Classification Error for No Dose vs. VC

Compound	pchloro	odi	2hex	chloroace	phenol	2methprop	2methfur	benzonitrile	2octbenz
% Error	9.09	9.09	10.00	11.11	8.33	10.00	0.00	0.00	0.00

Analysis of Tables 10 and 11 shows that when ethace is used in conjunction with all other compounds, the classification error is decreased.

3.5.2 Saliency Metrics. The Ruck saliency metric must be computed when the data is in the multiple feature data configuration. For each dosage condition, each two-class problem is classified using an MLP. Then, the "usefulness" of each input feature for determination of the correct output class (found by the MLP) is measured by the saliency metric. MATLAB® code was employed and is provided in Appendix E. Once the saliency metric is computed for each feature, they are rank ordered and then compared to the results obtained in the forward sequential selection technique in Section 3.6. Classification results will first be shown and then the saliency of the features which provided those classification results will be presented for each dosage condition.

3.5.2.1 VC vs. Carbon Tetrachloride. With the data in the multiple feature configuration using all of the compounds in the Fisher's discriminant results shown in Section 3.3, the VC versus carbon tetrachloride data was classified using an MLP. The results are shown in Table 12.

Table 12. Confusion Matrix for VC vs. Carbon Tetrachloride

Actual	*Assigned*	
-	VC	CCl4
VC	9	-
CCl4	-	9

Analysis of Table 12 shows that the MLP classified all data points perfectly. That is, all VC data points were classified as VC and all carbon tetrachloride data points were classified as carbon tetrachloride. Table 13 shows the saliency of the features used in the classification shown in Table 12.

Table 13. Feature Saliency for VC vs. Carbon Tetrachloride

Compound	chlorobenz	unkflo	tetra	2butanal	c7h12	sat2	2pent	npent	1butanol	2hex	chloroace
Saliency	0.9032	1.0000	0.8335	0.7747	0.3818	0.5104	0.8070	0.7736	0.8694	0.8882	0.7552

Analysis of Table 13 shows the most salient feature/compound is unkflo while the least salient is c7h12. Therefore, the feature that was most useful in perfectly classifying the dosage condition, VC versus carbon tetrachloride, is unkflo according to the Ruck saliency metric.

3.5.2.2 No Dose vs. Carbon Tetrachloride. With the data in the multiple feature configuration using all of the compounds in the Fisher's discriminant results shown in Section 3.3, the carbon tetrachloride versus no dose data was classified using an MLP. The results are shown in Table 14.

Table 14. Confusion Matrix for No Dose vs. Carbon Tetrachloride

Actual	*Assigned*	
-	No	CCl4
No	18	-
CCl4	-	9

Analysis of Table 14 shows that the MLP classified all data points perfectly. That is, all no dose data points were classified as no dose and all carbon tetrachloride data points were classified as carbon tetrachloride. Table 15 shows the saliency of the features used in the classification shown in Table 14.

Table 15. Feature Saliency for No Dose vs. Carbon Tetrachloride

Compound	chlorobenz	unkflo	12di	2butanal	tetra	2pent	2hex	npent	methmeth	1butanol	chloroace
Saliency	1.0000	0.8858	0.2324	0.7257	0.7009	0.6266	0.8972	0.5792	0.6522	0.9714	0.6701

Analysis of Table 15 shows the most salient feature/compound is chlorobenz while the least salient is 12di. Therefore, the feature that was most useful in perfectly classifying the dosage condition, no dose versus carbon tetrachloride, is chlorobenz according to the Ruck saliency metric.

3.5.2.3 No Dose vs. VC. With the data in the multiple feature configuration using all of the compounds in the Fisher's discriminant results shown in Section 4.3, the no dose versus VC data was classified using an MLP. The results are shown in Table 16.

Table 16. Confusion Matrix for No Dose vs. VC

Actual	*Assigned*	
-	No	VC
No	8	-
VC	-	5

Analysis of Table 16 shows that the MLP classified all data points perfectly. That is, all no dose data points were classified as no dose and all VC data points were classified as VC. Table 17 shows the saliency of the features used in the classification shown in Table 16.

Table 17. Feature Saliency for No Dose vs. VC

Compound	pchloro	ethace	odi	2hex	chloroace	phenol	2methprop	2methfur	benzonitrile	2octbenz
Saliency	0.6465	1.0000	0.6789	0.8008	0.8826	0.5460	0.9665	0.9593	0.9726	0.8084

Analysis of Table 17 shows the most salient feature/compound is ethace while the least salient is phenol. Therefore, the feature that was most useful in perfectly classifying the dosage condition, no dose versus VC, is ethace according to the Ruck saliency metric.

3.6 *Results Comparison*

3.6.1 VC vs. Carbon Tetrachloride. Table 18 compares the results of the tests performed to compute Fisher's discriminant (f), forward sequential selection classification error, and feature saliency.

Table 18. Results Comparison for VC vs. Carbon Tetrachloride

Compound	chlorobenz	unkflo	tetra	2butanal	c7h12	sat2	2pent	npent	1butanol	2hex	chloroace
f Rank	1	2	3	4	5	6	7	8	9	10	14
1st Step Fwd Seq Rank	2	4	9	6	7	11	5	8	2	1	10
Saliency Rank	2	1	5	7	11	10	6	8	4	3	9

Analysis of Table 18 shows that both feature selection techniques, forward sequential selection and feature saliency, have very comparable results for all features while Fisher's discriminant results are not consistent with results of either technique.

3.6.2 No Dose vs. Carbon Tetrachloride. Table 19 compares the results of the tests performed to compute Fisher's discriminant (f), forward sequential selection classification error, and feature saliency.

Table 19. Results Comparison for No Dose vs. Carbon Tetrachloride

Compound	chlorobenz	unkflo	12di	2butanal	tetra	2pent	2hex	npent	methmeth	1butanol	chloroace
f Rank	1	2	3	4	5	6	7	8	9	10	12
1st Step Fwd Seq Rank	3	7	5	5	2	10	1	7	7	4	11
Saliency Rank	1	4	11	5	6	9	3	10	8	2	7

Analysis of Table 19 shows that both feature selection techniques, forward sequential selection and feature saliency, have very comparable results for all features while Fisher's discriminant results are not consistent with results of either technique.

3.6.3 No Dose vs. VC. Table 20 compares the results of the tests performed to compute Fisher's discriminant (f), forward sequential selection classification error, and feature saliency.

Table 20. Results Comparison for No Dose vs. VC

Compound	pchloro	ethace	odi	2hex	chloroace	phenol	2methprop	2methfur	benzonitrile	2octbenz
f Rank	1	2	3	4	5	6	7	8	9	10
1st Step Fwd Seq Rank	5	1	7	4	10	2	2	8	6	8
Saliency Rank	9	1	8	7	5	10	3	4	2	6

Analysis of Table 20 shows that both feature selection techniques, forward sequential selection and feature saliency, have very comparable results for all features while Fisher's discriminant results are not consistent with results of either technique.

Figure 18 shows a scatter plot of M1 for 2hex for the carbon tetrachloride versus VC classification case. Note the wide variance of the class one data points as compared to the variance of the class zero data points. This plot provides an excellent example of how Fisher's discriminant may be relatively small but there is still separability between class one and class zero data points.

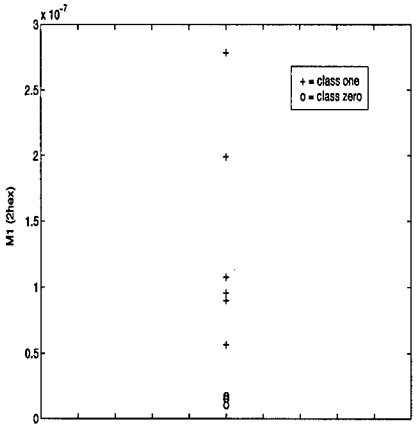

Figure 18. Scatter Plot of the Two-Class Data for 2hex

Figure 19 shows a scatter plot of M1 for 1butanol for the carbon tetrachloride versus VC classification case. Note that the separability of the class zero and class one data points is not as large as in the 2hex data although their Fisher's discriminants are similar (f is 1.4913 and 1.4946 for 2hex and 1butanol, respectively).

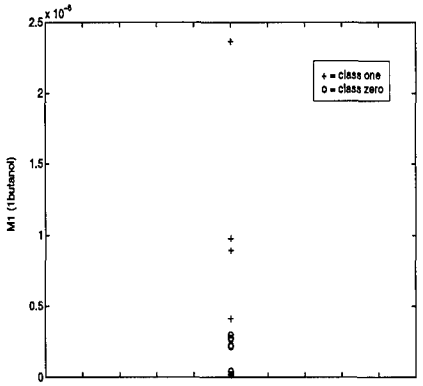

Figure 19. Scatter Plot of the Two-Class Data for 1butanol

Figure 20 shows 2hex in conjunction with 1butanol in the multiple feature configuration. Note the separability between class zero and class one data points. Figure 20 shows how 2hex decreases the classification error by increasing the separability of the class zero and class one data points.

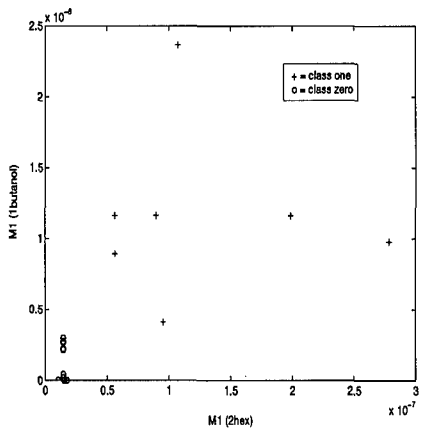

Figure 20. Scatter Plot of the Two-Class Data for 2hex and 1butanol

3.7 Summary

This chapter has presented the methods used to analyze the Fisher-344 rat breath data and the corresponding results. Conclusions about each of the three classification cases and an overall summary of the research will be provided in the next chapter.

IV. Conclusions

4.1 Introduction

In this chapter, conclusions will be drawn from the results presented in Chapter III. The results will be summarized for each dosage condition, recommendations for follow-on research are provided, and an overall summary of the research performed in this thesis is provided.

4.2 Discussion of Results

4.2.1 Overall Results. In this section, results for each of the three classification cases will be discussed. For each classification case, the data were in the multiple feature configuration since the single feature configuration could not achieve desired classification results. Using the techniques presented in this thesis, this research was very successful in showing that the marker chemical compounds for each of the 3 dosage conditions could be found from the rat breath data.

4.2.2 Carbon Tetrachloride vs. VC. The Bayes error bounding results show that the Bayes error can be estimated to be between 0 and 6 percent. The classification error was shown to be 0 percent for this data set. The feature selection results showed that Fisher's discriminant provides an initial analysis to eliminate chemical compounds with very poor separability. Although Fisher's discriminant provides an analysis starting point, it was shown that it cannot be relied upon to provide analogous separability. For instance, 2hex had a Fisher's discriminant rank of ten, but achieved 0 percent classification error in the first step of the forward sequential selection. The feature saliency results validated the forward sequential selection results for all but two chemical compounds (tetra and c7h12). The top three chemical compounds which provide the best discrimination between VC and a carbon tetrachloride dose are 2hex, chlorobenz, and unkflo.

4.2.3 Carbon Tetrachloride vs. No Dose. The Bayes error bounding results show that the Bayes error can be estimated to be between 0 and 3.5 percent. The classification error was shown to be 0 percent for this data set. The feature saliency results validated the forward sequential selection results for all but three chemical compounds (tetra, 12di, and chloroace). The top three chemical compounds which provide the best discrimination between VC and a carbon tetrachloride dose are 2hex, chlorobenz, and 1butanol.

4.2.4 No Dose vs. VC. The Bayes error bounding results show that the Bayes error can be estimated to be between 0 and 14 percent. The classification error was shown to be 0 percent for this data set. The feature saliency results validated the forward sequential selection results for all but four chemical compounds (phenol, pchloro, 2methfur, and chloroace). The top three chemical compounds which provide the best discrimination between VC and a carbon tetrachloride dose are ethace, 2methprop, and benzonitrile.

4.3 Recommendations for Follow-on Research

Several techniques may be employed in follow-on research of this type of data. For instance, time dependency can be factored into the analysis. The time after injection of a carbon tetrachloride dose or VC can be studied because the chromatograms of the rat breath will vary as the injected chemicals dissipate. Secondly, these results could be validated by analyzing the actual chromatograms of the rat breath using principal components analysis (PCA). Performing PCA and then employing an MLP to classify the data will verify any results obtained using the techniques of this research. Lastly, dosage levels of carbon tetrachloride could be analyzed to determine if there is a threshold below which no carbon tetrachloride can be effectively classified versus a VC or no dose.

4.4 Overall Summary of Research

This research was very successful in demonstrating that neural networks can be effectively used to analyze chromatographic data. The complexity of each classification case was estimated by bounding the Bayes error. From the estimation of the Bayes error, a mini-

mum performance level was expected and achieved by the MLP during classification of each two-class problem. The classification results are very important because they demonstrate a no-exposure versus exposure condition can be detected as was seen for the VC versus carbon tetrachloride and no dose versus carbon tetrachloride cases. An interesting result was seen in the classification results of the no dose versus VC case because the neural network was even able to distinguish between two different no-exposure conditions (no dose and VC). The feature selection results show that neural networks can not only be used to classify between exposure conditions but also to demonstrate which chemical compounds provided the best discrimination between each of the dosage conditions. In summary, this research can be deemed highly successful because all research objectives have been met and the techniques presented in this thesis have been demonstrated to be certainly effective in analyzing complex chromatographic data.

Appendix A. Learning Law Derivations

A.1 Introduction

Four combinations of transformations are considered: Sigmoid-Sigmoid, Sigmoid-Linear, Tanh-Tanh, and Tanh-Linear.

A.1.1 Case I: Sigmoid-Sigmoid.

For the output layer,

$$W_{j_0k_0}^{2+} = W_{j_0k_0}^{2-} - \eta \frac{\partial E}{\partial W}$$

Now, just analyzing the partial derivative term in the expression above yields the following.

$$\frac{\partial E}{\partial W_{j_0k_0}^2} = \frac{\partial}{\partial W_{j_0k_0}^2}\{\frac{1}{2}\sum_{k=1}^{K}(d_k - y_k)^2\}$$

$$\frac{\partial E}{\partial W_{j_0k_0}^2} = \frac{1}{2}\{(d_1 - y_1)^2 + \cdots + (d_{k_0}^2 - y_{k_0})^2 + \cdots + (d_k - y_k)^2\}$$

$$\frac{\partial E}{\partial W_{j_0k_0}^2} = \frac{\partial}{\partial W_{j_0k_0}^2}\{\frac{1}{2}\sum_{k=1}^{K}(d_k - y_k)^2\} = (d_{k_0} - y_{k_0})(-1)\frac{\partial y_{k_0}}{\partial W_{j_0k_0}^2}$$

$$\frac{\partial E}{\partial W_{j_0k_0}^2} = -(d_{k_0} - y_{k_0})\frac{\partial}{\partial W_{j_0k_0}^2}(1 + e^{-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2})^{-1}$$

$$\frac{\partial E}{\partial W_{j_0k_0}^2} = -(d_{k_0} - y_{k_0})(-1)(1+e^{-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2})^{-2}(e^{-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2})\frac{\partial}{\partial W_{j_0k_0}^2}(-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2)$$

$$\frac{\partial E}{\partial W_{j_0k_0}^2} = -(d_{k_0} - y_{k_0})\frac{e^{-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2}}{(1 + e^{-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2})^2}(X_{j_0}^2)$$

$$\frac{\partial E}{\partial W_{j_0k_0}^2} = -(d_{k_0} - y_{k_0})(y_{k_0})(1 - y_{k_0})(X_{j_0}^2)$$

Therefore:

$$W_{j_0k_0}^{2+} = W_{j_0k_0}^{2-} + \eta(d_{k_0} - y_{k_0})(y_{k_0})(1 - y_{k_0})(X_{j_0}^2)$$

Now, the hidden layer weights must be updated.

$$W_{i_0j_0}^{1+} = W_{i_0j_0}^{1-} - \eta \frac{\partial E}{\partial W_{i_0j_0}^1}$$

The partial derivative term of the above expression will be analyzed as before.

$$\frac{\partial E}{\partial W_{i_0j_0}^1} = \frac{\partial}{\partial W_{i_0j_0}^1}\{\frac{1}{2}\sum_{k=1}^{K}(d_k - y_k)^2\} = -\sum_{k=1}^{K}(d_k - y_k)\frac{\partial y_k}{\partial W_{i_0j_0}^1}$$

$$\frac{\partial E}{\partial W_{i_0j_0}^1} = -\sum_{k=1}^{K}(d_k - y_k)\frac{\partial}{\partial W_{i_0j_0}^1}(1 + e^{-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2})^{-1}$$

$$\frac{\partial E}{\partial W_{i_0j_0}^1} = -\sum_{k=1}^{K}(d_k - y_k)(y_k)(1 - y_k)\frac{\partial}{\partial W_{i_0j_0}^1}(-\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2)$$

$$\frac{\partial E}{\partial W_{i_0j_0}^1} = -\sum_{k=1}^{K}(d_k - y_k)(y_k)(1 - y_k)(-W_{j_0k}^2)\frac{\partial}{\partial W_{i_0j_0}^1}(X_{j_0}^2)$$

$$\frac{\partial E}{\partial W_{i_0j_0}^1} = -\sum_{k=1}^{K}(d_k - y_k)(y_k)(1 - y_k)(-W_{j_0k}^2)(X_{j_0}^2)(1 - X_{j_0}^2)(-X_{i_0}^1)$$

Therefore:

$$W_{i_0j_0}^{1+} = W_{i_0j_0}^{1-} + \eta\sum_{k=1}^{K}(d_k - y_k)(y_k)(1 - y_k)(W_{j_0k}^2)(X_{j_0}^2)(1 - X_{j_0}^2)(X_{i_0}^1)$$

A.1.2 Case II: Sigmoid-Linear. For the output layer,

$$W_{j_0k_0}^{2+} = W_{j_0k_0}^{2-} - \eta \frac{\partial E}{\partial W_{j_0k_0}^2}$$

Now, just analyzing the partial derivative term in the expression above yields the following.

$$\frac{\partial E}{\partial W_{j_0k_0}^2} = \frac{\partial}{\partial W_{j_0k_0}^2}\frac{1}{2}\sum_{k=1}^{K}(d_k - y_k)^2 = (d_{k_0} - y_{k_0})(-1)\frac{\partial y_{k_0}}{\partial W_{j_0k_0}^2}$$

$$\frac{\partial E}{\partial W_{j_0k_0}^2} = -(d_{k_0} - y_{k_0})\frac{\partial}{\partial W_{j_0k_0}^2}\sum_{j=1}^{J+1} w_{jk_0}^2 x_j^2$$

$$\frac{\partial E}{\partial W^2_{j_0k_0}} = -(d_{k_0} - y_{k_0})(X^2_{j_0})$$

Therefore:

$$W^{2+}_{j_0k_0} = W^{2-}_{j_0k_0} + \eta(d_{k_0} - y_{k_0})(X^2_{j_0})$$

The hidden layer weights must be updated as shown below.

$$W^{1+}_{i_0j_0} = W^{1-}_{i_0j_0} - \eta\frac{\partial E}{\partial W^2_{i_0j_0}}$$

The partial derivative term of the above expression will be analyzed as before.

$$\frac{\partial E}{\partial W^1_{i_0j_0}} = \frac{\partial}{\partial W^1_{i_0j_0}}\{\frac{1}{2}\sum_{k=1}^{K}(d_k - y_k)^2\} = -\sum_{k=1}^{K}(d_k - y_k)\frac{\partial y_k}{\partial W^1_{i_0j_0}}$$

$$\frac{\partial E}{\partial W^1_{i_0j_0}} = -\sum_{k=1}^{K}(d_k - y_k)\frac{\partial}{\partial W^1_{i_0j_0}}\{\sum_{j=1}^{J+1} w^2_{jk_0}x^2_j\}$$

$$\frac{\partial E}{\partial W^1_{i_0j_0}} = -\sum_{k=1}^{K}(d_k - y_k)(W^2_{j_0k})\frac{\partial}{\partial W^1_{i_0j_0}}(X^2_{j_0})$$

$$\frac{\partial E}{\partial W^1_{i_0j_0}} = -\sum_{k=1}^{K}(d_k - y_k)(W^2_{j_0k})(X^2_{j_0})(1 - X^2_{j_0})(X^1_{i_0})$$

Therefore:

$$W^{1+}_{i_0j_0} = W^{1-}_{i_0j_0} + \eta\sum_{k=1}^{K}(d_k - y_k)(W^2_{j_0k})(X^2_{j_0})(1 - X^2_{j_0})(X^1_{i_0})$$

A.1.3 Case III: Tanh-Tanh. For, the output layer,

$$W^{2+}_{j_0k_0} = W^{2-}_{j_0k_0} - \eta\frac{\partial E}{\partial W^2_{j_0k_0}}$$

Now, Just analyzing the partial derivative term in the expression above yield the following.

$$\frac{\partial E}{\partial W^2_{j_0k_0}} = \frac{\partial}{\partial W^2_{j_0k_0}}\{\frac{1}{2}\sum_{k=1}^{K}(d_k - y_k)^2\} = (d_{k_0} - y_{k_0})(-1)\frac{\partial y_{k_0}}{\partial W^2_{j_0k_0}}$$

$$\frac{\partial E}{\partial W^2_{j_0k_0}} = -(d_{k_0} - y_{k_0})\frac{\partial}{\partial W^2_{j_0k_0}}\{\tanh\sum_{j=1}^{J+1} w^2_{jk_0}x^2_j\}$$

$$\frac{\partial E}{\partial W^2_{j_0k_0}} = -(d_{k_0} - y_{k_0})(\cosh\sum_{j=1}^{J+1} w^2_{jk_0}x^2_j)^{-2}\frac{\partial}{\partial W^2_{j_0k_0}}\{\sum_{j=1}^{J+1} w^2_{jk_0}x^2_j\}$$

$$\frac{\partial E}{\partial W^2_{j_0k_0}} = -(d_{k_0} - y_{k_0})(\cosh\sum_{j=1}^{J+1} w^2_{jk_0}x^2_j)^{-2}(X^2_{j_0})$$

$$W^{2+}_{j_0k_0} = W^{2-}_{j_0k_0} + \eta(d_{k_0} - y_{k_0})(\cosh\sum_{j=1}^{J+1} w^2_{jk_0}x^2_j)^{-2}(X^2_{j_0})$$

Therefore:

$$W^{2+}_{j_0k_0} = W^{2-}_{j_0k_0} + \eta(d_{k_0} - y_{k_0})(1 - (y_{k_0})^2)(X^2_{j_0})$$

The hidden layer weights must be updated as shown below.

$$W^{1+}_{i_0j_0} = W^{1-}_{i_0j_0} - \eta\frac{\partial E}{\partial W^2_{i_0j_0}}$$

The partial derivative term of the above expression will be analyzed as before.

$$\frac{\partial E}{\partial W^1_{i_0j_0}} = \frac{\partial}{\partial W^1_{i_0j_0}}\{\frac{1}{2}\sum_{k=1}^{K}(d_k - y_k)^2\} = -\sum_{k=1}^{K}(d_k - y_k)\frac{\partial y_k}{\partial W^1_{i_0j_0}}$$

$$\frac{\partial E}{\partial W^1_{i_0j_0}} = -\sum_{k=1}^{K}(d_k - y_k)(\cosh\sum_{j=1}^{J+1} w^2_{jk_0}x^2_j)^{-2}\frac{\partial}{\partial W^1_{i_0j_0}}\sum_{j=1}^{J+1} w^2_{jk_0}x^2_j$$

$$\frac{\partial E}{\partial W^1_{i_0j_0}} = -\sum_{k=1}^{K}(d_k - y_k)(\cosh\sum_{j=1}^{J+1} w^2_{jk_0}x^2_j)^{-2}\frac{\partial}{\partial W^1_{i_0j_0}}\sum_{j=1}^{J+1} w^2_{jk_0}x^2_j$$

$$\frac{\partial E}{\partial W^1_{i_0j_0}} = -\sum_{k=1}^{K}(d_k - y_k)(\cosh\sum_{j=1}^{J+1} w^2_{jk_0}x^2_j)^{-2}(W^2_{j_0k})\frac{\partial}{\partial W^1_{i_0j_0}}(X^2_{j_0})$$

$$\frac{\partial E}{\partial W^1_{i_0 j_0}} = -\sum_{k=1}^{K}(d_k - y_k)(\cosh \sum_{j=1}^{J+1} w^2_{jk_0} x^2_j)^{-2}(W^2_{j_0 k})(\cosh \sum_{i=1}^{I+1} w^1_{ij_0} x^1_i)^{-2} \frac{\partial}{\partial W^1_{i_0 j_0}} \{\sum_{i=1}^{I+1} w^1_{ij_0} x^1_i\}$$

$$\frac{\partial E}{\partial W^1_{i_0 j_0}} = -\sum_{k=1}^{K}(d_k - y_k)(\cosh \sum_{j=1}^{J+1} w^2_{jk_0} x^2_j)^{-2}(W^2_{j_0 k})(\cosh \sum_{i=1}^{I+1} w^1_{ij_0} x^1_i)^{-2}(X^1_{i_0})$$

$$W^{1+}_{i_0 j_0} = W^{1-}_{i_0 j_0} + \eta \sum_{k=1}^{K}(d_k - y_k)(\cosh \sum_{j=1}^{J+1} w^2_{jk_0} x^2_j)^{-2}(W^2_{j_0 k})(\cosh \sum_{i=1}^{I+1} w^1_{ij_0} x^1_i)^{-2}(X^1_{i_0})$$

Therefore:

$$W^{1+}_{i_0 j_0} = W^{1-}_{i_0 j_0} + \eta \sum_{k=1}^{K}(d_k - y_k)(1 - (y_{k_0})^2)(W^2_{j_0 k})(1 - (x^2_{j_0})^2)(X^1_{i_0})$$

A.1.4 Case IV: Tanh-Linear. For the output layer, the learning law is the same as the learning law derived earlier in Case II.

$$W^{2+}_{j_0 k_0} = W^{2-}_{j_0 k_0} + \eta(d_{k_0} - y_{k_0})(X^2_{j_0})$$

For the hidden layer weights the derivation is provided.

$$W^{1+}_{i_0 j_0} = W^{1-}_{i_0 j_0} - \eta \frac{\partial E}{\partial W^2_{i_0 j_0}}$$

The partial derivative term of the expression above is analyzed below.

$$\frac{\partial E}{\partial W^1_{i_0 j_0}} = \frac{\partial}{\partial W^1_{i_0 j_0}} \{\frac{1}{2} \sum_{k=1}^{K}(d_k - y_k)^2\} = -\sum_{k=1}^{K}(d_k - y_k) \frac{\partial y_k}{\partial W^1_{i_0 j_0}}$$

$$\frac{\partial E}{\partial W^1_{i_0 j_0}} = -\sum_{k=1}^{K}(d_k - y_k)(W^2_{j_0 k}) \frac{\partial}{\partial W^1_{i_0 j_0}} \{\sum_{i=1}^{I+1} w^1_{ij_0} x^1_i\}$$

$$\frac{\partial E}{\partial W^1_{i_0 j_0}} = -\sum_{k=1}^{K}(d_k - y_k)(W^2_{j_0 k})(\cosh \sum_{i=1}^{I+1} w^1_{ij_0} x^1_i)^{-2}(X^1_{i_0})$$

Therefore:

$$W_{i_0 j_0}^{1+} = W_{i_0 j_0}^{1-} + \eta \sum_{k=1}^{K} (d_k - y_k)(W_{j_0 k}^{2})(1 - (x_{j_0}^{2})^2)(X_{i_0}^{1})$$

Appendix B. Gradient Descent Search Algorithms

B.1 Introduction

In this appendix the concepts of momentum and the conjugate gradient method as means of accelerating the convergence of the gradient descent search of the MLP are introduced.

B.2 Momentum

The backward error propagation technique discussed in Chapter 2 performs a gradient descent search. Although this technique is very useful, it converges slowly at times. In an effort to speed up the convergence, a momentum term can be added to the learning law. The generalized learning law introduced in Chapter 2 is shown below.

$$W^{+} = W^{-} - \eta \frac{\partial E}{\partial W}$$

Momentum is defined below.

$$\Delta W = W^{-} - W^{--}$$

Adding momentum to the learning law yields the following equation.

$$W^{+} = W^{-} - \eta \frac{\partial E}{\partial W} + \alpha \Delta W$$

α is similar to η in that it is simply a constant. Once α is selected, the size of η is critical to optimize the performance of the gradient descent. If η is too large for a given α, wide oscillations will occur in the gradient descent search. If η is too small for a given α, the result will be a very slow learning rate. Figure 21 illustrates momentum in a vector sense.

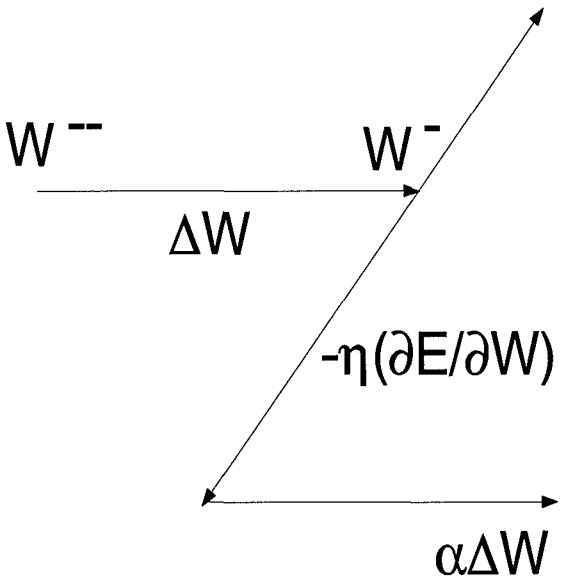

Figure 21. Momentum in a Vector Sense

For further study into acceleration methods of the MLP using momentum, the reader is encouraged to explore the presentation by Rumelhart (20).

B.3 Conjugate Gradient Algorithm

The conjugate gradient method is analogous to momentum in that it attempts to accelerate the convergence of the gradient descent search. The variables used to describe the conjugate gradient method are shown in Table 21.

Table 21. Conjugate Gradient Algorithm Variables

Variable	Description
E	mean squared error over an epoch (objective function)
W	MLP weights
G	gradient vector of objective function
D	search direction vector
α	search distance coefficient
β	deflection coefficient

The conjugate gradient algorithm is shown below.

Step 1: Set the initial weights (W) randomly.

Step 2: Calculate the initial gradient for one epoch.

$$G = \frac{\partial E}{\partial W}$$

Step 3: Set the initial search direction vector to be the negative of the gradient.

$$D = -G$$

Step 4: Conduct a Fibonacci line search for α to minimize the error, E(W+αD).

Step 5: Calculate the weights using the learning law below.

$$W^+ = W^- + \alpha D$$

Step 6: Calculate the new gradient while saving the old gradient.

$$G^+ = \frac{\partial E}{\partial W}$$

Step 7: Calculate the deflection term.

$$\beta = \frac{(G^+ - G^-)^T (G^+)}{(G^-)^T (G^-)}$$

Step 8: Calculate a new search direction which should be nearly orthogonal to last search direction.

$$D^+ = -G^+ + \beta D^-$$

Step 9: Go to step 4 as long as E is greater than some specified arbitrary constant or for some specified number of iterations.

The results of an experiment using XOR data are shown in Figure 22. The experiment was set up to continue as long as E was greater than 0.02 and as can be seen in Figure 22, only 10 epochs were required to reach this goal. In contrast to these results, an MLP without conjugate gradient search required 100 epochs to achieve the same results on the same data.

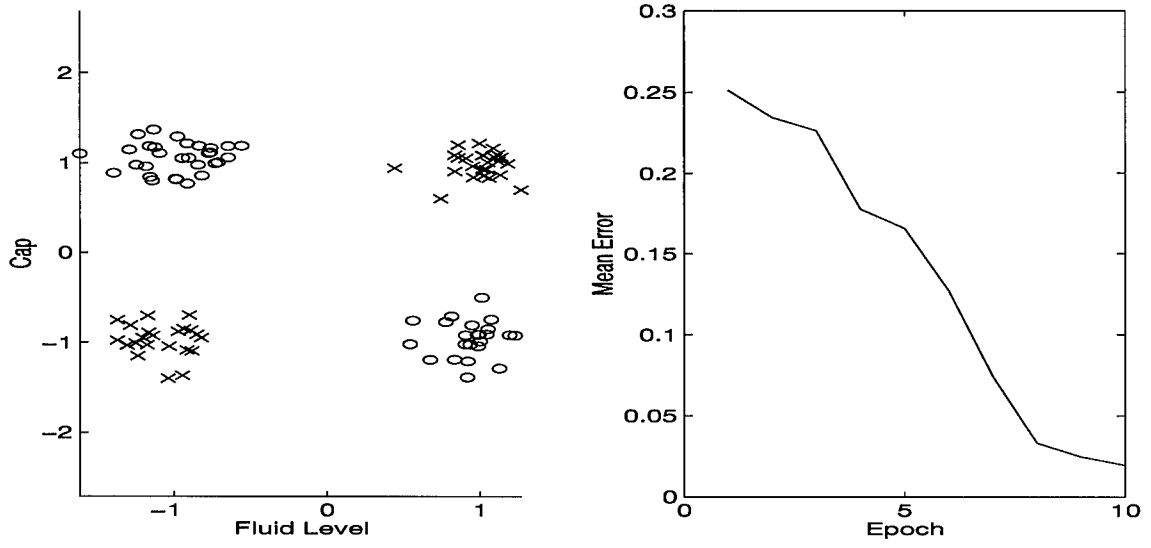

Figure 22. Results of the Conjugate Gradient Experiment on XOR Data

Appendix C. Derivation of the Ruck Saliency Metric

C.1 Introduction

In this appendix the Ruck Saliency metric is derived. Refer to Section 2.3 for the notation presented here.

C.2 Derivation

The definition of activation is the weighted sum of the input values plus the threshold as shown below for the output layer.

$$a_k^2 = \sum_{j=1}^{J+1} W_{jk}^2 x_j^2$$

(Note: The output of the hidden nodes is defined as x_j^2)

The output of the MLP is y_k and using a sigmoidal transformation it is shown below.

$$y_k = f_H(a_k^2) = \frac{1}{1+e^{-a_k^2}}$$

The first step in deriving the Ruck saliency metric involves finding the derivative of the output with respect to the input as shown below.

$$\frac{\partial y_k}{\partial x_i} = \frac{\partial f_H(a_k^2)}{\partial x_i} = y_k(1-y_k)\frac{\partial a_k^2}{\partial x_i}$$

$$\frac{\partial y_k}{\partial x_i} = y_k(1-y_k)\frac{\partial}{\partial x_i}(\sum_{j=1}^{J+1} W_{jk}^2 x_j^2)$$

$$\text{Let } \delta_k^2 = y_k(1-y_k)$$

$$\frac{\partial y_k}{\partial x_i} = \delta_k^2 \sum_{j=1}^{J+1} W_{jk}^2 \frac{\partial x_j^2}{\partial x_i}$$

$$\text{Since } x_j^2 = f_H(\sum_{i=1}^{I+1} W_{ij}^1 x_i^1),$$

$$\frac{\partial y_k}{\partial x_i} = \delta_k^2 \sum_{j=1}^{J+1} W_{jk}^2 x_j^2 (1 - x_j^2) \frac{\partial a_j^1}{\partial x_i}$$

$$\text{Let } \delta_j^1 = x_j^2 (1 - x_j^2)$$

$$\frac{\partial y_k}{\partial x_i} = \delta_k^2 \sum_{j=1}^{J+1} W_{jk}^2 \delta_j^1 W_{ij}^1$$

If Λ_i represents the saliency of input i, saliency is now defined as shown below.

$$\Lambda_i = \sum_{k=1}^{K} \mid \frac{\partial y_k}{\partial x_i} \mid = \sum_{k=1}^{K} \mid \delta_k^2 \sum_{j=1}^{J+1} W_{jk}^2 \delta_j^1 W_{ij}^1 \mid$$

The resulting saliency metric measures the usefulness of each input feature for determination of the correct output class.

Appendix D. Chemical Compound Legend

D.1 Introduction

In this appendix the legend of chemical compounds is provided in table 22. The abbreviated chemical compounds are used in Section 4.

D.2 Chemical Compound Legend

Chemical Compound	*Abbreviation*
Benzonitrile	benzonitrile
C_7H_{12} Isomer	c7h12
Carbontetrachloride	tetra
Chloroacetone	chloroace
Chlorobenzene	chlorobenz
Ethyl Acetate	ethace
Methylmethacrylate	methmeth
Phenol/n-Propylbenzene	phenol
Saturated Hydrocarbon # 2	sat2
Unknown Flourinate	unkflo
1-Butanol	1butanol
1,2-Dichloroethane	12di
2-Butanal	2butanal
2-Hexanone	2hex
2-Methylfuran	2methfur
2-Methylpropenal	2methprop
2-Octanone/Benzofuran	2octbenz
2-Pentanone	2pent
n-Pentanal	npent
o-Dichlorobenzene	odi
p-Chlorotoluene	pchloro

Table 22. Chemical Compound Key

Appendix E. Feature Saliency Code

E.1 Introduction

In this appendix the MATLAB® code created by William Polakowski is presented for the reader. This code will compute the Ruck feature saliency metric in an output labelled dz/dx. The data must be in LNKnet format to use this code.

E.2 Master Code (neural.m)

```
% Top level neural net and feature saliency script

fprintf(1, '\n');
fprintf(1, ' The next inputs allow you to specify the neural net.\n');
fprintf(1, '\n');
fprintf(1, ' This assumes your data file is already loaded in Matlab with the name "data" .\n');
fprintf(1, '\n');

k = input('Specify the number of middle nodes (default = 3): ');

if k == []

k = 3;
end

fprintf(1, '\n');

maxerr = input('Specify the maximum epoch error (default = 0.01 for 1% error): ');

if maxerr == []

maxerr = 0.01;
end

fprintf(1, '\n');

maxepochs = input('Specify the maximum number of epochs per iteration (default = 25): ');

if maxepochs == []

maxepochs = 25;
end

fprintf(1, '\n');

fprintf(1, 'Specify the number of folds:');
fprintf(1, '\n');
fprintf(1, '  input "# of samples" for leave one out method \n');
fprintf(1, '  input "2" for half and half Cross Validation \n');
fprintf(1, '  input "2 to # of samples" for other data partitioning \n');
fprintf(1, '\n');
```

```
fold = input('Number of folds (default = 2):  ');

if fold == []

fold = 2;
end

fprintf(1, '\n');

h = input('Specify the initial learning parameter step size (default = 1):  ');

if h == []

h = 1;
end

fprintf(1, '\n');

fprintf(1, 'Specify the nonlinear operators for the input and output layers:');
fprintf(1, '\n');
fprintf(1, '  1 for Sigmoid-Sigmoid \n');
fprintf(1, '  2 for Sigmoid-Linear \n');
fprintf(1, '  3 for Tanh-Tanh \n');
fprintf(1, '  4 for Tanh-Linear \n');
fprintf(1, '\n');

nonlinear = input('Nonlinear operator (default = 1):  ');

if nonlinear == []

nonlinear = 1;
end

fprintf(1, '\n');

[confusion,classify,dzdx,epoch_err,misfits,w1,w2] = saliency(data,k,maxerr,maxepochs,fold,nonlinear,h)
```

E.3 Slave Code (saliency.m)

```
%  SALIENCY:  A neural net with one hidden layer using various operators.
%  It computes feature saliency using Dr Ruck's derivative-based method.
%  This uses the number of folds for determining training and testing sets.
%
%     [CONFUSION,CLASSIFY,DZDX,EPOCH_ERR,MISFITS,W1,W2] = SALIENCY(DATA,K,MAXERR,MAXEPOCHS,FOLD,NONLINEAR,H)
%
%  Inputs: DATA: Training vectors (one sample per column)
%          K:       Number of hidden nodes
%          MAXERR:  Max average MSE allowed
%          MAXEPOCHS:   Max number of epochs allowed
%    FOLD: Specifies number of samples to train and test on.
% 2 = cross validation, # of samples = leave-one-out
%    H: Initial learning parameter
%
% Outputs: CONFUSION: Confusion matrix
%    CLASSIFY: Classification accuracy
%    DZDX: Lists the derivative-based feature saliencies
%          EPOCH_ERR:  Average mse for each epoch of the last iteration
```

```
%    MISFITS: Lists the misclassified samples
%     W1, W2:   First, second layer weights of trained net

function [confusion,classify,dzdx,epoch_err,misfits,w1,w2] = saliency(data,k,maxerr,maxepochs,fold,nonlinear,h)

% Determine the characteristics of the data and randomize the order

l = max(data(:,1))+1; % Class id starts with 0,1,2,...
[nsamples, nfeatures] = size(data); % Determine number of features & samples
nfeatures=nfeatures-1;
index2 = randperm(nsamples);

for i = 1:nsamples,

       x(i,:) = data(index2(i),:);

end %  (for i = 1 : nsamples)

data = x;

% Initialize variables

d  = zeros(l,l);
confusion  = zeros(l);
test_position = [];
count = 1;
misfits  = [];
dzdx1 = [];
dzdx = zeros(fold,nfeatures);
deltafprime1 = zeros(k,l);
deltafprime2 = zeros(l);
reseth  = h;
eta1  = h;
missed = zeros(1,nsamples);

if nonlinear == 2 | nonlinear == 4

eta2 = h/2;
else
eta2 = h;

end

% Start the net!!

for foldnumber = 1:fold

% Split the data into equal training and test sets.
% The classes are equally represented in each set.
% The size of the split is determined by the number of folds specified.

datatrain = [];
datatest = [];

for m = 0 : 1 - 1 %  Determines the minimum samples in a class

class_position = find(data(:,1)==m);
max_samples(1,m+1) = size(class_position,1);
```

```
end

max_fold = min(max_samples);

if fold <= max_fold %  Goes to leave one out method if false

for m = 0 : l - 1

class_position = find(data(:,1)==m);
class_samples = size(class_position,1);
split = ceil(class_samples/fold);

for p=1:class_samples

if p >= ((foldnumber - 1) * split + 1) & p <= foldnumber * split

datatest = [datatest; data(class_position(p),:)];
test_position = [test_position class_position(p)];
else
datatrain=[datatrain; data(class_position(p),:)];
end   %  (if p)
end %  (for p = ...)
end %  (for m = ...)

else

    if foldnumber==1
datatrain=data(2:nsamples,:);
datatest = data(1,:);
for m = 1 : nsamples
   test_position(m) = m;
end
    elseif foldnumber==nsamples
datatrain=data(1:nsamples-1,:);
datatest=data(nsamples,:);
    else
datatrain=[data(1:foldnumber-1,:); data(foldnumber+1:nsamples,:)];
datatest=data(foldnumber,:);
    end  % if

end % (if fold ~= nsamples)

trainsamples = size(datatrain,1);
testsamples =  size(datatest,1);

%  Normalize the features
%  Calculate the means and standard deviations of each feature in the training data.

ave=mean(datatrain(:,2:nfeatures+1));
dev=std(datatrain(:,2:nfeatures+1));

%  Normalize the training features

average = ones(trainsamples,1) * ave;
sigma = ones(trainsamples,1) * dev;
datatrain(:,2:nfeatures+1)=(datatrain(:,2:nfeatures+1)-average)./sigma;
```

```
% Normalize the test features with the training mean and standard deviation

average = ones(testsamples,1) * ave;
sigma = ones(testsamples,1) * dev;
datatest(:,2:nfeatures+1)=(datatest(:,2:nfeatures+1)-average)./sigma;

datatrain=datatrain';
datatest=datatest';

% Initialize weights and variables

w1 = rand(k, nfeatures+1) - 0.5;
w2 = rand(l, k+1) - 0.5;

err = [];
nepochs = 0;
epoch_err = 1;
h = reseth;

fprintf(1, 'Training network:\n');

while  nepochs < maxepochs & epoch_err > maxerr,

%    fprintf(1,' Epoch %d ... ',nepochs+1);

   % Clear the mse vector and get random presentation order

   mse = [];
   index = randperm(trainsamples);

   for i = 1:trainsamples,

      id = datatrain(1,index(i))+1;
      x = [datatrain(2:nfeatures+1,index(i)); 1];

      % Compute activations and their derivatives

if nonlinear==1 %  Sigmoid - Sigmoid operators

      z1 = 1 ./ (1 + exp(-w1 * x));
       z2 = 1 ./ (1 + exp(-w2 * [z1; 1]));
fprime1 = z1 .* (1-z1);
fprime2 = z2 .* (1-z2);

end

if nonlinear==2 %  Sigmoid - Linear operators

z1 = 1 ./ (1 + exp(-w1 * x));
       z2 = w2 * [z1; 1];
fprime1 = z1 .* (1-z1);
fprime2 = ones(1,1);

end

if nonlinear==3 %  Tanh - Tanh operators

      z1 = tanh(w1 * x);
```

```
        z2 = tanh(w2 * [z1; 1]);
fprime1 = 1-(z1.^2);
fprime2 = 1-(z2.^2);

end

if nonlinear==4 %  Tanh - Linear operators

       z1 = tanh(w1 * x);
        z2 = w2 * [z1; 1];
fprime1 = 1-(z1.^2);
fprime2 = ones(1,1);

end

%  Do the backpropagation weight correction

       %  Compute desired output d and the actual output's difference

       d(id) = 1.0;
       delta_out = fprime2 .* (d-z2);
       sigma = w2' * delta_out;
       delta_hid  = fprime1 .* sigma(1:k);

       %  Update the weights

w1 = w1 + eta1 * (delta_hid * x');
w2 = w2 + eta2 * (delta_out * [z1;1]');

       %  Compute mean square error for input, and reset desired output

mse(i) =  sum((d-z2).^2) / 1;
       d(id) = 0;

    end   % (for i = 1:trainsamples)

    %  Compute the epoch error

    epoch_err = mean(mse);
    err = [err epoch_err];
    nepochs = nepochs + 1;
%    fprintf(1, 'Average mse = %f\n', epoch_err);

    %  Vary the learning parameter

    if nepochs > 1

if  0.9 * err(nepochs-1) < err(nepochs) & err(nepochs) < err(nepochs-1) & h < 20

h = 1.5 * h;

end

if err(nepochs) > err(nepochs-1)

h = 0.5 * h;

end
```

```
    end

    eta1 = h * epoch_err;

    if nonlinear == 2 | nonlinear == 4

eta2 = eta1 / 2;

    else

eta2 = eta1;

    end % (if nonlinear ...)


end   % (while)

%  Compute the feature saliency

fprintf(1, 'Computing feature saliency:\n');

for i = 1:trainsamples,

    x = [datatrain(2:nfeatures+1,i); 1];

        %  Compute activations and their derivatives

if nonlinear==1 %  Sigmoid - Sigmoid operators

        z1 = 1 ./ (1 + exp(-w1 * x));
         z2 = 1 ./ (1 + exp(-w2 * [z1; 1]));
fprime1 = z1 .* (1-z1);
fprime2 = z2 .* (1-z2);

end

if nonlinear==2 %  Sigmoid - Linear operators

z1 = 1 ./ (1 + exp(-w1 * x));
         z2 = w2 * [z1; 1];
fprime1 = z1 .* (1-z1);
fprime2 = ones(1,1);

end

if nonlinear==3 %  Tanh - Tanh operators

        z1 = tanh(w1 * x);
         z2 = tanh(w2 * [z1; 1]);
fprime1 = 1-(z1.^2);
fprime2 = 1-(z2.^2);

end

if nonlinear==4 %  Tanh - Linear operators

        z1 = tanh(w1 * x);
```

```
        z2 = w2 * [z1; 1];
fprime1 = 1-(z1.^2);
fprime2 = ones(1,1);

end

% Compute the feature saliencies

% Expand both vectors to matrices (k x 1) and (1 x 1)

for a=1:1
  deltafprime1(:,a) = fprime1;
end

deltafprime2=diag(fprime2);

% dzdx is a matrix containing each feature's saliency for all training samples

dzdx1 = sum(abs((w1(:,1:nfeatures)' * (((w2(:,1:k)' * deltafprime2) .* deltafprime1)))'));
dzdx(foldnumber,:) = dzdx(foldnumber,:) + dzdx1;

end % (for i=1:trainsamples)

% Test the remaining samples

fprintf(1, 'Testing network:\n');

for i = 1:testsamples,

   x = [datatest(2:nfeatures+1,i); 1];

   % Apply non-linearity to activations

if nonlinear==1 % Sigmoid - Sigmoid operators

       z1 = 1 ./ (1 + exp(-w1 * x));
        z2 = 1 ./ (1 + exp(-w2 * [z1; 1]));

end

if nonlinear==2 % Sigmoid - Linear operators

z1 = 1 ./ (1 + exp(-w1 * x));
        z2 = w2 * [z1; 1];

end

if nonlinear==3 % Tanh - Tanh operators

       z1 = tanh(w1 * x);
        z2 = tanh(w2 * [z1; 1]);

end

if nonlinear==4 % Tanh - Linear operators

       z1 = tanh(w1 * x);
        z2 = w2 * [z1; 1];
```

```
end


    %  Compile output data

    [maxpost, guess] = max(z2);
    if guess ~= datatest(1,i)+1

        misfits = [misfits; index2(test_position(count)) guess];

    end   % (if)

    count = count + 1;
    confusion(datatest(1,i)+1, guess) = confusion(datatest(1,i)+1, guess) + 1;

end   % (for i = 1:testsamples)

end   % (foldnumber=1:fold)

%  Outputs

dzdx=sum(dzdx)/max(sum(dzdx));
epoch_err=err;

classify=trace(confusion)/nsamples;
```

Bibliography

1. Cybenko, G. "Correction: Approximation by Superpositions of a Sigmoidal," *Mathematics of Control, Signals, and Systems*, *5*(4) (1992).
2. Devore, Jay L. *Probability and Statistics for Engineering and the Sciences* (Third Edition). Brooks/Cole Publishing Company, 1991.
3. Duda, Richard O. and Peter E. Hart. *Pattern Classification and Scene Analysis.* John Wiley and Sons, 1973.
4. EENG 617 Class, Air Force Institute of Technology. "Multilayer Perceptron: The Solution to the XOR Problem." Group Midterm Exam for EENG 617, Math Modeling of the Central Nervous System, February 1995.
5. Frankel, Donald S., et al. "Use of a Neural Net Computer System for Analysis of Flow Cytometric Data of Phytoplankton Populations," *Cytometry*, *10*:540–550 (April 1989).
6. Fukunaga, Keinosuke. *Introduction to Statistical Pattern Recognition.* Boston: Academic Press Inc., 1990.
7. Fukunaga, Keinosuke and Donald M. Hummels. "Bayes Error Estimation Using Parzen and k-NN Procedures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-9*(5) (September 1987).
8. Hofstad, Tor. "Utility of Newer Techniques for Classification and Identification of Pathogenic Anaerobic Bacteria," *Clinical Infectious Diseases*, *18*:S250–S252 (1994). Supplemental.
9. Lachenbruch, Peter A. "An Almost Unbiased Method of Obtaining Confidence Intervals for the Probability of Misclassification in Discriminant Analysis," *Biometrics*, *23* (1967).
10. Lippmann, Richard P. "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, 4–22 (April 1987).
11. Long, James R., et al. "Pattern Recognition of Jet Fuel Chromatographic Data by Artificial Neural Networks with Back-Propagation of Error," *Analytical Chemistry*, *63*(13):1256 – 1261 (July 1991).
12. Martin, Curtis E. *Non-Parametric Bayes Error Estimation For UHRR Target Identification.* MS thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base Ohio, December 1993.
13. Martin, Curtis E., et al. "Nonparametric Bayes Error Estimation for HRR Target Identification." *Applications of Artificial Networks V*, Proc. SPIE 2243, edited by Steven K. Rogers and Dennis W. Ruck. 2 – 10. 1994.
14. Parsons, Thomas W. *Voice and Speech Processing.* McGraw-Hill, Inc., 1987.
15. Priddy, K. L., et al. "Bayesian Selection of Important Features for Feedforward Neural Networks," *Neurocomputing* (1992).

16. Raymer, J. H., et al. "A Nonrebreathing Breath Collection System for the Study of Exogenous and Endogenous Compounds in the Fisher-344 Rat," *Toxicology Methods*, *4*(4):243 – 258 (1994).

17. Rogers, Steven K., et al. *An Introduction to Biological and Artificial Neural Networks.* Air Force Institute of Technology, 1990.

18. Rogers, Steven K. "Class Lecture, EENG 617, Math Modeling of the Central Nervous System." School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, Winter Quarter 1995.

19. Ruck, Dennis W., et al. "Feature Selection Using a Multilayer Perceptron," *Journal of Neural Network Computing*, *2*(2) (Fall 1990).

20. Rumelhart, David E., et al. *Learning Internal Representations by Error Propagation, September 1985.* Technical Report, University of California, San Diego: Institute for Cognitive Science, September 1985.

21. Steppe, Jean M. *Feature and Model Selection in Feedforward Neural Networks.* PhD dissertation, Air Force Institute of Technology, 1994.

22. Stewart, James A. *Nonlinear Time Series Analysis.* MS thesis, Air Force Institute of Technology, 1995.

23. Tarr, G. L. *Multi-Layered Feedforward Neural Networks for Image Segmentation.* PhD dissertation, Air Force Institute of Technology, 1991.

Vita

Capt Robert E. Sackett Jr. [redacted] He graduated from high school in Newton Falls, Ohio in 1985, and earned a bachelors degree in Electrical Engineering at the University of Akron in 1990. He was commissioned through the Air Force Reserve Officer Training Corps at the University of Akron. Upon graduation, he entered active duty with the USAF at K. I. Sawyer AFB, Michigan where he served as an Electrical Engineer, Squadron Section Commander, and Readiness Flight Chief. He graduated from the Air Force Institute of Technology with a Masters in Engineering and Environmental Management from class GEE-95D in 1995.

Robert married the former Lisa M. Stutler in 1991, and they have one daughter, Lauren.

[redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1995	Master's Thesis

4. TITLE AND SUBTITLE
Neural Network Analysis of Chemical Compounds in Nonrebreathing Fisher-344 Rat Breath

5. FUNDING NUMBERS

6. AUTHOR(S)
Robert E. Sackett Jr.
Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GEE/ENG/95D-02

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
NA

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This research applies statistical and artificial neural network analysis to data obtained from measurement of organic compounds in the breath of a Fisher-344 rat. The Research Triangle Institute (RTI) developed a breath collection system for use with rats in order to collect and determine volatile organic compounds (VOCs) exhaled. The RTI study tested the hypothesis that VOCs, including endogenous compounds, in breath can serve as markers to exposure to various chemical compounds such as drugs, pesticides, or carcinogens normally foreign to living organisms. From a comparative analysis of chromatograms, it was concluded that the administration of carbon tetrachloride dramatically altered the VOCs measured in breath; both the compounds detected and their amounts were greatly impacted using the data supplied by RTI. This research will show that neural network analysis and classification can be used to discriminate between exposure to carbon tetrachloride versus no exposure and find the chemical compounds in rat breath that best discriminate between a dosage of carbon tetrachloride and either a vehicle control or no dose at all. For the data set analyzed, 100 percent classification accuracy was achieved in classifying two cases of exposure versus no exposure. The top three marker compounds were identified for each of three classification cases. The results obtained show that neural networks can be effectively used to analyze complex chromatographic data.

14. SUBJECT TERMS
Neural Networks, Bayes Error Estimation, Feature Selection, Breath Analysis

15. NUMBER OF PAGES
76

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL