12-1992

Computational Neural Networks in Chemistry: Model Free Mapping Devices for Predicting Chemical Reactivity from Molecular Structure

David Wayne Elrod
Western Michigan University

COMPUTATIONAL NEURAL NETWORKS IN CHEMISTRY: MODEL FREE MAPPING DEVICES FOR PREDICTING CHEMICAL REACTIVITY FROM MOLECULAR STRUCTURE

by

David Wayne Elrod

A Dissertation
Submitted to the
Faculty of The Graduate College
in partial fulfillment of the
requirements for the
Degree of Doctor of Philosophy
Department of Chemistry

Western Michigan University
Kalamazoo, Michigan
December 1992

COMPUTATIONAL NEURAL NETWORKS IN CHEMISTRY: MODEL FREE MAPPING DEVICES FOR PREDICTING CHEMICAL REACTIVITY FROM MOLECULAR STRUCTURE

David Wayne Elrod, Ph.D.

Western Michigan University, 1992

Computational neural networks (CNNs) are a computational paradigm inspired by the brain's massively parallel network of highly interconnected neurons. The power of computational neural networks derives not so much from their ability to model the brain as from their ability to learn by example and to map highly complex, nonlinear functions, without the need to explicitly specify the functional relationship. Two central questions about CNNs were investigated in the context of predicting chemical reactions: (1) the mapping properties of neural networks and (2) the representation of chemical information for use in CNNs.

Chemical reactivity is here considered an example of a complex, nonlinear function of molecular structure. CNN's were trained using modifications of the backpropagation learning rule to map a three dimensional response surface similar to those typically observed in quantitative structure-activity and structure-property relationships. The computational neural network's mapping of the response surface was found

to be robust to the effects of training sample size, noisy data and intercorrelated input variables.

The investigation of chemical structure representation led to the development of a molecular structure-based connection-table representation suitable for neural network training. An extension of this work led to a BE-matrix structure representation that was found to be general for several classes of reactions. The CNN prediction of chemical reactivity and regiochemistry was investigated for electrophilic aromatic substitution reactions, Markovnikov addition to alkenes, Saytzeff elimination from haloalkanes, Diels-Alder cycloaddition, and retro Diels-Alder ring opening reactions using these connectivity-matrix derived representations. The reaction predictions made by the CNNs were more accurate than those of an expert system and were comparable to predictions made by chemists.

Computational neural networks were shown to have robust mapping properties and were capable of giving excellent predictions of chemical reactivity when trained with suitable molecular structure representations. The CNN methodology developed here may be useful for extracting reactivity rules from databases of chemical reactions.

Order Number 9310447

Computational neural networks in chemistry: Model free mapping devices for predicting chemical reactivity from molecular structure

Elrod, David Wayne, Ph.D.

Western Michigan University, 1992

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

ACKNOWLEDGMENTS

It is difficult, if not impossible, to adequately express my appreciation to everyone to whom I am indebted for helping me to fulfil this dream. I would like to especially thank Dr. Gerald Maggiora, Director of Computational Chemistry at The Upjohn Company, Kalamazoo, Michigan, for his enthusiastic support of my research, for serving as my mentor and for sharing his vision for computational chemistry. Thanks, Gerry, for being a highly effective leader who encourages and inspires all he leads.

I would like to thank my committee for their willingness to take a chance on a research project that spanned the disciplines of chemistry and computer science. Thanks to Dr. Robert Trenary for his deep insights into neural networks, for his encouragement and excellent teaching ability. To Dr. Robert Harmon, chairman of my committee, goes my sincere appreciation for sharing his interest in computers applied to organic chemistry problems. I express my appreciation to the other committee members, Dr. Michael McCarville, Dr. John Kapenga, and Dr. William Kelly, for their guidance, and for the lessons they taught me.

My wife Nikki, and son Tim deserve special thanks for their persistent support and encouragement over many long years.

I dedicate this dissertation to the memory of Dr. Paul Wiley, my first

research director, who taught me the importance of the rigorous pursuit of excellence in research. He built on the foundation laid by Dr. Barney Magerlein, my SIP advisor, and the chemistry faculty at Kalamazoo College, who gave me an excellent preparation for chemistry research. The "K" professors included Dr. Kurt Kaufman, who taught me the wonders of organic chemistry, and Dr. Ralph Deal, who first exposed me to computational chemistry. Thanks also belongs to my parents, James and Doris Elrod, who instilled in me a lifelong love of reading and learning. Indirectly, Professor E. J. Corey of Harvard deserves partial credit for leading me astray from pure synthetic organic chemistry and into the realm of computer-assisted organic synthesis. Corey's development of the LHASA synthesis design program served as the inspiration for much of this work.

I would also like to express my sincere appreciation to The Upjohn Company for providing me with challenging employment in pharmaceutical research and for providing educational assistance for this dissertation.

In science, as in all of life, we build upon the foundation left by those who went before us. Sometimes the influences of others on us, and the influences we have on others, are not recognized until later. May those who come behind us find us faithful in all that we pass on. *Lux Esto.*

David Wayne Elrod

TABLE OF CONTENTS

ACKNOWLEDGEMENTS ii

LIST OF TABLES..... x

LIST OF FIGURES..... xi

CHAPTER

I. INTRODUCTION..... 1

Introduction to Computer Assisted Organic Synthesis Methods 1

Introduction to Computational Neural Networks (CNN) 6

Characteristics of Computational Neural Networks 10

Neural Network Simulators 17

Example of Training a Computational Neural Network 19

Important Computational Neural Network Issues 26

Data Representation 28

Network Optimization 30

Generalization 33

Error Analysis 34

Survey of Chemical Applications of Neural Networks 36

Table of Contents--Continued

CHAPTER

Reviews, Theoretical Studies, and Miscellaneous Applications 40

Chemical Process Control and Chemical Engineering Applications 41

Analytical Chemistry Applications 42

Protein and Polymer Structure and Analysis Applications 43

Biomolecular Informatics Applications 44

Quantitative Structure-Activity Relationships and Pattern Recognition Applications 45

Spectra-Structure Correlation Applications 46

Property Prediction and Parameter Estimation Applications 48

Nuclear Chemistry and Nuclear Power Applications 49

Overview 50

II. COMPUTATIONAL NEURAL NETWORKS: MODEL-FREE MAPPING DEVICES 51

Introduction 51

Mapping Methods 54

Model Problem: Mapping a 3D Response Surface With a Neural Network 56

Table of Contents--Continued

CHAPTER

Sample Size 62

Transfer Functions 68

Noisy Data 69

Representation Issues: Correlated Variables 72

Simple Chemical Relationships Mapped by a Neural Network 74

Prediction of ^{17}O NMR Shifts 75

Nonlinear or Bilinear Data Mapped by CNN 75

Summary 76

III. NEURAL NETWORK PREDICTION OF REACTIVITY IN EAS REACTIONS 80

Representation of Chemical Structures for Neural Networks 80

Background 80

General Requirements for a Representation 81

Graph Theoretical Index Representation for Predicting Boiling Points 84

Connection Table Representations 88

Semi-Canonical Numbering of Atoms in Connection Table 90

CHAPTER

Modified Connection Table for Neural Net Input 94

Other Representations 97

Charge Vector 97

Graph Transforms 97

Randic Indices 98

CNN Prediction of Electrophilic Aromatic Substitution Reactions 100

Selection and Background of the Problem 100

Methods 103

Data 103

Connection Table Representation 105

Charge Vector Representation 106

Network Simulator 106

Network Configuration 107

Network Training and Testing 108

Comparison With Non-Neural Net Prediction Methods 111

Results For Connection Table and Charge Vector CNNs 112

CHAPTER

Results for Graph Transforms and Randic Indices 116

Graph Transforms 116

Randic Indices 120

Summary 122

IV. EXTENSION OF CNN PREDICTION TO OTHER REACTIONS 124

Introduction 124

Ugi B-E Matrix Representation of Reactions 125

Methods 131

Neural Network Simulator 131

Neural Network Configuration 132

Neural Network Training 135

Results 136

Markovnikov Addition Reactions 136

Diels-Alder Cycloaddition and Retro Diels-Alder Reactions 138

Saytzeff Elimination Reactions 138

Summary 139

Table of Contents--Continued

CHAPTER

V. CONCLUSIONS AND FUTURE DIRECTIONS 144

Results and Conclusions 144

Mapping Properties 145

Molecular Representation for Reaction Prediction 147

Challenges and Opportunities for the Future 149

Final Comments 153

REFERENCES 154

LIST OF TABLES

1. Neural Network Characteristics 16

2. Types of Published Neural Network Applications in Chemistry 39

3. Characteristics of Feedforward Neural Net Descriptions of the "Three Gaussian" Response-Surface Model. 64

4. CNN 2-2-1 Predictions of Boiling Points of Octanes from Paths P2 & P3 87

5. Graph Theoretic Transforms for EAS Reaction and CNN Results 118

6. Randic Indices and Results for EAS Reactions 121

7. CNN Results for Markovnikov Reaction 137

8. CNN Results for Diels-Alder and Retro-Diels-Alder Reactions 140

9. CNN Results for Saytzeff Reactions 141

LIST OF FIGURES

1. Example of a Multi-Layer, Feedforward Neural Network, Also Known as a Generalized Perceptron 8

2. Example of a Nonlinear Transfer Function *tanh* 8

3. E-State Index & ^{17}O NMR Shift for 19 Ethers and Carbonyl Compounds..... 20

4. CNN Weights Before and After Training 22

5. Plot of E-state *vs* ^{17}O NMR shifts for Untrained and Trained Nets 27

6. Histogram Showing the Number of Chemical Applications of CNN Papers per Year From 1964 to the Present 38

7. Mapping *vs* Representation Problem 52

8. Mapping Methods 55

9. Response Surface 58

10. Data Sets of 1600, 156, 104, and 52 samples 61

11. CNN Mapped Response Surfaces Trained on 900, 156, 104, & 52 Samples 65

12. Outputs From Hidden Layer 1 PEs 66

13. Outputs From Hidden layer 2 PEs 67

14. CNN Mapped Response Surfaces Trained on Noisy Data 71

15. Phorbol Esters: Plot of Tumor Promoting Ability *vs* CLogP 77

16. Octanes: Structures, Boiling Points and Paths 86

17. Connection Table and Connectivity Matrix for Acetanilide 90

18. Connection Table Representation Suitable for Neural Net Input 96

19. Other Representations for Acetanilide 99

20. EAS Reaction: Mechanism and *o, m, p* Products 101

21. EAS Literature Data Ranked by Percent of Meta Product 104

22. Network Used to Predict EAS Reactions From Connection Table 109

23. EAS Reaction. CNN Results on Training Set Reactions 114

24. EAS Reaction. CNN Results on Test Set Reactions 115

25. Markovnikov Addition of Hydrogen Bromide to Alkenes 126

26. Diels-Alder and Retro Diels-Alder Reactions 127

27. E1 (Saytzeff's Rule) Elimination to Form Alkenes 128

28. BE-Matrix Representation of Reactions 130

29. Neural Network for Reaction Prediction With BE-Matrices 133

30. BE-Matrix Network Sizes and Results for Reaction Prediction 134

CHAPTER I

INTRODUCTION

Introduction to Computer Assisted Organic Synthesis Methods

Organic chemists have been employing computer methods to assist in the design and analysis of organic reactions for over 20 years.[1] Considerable progress has been made during this time, but many of these methods have received only limited acceptance by laboratory chemists. Noordik, in the introduction to the April 1992 issue of *Recueil des Travaux Chimiques des Pays-Bas*, which contained eleven articles entirely devoted to Computer Assisted Organic Synthesis (CAOS), says[2]

> Although it is now well recognized that the direct influence of these programs on the day-to-day synthetic work in the laboratory will not be revolutionary, the impact of many ideas which have evolved from computer-application related research is substantial. Computer storage and retrieval of structures and reactions is common practice in industrial and academic laboratories. Synthesis design and reaction prediction programs have matured to a state in which they can contribute to the design process and are of great potential value in education.

In a review article in the same journal,[3] Ott and Noordik speculate that synthetic chemists do not commonly use CAOS programs because the enormous size and complexity of the "search space" for synthetic problems

has been a major obstacle for software development. They also cite the fact that the "synthetic way of thinking is much more structure-oriented than that in many physico-chemical disciplines, making the lack of affordable graphics hardware in most organic laboratories an obstacle to the use of the more advanced programs."(p 239)[3] Additionally, many such CAOS programs either generate too many possibilities to be useful or else they cannot make predictions about the reactions of interest because the applicable reactivity rules have not yet been encoded into the program.

Chemists are better at solving many of the prediction problems they face in laboratory synthesis than are the computer programs that are available. This is particularly true for problems that require judgement, extrapolation from analogous situations, or the application of conflicting or overlapping rules. Chemists have developed chemical intuition through training and experience and can bring this chemical intuition to bear on synthesis problems. Many rules about chemical reactivity have been developed but their application is often difficult because there may be conflicting or overlapping rules that apply. For example, in predicting the products of electrophilic aromatic substitution, the resonance effects of substituents may reinforce or oppose the inductive effects of the substituents, and there may also be steric effects, depending on the size and location of the substituents.

The field of artificial intelligence (AI) has led to "expert systems,"

which attempt to mimic a human expert by applying a set of rules, developed by interviewing an expert in that area. These AI-based expert systems can make limited judgments and inferences by applying their rules or heuristics but often fail for ambiguous cases, where human experts may succeed. The AI-inspired CAOS programs generally use one of three methods to apply chemical knowledge to synthesis problems: (1) a library of reactions or transforms, as in the LHASA program[4]; (2) mathematical models to generate all possible products or precursors, such as the IGOR,[5] EROS,[6] and SYNGEN[7] programs; or (3) mechanistic rules governing reaction types, as in CAMEO.[8] Gasteiger recently reported the WODCA[9] program which combines mathematical models for synthesis design with mechanistic reasoning for reaction prediction. All of these methods have had some success, but are limited by the requirement that rules governing reactivity must be explicitly stated or heuristics must be devised to prune the number of possibilities to a manageable size. Rule-based expert systems have the advantage that they may be queried to determine how the rules have been applied to achieve a given result. However, expert systems have become very complex and deciphering the implications of the sequence of rules applied by the expert system may be difficult.

The generation of reaction rules from the chemical literature or from interviews with expert chemists is a difficult and tedious process that limits the rate at which progress can be made. The rate at which chemical

reaction information is being stored in computer-readable form is increasing rapidly, judging by the size of the CASREACT[10] chemical reaction database from Chemical Abstracts Service. A more automated method of extracting chemical knowledge from these computerized databases could enhance the rate of improvement in the CAOS methods and would benefit the synthetic chemists who might use them for idea-generation and troubleshooting. This desire to find improved methods for extracting chemical reaction information and applying it to predict synthetic and retrosynthetic reactions was the driving force for my research. Coupled with this desire to find automated methods was the opportunity to apply the newly revitalized field of computational neural networks, which were claimed to have the ability to perform feature extraction and to map complex functions.

When our research in this area commenced, there were no examples of neural networks being applied to the prediction of chemical reactions. Neural network applications to the prediction of organic reactions and reaction products are still very sparse, with only seven publications in this category to date (September 1992), three of which are derived from portions of this thesis work. A forerunner of our neural network approach can be found in the work of Wilcox and Levinson, who in 1986 described a "self-organized" method for the design and discovery of knowledge about organic chemistry.[11] Schulz, Hoffmann, and Gasteiger published an associative

memory method, which in this case was not implemented in a neural network, in 1988 in an obscure conference proceedings in German, to predict which bonds could be broken in chemical reactions, based on similarity to the training compounds.[12] Other than these two papers, there was no work on applying neural network methods to predicting chemical reactivity from chemical structure.

Some of the introductory material in Chapter I of this thesis, as well as a portion of the electrophilic aromatic substitution reaction predictions found in Chapter III, was published in 1991 in the proceedings of the First Great Lakes Computer Science Conference,[13] held at Western Michigan University in October 1989, and also by Elrod, Maggiora, and Trenary in the *Journal of Chemical Information and Computer Science* in 1990.[14] The work in Chapter IV on feedforward neural network learning of simple reactivity rules for elimination, addition and cycloaddition reactions was published in 1990 in *Tetrahedron Computer Methodology*.[15] Kvasnicka (1991) later studied the same electrophilic aromatic substitution reaction as we did, but used chemical graphs as network inputs.[16] Zou, Johnson and Tsai, based upon reports of our work,[13] also looked at predicting the same set of aromatic substitution reactions but using graph-theoretic transforms.[17] Concurrently with our research, Luce and Govind[18] proposed a hybrid neural network-expert system for retrosynthetic analysis, which they published in 1990. Recently, Marie, Nicolle-Adam, and Villemin

used a machine learning approach to extract reaction information from chemical reaction graphs.[19] Portions of the material in Chapter II on the model response surface will be reported in late 1992 in the *Journal of Chemical Information and Computer Science*.[20] The review of the literature on neural network applications in chemistry found in Chapter I, will appear in the proceedings of Online Information 92.[21]

Introduction to Computational Neural Networks (CNN)

Although based loosely on analogies to the brain, interest in neural networks derives not from their ability to model the brain, but rather from their ability to treat a diverse set of problems using a highly-distributed parallel computer architecture.[22,23,24,25,26] Neural networks, sometimes called artificial neural networks or computational neural networks (CNN) to emphasize their computational properties over their ability to model brain function, are made up of collections of highly-interconnected, but relatively simple processing elements (PE); each interconnection has an associated weight that specifies the strength of the connection. In general, CNNs are distinguished by their network paradigms, which define the nature of their PEs, the network topology or pattern of connectivity among the PEs, and the learning method. There are two general categories of learning methods, unsupervised and supervised.[22-26] In unsupervised learning the network itself determines the appropriate

set of weights, while in supervised learning weights are determined such that the error between a set of training data and network predictions is minimized. Unlike traditional computers that store programs and data separately, CNNs store information in the distributed pattern of their interconnections and values of the associated weights.

Of the many CNN paradigms that have been investigated, we will focus our attention here on multilayer, feedforward nets, also called generalized perceptrons, with either backpropagation or stochastic supervised learning.[27,28] These CNNs have been successfully applied to a wide variety of problems and are the most extensively studied to date. A typical example of such a CNN is depicted in Figure 1. As shown in Figure 1, each PE performs two operations, a summation denoted by "Σ", followed by a function evaluation denoted by "f"; f is called a transfer function (or activation function) and is often either a sigmoid function or the closely related hyperbolic tangent, *tanh*, function shown in Figure 2. Each PE also possesses a threshold or "bias weight," generally denoted Θ_i, which when combined with a constant, unit signal, effectively shifts the transfer function along its abscissa (*Cf.* Figure 1).

CNNs can be viewed in two ways, either as classifying or as mapping devices. Generally, but not always, classification represents a less stringent, although non-trivial, test of a CNNs performance than function mapping. This follows from the fact that in the latter case the value of the

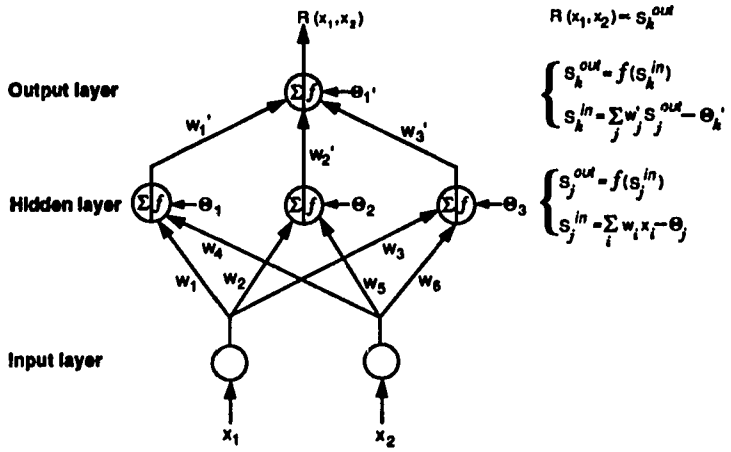


Figure 1. Example of a Multilayer, Feedforward Neural Network, Also Known as a Generalized Perceptron.

function over the domain of interest must be predicted, while in the former case only whether the function value lies above or below some threshold is required. Nevertheless, it has been shown by numerous workers that generalized perceptrons can accurately represent the mappings of

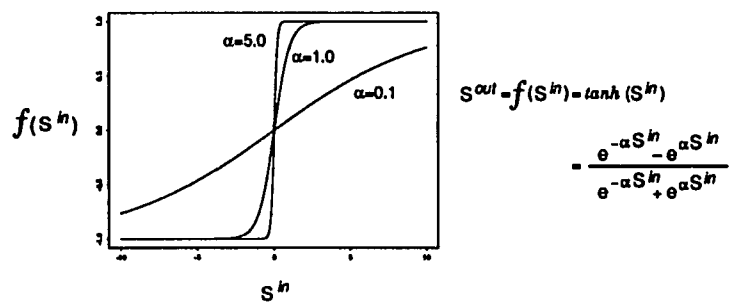


Figure 2. Example of a Nonlinear Transfer Function *tanh*.

essentially all reasonably well-behaved functions.[29,30,31,32,33] The proofs, however, are only proofs that a solution exists and unfortunately, do not indicate precisely how one should produce a CNN that can, in fact, carry out the desired mapping.

Kosko[34] has pointed out that CNNs can be considered as *model-free* mapping devices in so far as the functional form of the mapping need not be specified explicitly, in contrast to the situation in both linear and non-linear regression methods. While this may be true technically, it omits the fact that the network topology, *i.e.*, the number of hidden layers and the number of nodes in each layer, as well as the type of transfer function must at some point be specified, and this may be construed, in some sense, as being analogous to choosing a model. This model-free character would appear to provide an advantage to CNNs in cases, commonly encountered in chemistry, where complicated input-output relationships are an inherent feature of the system or systems under investigation.[35] To investigate such systems, however, is difficult due to the problem of finding suitably complete datasets. In Chapter II this issue is addressed by focusing on the study of a well-defined model system designed to simulate the salient features of response surfaces or functions, R, such as would typically be encountered in structure-activity or structure-property studies (*vide infra*). A distinct advantage of this approach is that a number of important problems can be addressed in a more "controlled" manner.

Characteristics of Computational Neural Networks

In close analogy to biological neural nets, CNNs can be viewed as a set of simple processing elements (PEs), *i.e.* neurons, interconnected such that information can flow among the PEs. The strength of the connection between two PEs, which is due to the biochemical state of the synapse in biological neurons, is determined by the connection weight, or simply the weight, in CNNs. Thus, learning in CNNs is any procedure for altering the weights in response to the input of new information. Artificial neural nets are usually distinguished by their network paradigms, which specify the nature of the PEs, the network architecture or topology, *i.e.* the pattern and types of interconnections among the PEs, and the learning rule which allows the network to adapt to new information. Although many network paradigms exist,[25] the focus here will be on multilayer, feedforward networks that are by far the most prevalent type of network in general use today as well as in chemical applications (*vide supra*).

Figure 1 depicts the network architecture of a typical three-layer feedforward net (*N.B.*, the input layer is sometimes not counted as it merely passes on the input values to the next layer in the network). Feedforward networks without hidden layers are generally called *perceptrons* and thus, feedforward networks with hidden layers are sometimes called *generalized perceptrons*, and that is the terminology that

will be used in the remainder of this work. While many problems can be handled by perceptrons, some extremely simple ones cannot. The classic example, as pointed out by Minsky and Papert,[36] is the inability of perceptrons to solve the simple XOR problem. It is usually assumed that this inability of perceptrons to solve the XOR problem was primarily responsible for the decline in popularity in the 1970s of neural network methods. Although generalized perceptrons can easily solve this problem, a satisfactory procedure for training them was not available until the landmark paper by Rumelhart, Hinton and Williams,[37] which was published in 1986. In that paper, the authors describe the now famous backpropagation-of-errors learning rule (*vide infra*), which helped revitalize neural net research. Interestingly, Werbos, in his 1974 Ph.D. thesis, essentially described the backpropagation learning rule as a technique for function optimization, but due to the limited distribution of his work, it was not recognized until much later.[38]

As is seen from Figure 1, each layer of the network in a generalized perceptron consists of a set of nodes, indicated by circles, joined by weighted (w_i), uni-directional connections indicated by arrows. There are no connections between nodes within a given layer in this type of network architecture. The values of the input variables, x_1 and x_2, are "passed through" the input nodes without change. The nodes in all other layers are called processing elements (PE) and carry out both a summation of the

incoming signals, denoted by "Σ," and an evaluation of the resultant sum by a non-linear transfer function, denoted by "*f*." The detailed form of these functions is shown to the right of the figure. In addition to the weighted inputs summed by each PE, an additional threshold or bias term, denoted by Θ_i, is also added to the sum. The final output, $R(x_1,x_2)$, is then obtained by an appropriate scaling of the value of the transfer function, S_k^{out}, in the output layer. Scaling is required as the output range of most transfer functions is given by [0,1] or [-1,1], as seen for the hyperbolic tangent transfer function, *tanh*, depicted in Figure 2.

The ability of CNNs to learn, *i.e.* adapt to new input data, distinguishes them from most other computational paradigms. Learning in CNNs consists of appropriately modifying the network weights in response to a given set of inputs and can be classified as either *supervised* or *unsupervised*. In the former case, learning, or *training*, is based upon a direct comparison of the network's output with its input, where the outputs represent "correct" network responses. In the latter case, the learning goal is not generally specified in terms of correct network responses, rather the network is expected to use correlations among the input data to create categories which reflect these interrelationships. In either case, memory is embodied in the set of weights which defines the network. Thus, there is a clear distinction between memory in CNNs compared to memory in

typical digital computers, even those which support highly-parallel computation.

Generalized perceptrons are trained by supervised learning methods. Currently the most popular training procedure is backpropagation. In this approach, an error function is defined that determines, for the given collection of weights, the difference between the desired network output and the output produced by the net for a set of inputs, *i.e* the training set.[39] During the learning process the weights are incrementally adjusted until the error function reaches a minimum, which is usually a local minimum. Finding the global minimum by such a procedure is very difficult and is generally not possible except in very special circumstances. The difficulty is a result of the fact that backpropagation training, like other gradient descent methods, makes changes in the weights proportional to the gradient, or first derivative of the error function. The more the error function decreases, the larger the gradient becomes and consequently, the greater the weight changes. These weight changes can only be made in the direction that leads towards lower error. Once the error function is in a valley on the error surface, the backpropagation algorithm will find the bottom of the valley, but it cannot climb over a ridge to a lower valley. The weight adjustment is carried out sequentially beginning with those weights closest to the output nodes. The error correction or adjustment of the weights is then "propagated," layer by layer, back towards the input

layer. Thus, the input signals are propagated forward and the "error signals" are propagated back towards the inputs, hence the use of the terminology backpropagation (BP) learning to describe the training process. Other generalized perceptron learning procedures that avoid some of the problems that backpropagation has in getting trapped in local minima have been implemented, but backpropagation remains the most widely used method to date.[40]

Numerous neural network architectures support unsupervised learning, and the number of applications in chemistry based upon these networks is increasing steadily. However, since only supervised learning methods were used to train the neural networks used in this research, further discussion of these types of unsupervised CNN methods will not be done here. There are, however, a number of excellent detailed treatments of these methods available.[22-26,41]

Generalized perceptrons can be viewed in a number of ways. Two of the most useful are as devices that can implement either discriminant functions or generalized mappings. The former is important in classification problems, while the latter is important in property prediction and in modeling both static and dynamic systems, to name just a few. Lippmann has presented a very clear discussion of generalized perceptrons from the perspective of discriminant functions.[26] He shows that the presence of hidden layers in generalized perceptrons allows them to

construct boundaries of great complexity, a feature that is missing in perceptrons, which can only construct linear discriminant functions. In fact, it is this latter limitation which prevents perceptrons from solving the XOR problem.

Perhaps the most powerful feature of generalized perceptrons is their ability to construct very general mappings between sets of input-output pairs. In fact, generalized perceptrons can construct very complex, non-linear mappings of arbitrary accuracy for all reasonably well-behaved functions.[24,27,42] These mappings can be classified as either *autoassociative*, if the input-output pairs are identical, or *heteroassociative*, if they differ. Networks with these types of behavior are called autoassociative and heteroassociative, respectively. Since, in general, one wants to predict something different than the starting materials in a chemical synthesis problem, the CNNs employed in this work were of the heteroassociative type.

In addition to the features noted above, CNNs possess a number of other features that make them potentially suitable for a variety of tasks that are difficult to address with more traditional computer architectures. These characteristics are listed in Table 1. It is appropriate to comment at this point on one of the distinguishing features of CNNs, namely their massively parallel architecture. The operation of a neural network is inherently parallel, with each PE processing its own inputs and outputs.

Table 1

Neural Network Characteristics

Advantages	Disadvantages
Massively Parallel	Slow To Train
Fault Tolerant	Can't Explain Predictions
Learn By Example	Non-algorithmic
Improve With Training	Local Minima Problem
Non-algorithmic	Overfitting/Overtraining
Associative Memory	Data Dependant
Content-Addressable Memory	Representation Dependent
Distributed Representation	Requires Vector Data Representation
Pattern Recognition	May Extract Incorrect Features
Pattern Classification	May Generalize in Valid but Undesired Ways
Non-Linear Mapping	Interpolate rather than Extrapolate
Feature Extraction	
Construct Rules From Data	
Generalization	
Model Free	
No assumptions about Data Distribution	
Global	
Arbitrary Complexity	
Adaptive	
Non-parametric	
Noise Tolerant	

However, in most work to date, including all of the research reported here, the CNN was simulated on a conventional computer with a single, sequential processor. When the appropriate network design has been worked out for a problem, then scaling up that solution by implementing the CNN on a multi-processor computer will be much easier than rewriting traditional computer programs to make them run on parallel computers.

Neural Network Simulators

The focus of this research effort was to understand and employ neural networks for solving organic chemistry problems. Therefore, commercially available neural network simulation programs were used to construct and train the neural nets instead of writing our own network simulator. There are a large number of neural network simulators available that cover the range from providing a single paradigm, like multi-layer generalized perceptrons trained by backpropagation, to others that incorporate more than 20 different paradigms. We used four different network simulators for this work. The earliest studies, those reported in Chapters III and IV on prediction of reactions, were done using the backpropagation trained generalized perceptron paradigm of the ANSIM[43] program from SAIC, San Diego, CA. ANSIM runs under the Microsoft Windows[44] program and has a very nice graphics interface. It has 13 different network paradigms, but SAIC stopped selling the ANSIM program in 1990. ANSIM also took a fairly long time (1 - 16 hours) to train large networks on the IBM[45] PS/2 Model 70 80386/80387 microcomputer on which it was run. One of the studies on representation reported in Chapter III employed the N-Net 210 program[46] from Ai-Ware, Inc, Cleveland. OH, which was run on a 80386/80387 IBM PS/2 Model 70 microcomputer. N-Net contains an unsupervised CNN for clustering as

well as backpropagation and functional link networks.[47] A functional link net is a variation of the perceptron where the input layer is enhanced with the crossproducts of the inputs. It is a feed-forward network trained using the delta rule (*vide infra*). The next phase of the studies used a backpropagation only simulator called Nets,[48] that was developed at the NASA Johnson Space Center and is commercially available through the COSMIC group at the University of Georgia. Nets comes compiled and ready to run on a microcomputer but also includes source code and instructions for compiling on many other computers. The Nets program was used for the ^{17}O NMR study in Chapter I and several of the alternative representations in Chapter III. Nets was run on a 16.8 MIPS Sun Microsystems[49] Sparcstation 1+. A comparison of the training times for the same problem between Nets on the 80386 microcomputer and Nets on the Sparcstation showed that the microcomputer took about 15 minutes to train the network and the Sparcstation took about 1 minute. More recent work, the response surface study in Chapter II, was done using the NeuralWorks Professional II Plus (NW2P)[50] simulator on the Sparcstation. NW2P includes 22 network paradigms, as well as a number of analysis and preprocessing tools.

Example of Training a Computational Neural Network

An example of how a CNN was designed and trained to solve a chemistry problem will serve to clarify some of the points made in the previous section. Hall and Kier[51] proposed a new index derived from chemical graph theory, which they called the Electrotopological state index (E-state index), for describing the electronic character and the topological environment of atoms in organic molecules. An intrinsic value I is defined for each atom as $I = (\delta^v - 1)/\delta$, where δ^v is the number of valence electrons and δ is the number of σ electrons. Each atom is considered to reside in the field ΔI produced by the sum of the effects of every other atom in the molecule, modulated by the square of the distance, in number of bonds, between the atoms. The E-state index for an atom is then $I + \Delta I$. Hall and Kier found that the ^{17}O NMR shifts of some simple ethers, aldehydes and ketones (shown in Figure 3) were well correlated with the E-state index of the oxygen atoms. Hall and Kier developed two linear regression equations that related their E-state index to the ^{17}O NMR shift δ^{NMR}, one with a positive slope for the ten ethers:

$$\delta^{NMR} = 92.5 \text{ (E-State)} - 441.6 \text{ , avg. error} = 2.3$$

and another, with the a negative slope for the nine carbonyl compounds,

$$\delta^{NMR} = -27.8 \text{ (E-State)} + 834.5 \text{ , avg error} = 2.82.$$

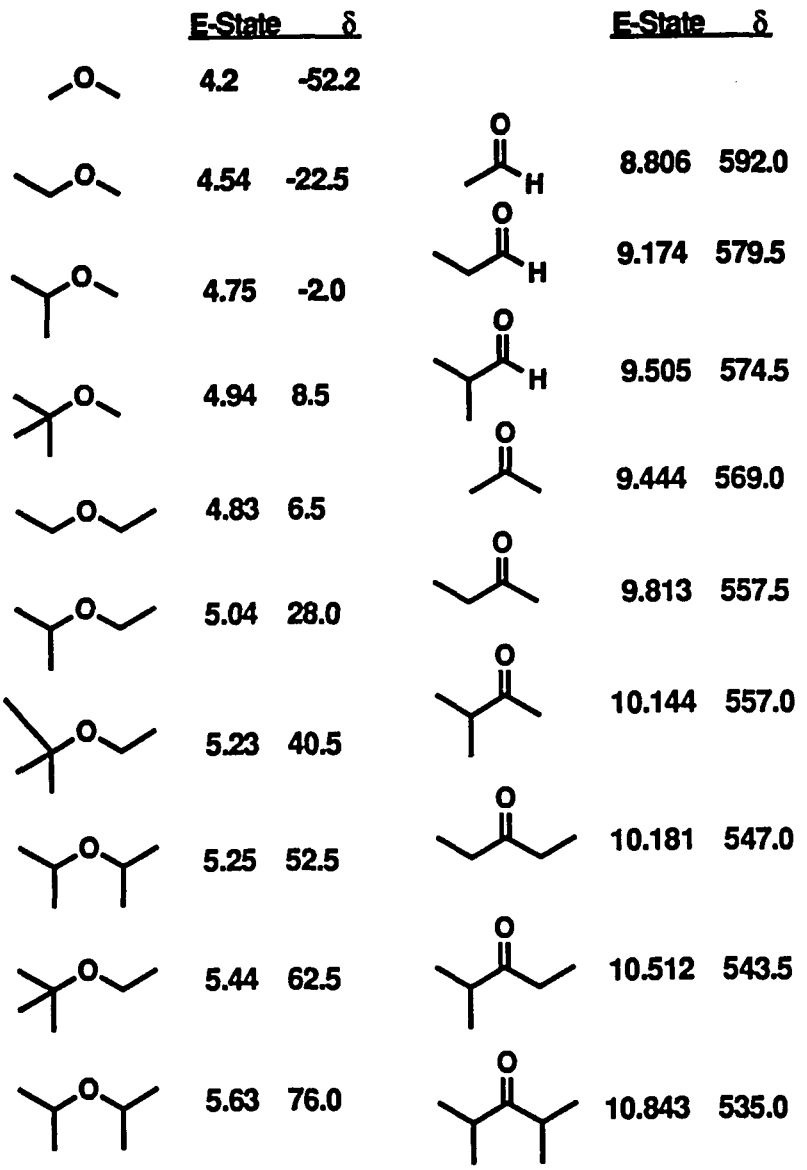


Figure 3. E-State-Index & ^{17}O NMR Shift for 19 Ethers and Carbonyl Compounds.

A generalized perceptron neural network was used to learn the relationship between the E-state index of the oxygen atoms in the 19 compounds shown in Figure 3. Instead of training a separate CNN for each class of compounds, the more global character of the CNN allows a single neural network to treat both classes. The size of the network is constrained by the problem and the amount of data that is available. In this case, there is one input, the E-state index of the oxygen atom, and one output, the δ^{NMR}. Choosing the number of hidden units is still done by trial and error. In general, one should use the smallest number of hidden units that will solve the problem and still give good generalization. (*N.B.* The number of hidden units and layers required will be addressed in the following section on Important Computational Neural Network Issues and also in Chapter II). There are two common approaches to choosing the number of hidden units: either by starting with one PE and adding others one at a time until the best solution is obtained, or by starting with the maximum number of PEs and gradually reducing the number of hidden units until the results are satisfactory. Here, we started with one hidden unit and added one at a time, producing the network with three hidden units shown in Figure 4.

Networks are conveniently designated by the training method and the number of layers and units. This network will thus be designated BP 1-3-1, to indicate that it was trained with the backpropagation algorithm and that it has three layers with one input, three hidden units and one

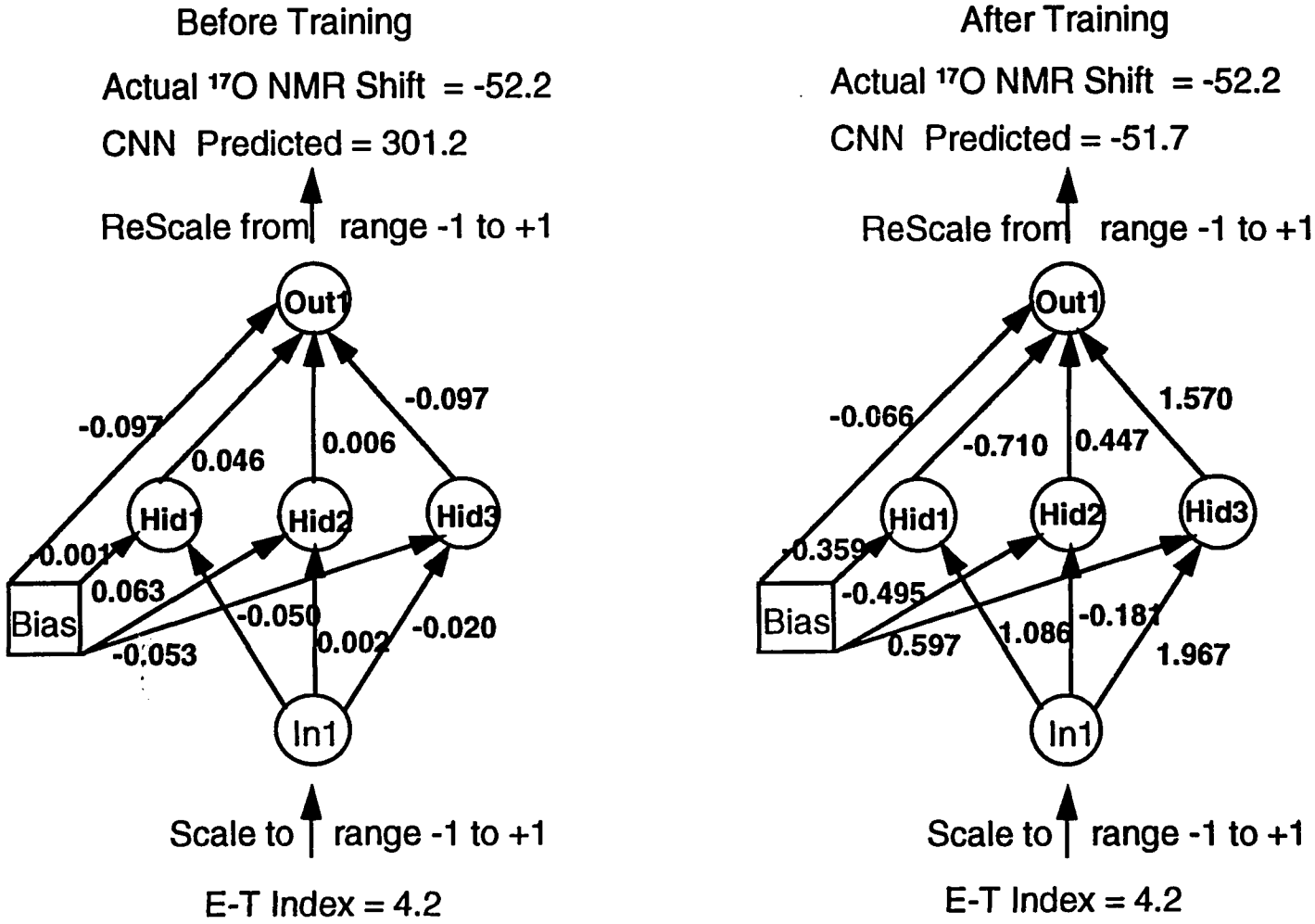


Figure 4. CNN Weights Before and After Training.

output. The number of weights or connections can be calculated as $1 \times 3 + 3 \times 1 = 6$ weights plus $3 + 1 = 4$ bias weights for a total of 10 weights. These weights correspond to the coefficients or parameters of the system that must be determined. There are 19 compounds in the training set, giving a ratio of 1.9 compounds per weight. If the exact form of the mapping between inputs and outputs were known it would be possible to set the connection weights to the correct values at the outset and the network training would not be needed. In most cases this mapping is not known explicitly and the weights must be found by the training procedure. Initially the weights are set randomly to small values near zero, as shown in the left-hand network in Figure 4. If all the weights were initially set to the same small value, there is no way for the training procedure to overcome the symmetry of the weights and the learning process would not converge because all of the weights would be adjusted by the same amount. Thus the weights are randomly chosen in order to avoid the problems caused by symmetry.[52]

In this example, the desired result is the ability of the network to fit the data better than the regression method. Ordinarily, part of the data would be reserved for testing the network and would not be used for training. The number of hidden units would be chosen on the basis of which number gave the best predictions on the test set. Training would thus be stopped at the point where the error on the test set reached a

minimum and began to get larger. Here the training was continued until the error over the training set did not improve.

The training process starts when the first input sample is presented to the network. In the example in Figure 4, it is the dimethyl ether, which has an E-state index of 4.2. The inputs are first scaled to the useful range of the transfer function, which is between 0 and 1 for the sigmoid transfer function used here. The *tanh* transfer function would require that the inputs be scaled between -1 and +1. The network calculates an output, based on its initial set of weights, which, when rescaled from the 0 to 1 range of the transfer function to the scale of the NMR shift data, gives a predicted value of 301.2. The desired output is the chemical shift of the dimethyl ether oxygen, -52.2 ppm, so the error, called delta, is equal to the (Desired Target Output - Actual Network Output) = -353.4.

The backpropagation training method uses what is called the delta rule to adjust the weight of a connection. The amount of the weight change Δw is a function of the size of the delta (T - R), the magnitude of the activation or output S^{out} of the PE on the input end of that connection and the learning rate η, which is a decimal fraction between 0 and 1. Thus, the delta rule, for changing the weights is given by

$$\Delta w = \eta(T - R)S^{out}$$

The delta rule is a steepest descent method that minimizes the sum of squares of the deltas by assigning the credit or blame[53] to units according

to their activation levels. The more active a unit is or the larger its error, delta, the more its weight gets changed. Unfortunately, the delta rule only works for the output units, because it is not possible to define the target value needed to calculate the delta for the hidden units in the same way that can be done for the output units. This limitation was circumvented with the development of the backpropagation method.[37] In the back-propagation method, the delta rule is extended to become the generalized delta rule, where the weight change for a PE is given by

$$\Delta w = \eta(delta)S^{out}.$$

The definition of delta differs for output units and for hidden units. For output units, $delta = (T - R)f'(net)$, where $f'(net)$ is the first derivative of the net input to that node. For hidden units, the delta is the weighted sum of the deltas of the units to which the unit has output connections, times the derivative of the transfer function. Thus, for hidden units

$$delta = \sum((delta) \times w) f'(net).$$

The generalized delta rule thus allows weight changes to be made to the hidden units.

The training process involves repeatedly presenting each input to the network, calculating the network output, calculating the deltas, and adjusting the weights until the root mean squared error over all the samples is minimized. The right hand picture of the network in Figure 4 shows the final set of weights in the trained network. To test the accuracy

of the set of weights thus obtained, each sample in turn is presented to the network and the resulting network calculated output is compared with the actual value. In this case, the CNN predicted value of -51.7 is in good agreement with the experimental value of -52.2 ppm. Figure 5 shows the initial predictions by the untrained network and the final predictions by the trained network. The lines in the figure are the least squares fit lines of the experimental chemical shift values. The power of the neural network can be seen from Figure 5 where a single neural network gave a very good fit to the chemical shifts in both sets of compounds (avg. error = 3.43), in contrast to the regression methods, which had to treat each class of compounds separately. One would expect that if there is chemical relevance to the E-state index that there would be a consistent, albeit complicated, relationship to a measurable parameter such as the ^{17}O NMR chemical shift.

Important Computational Neural Network Issues

In the following section the issues of data representation, network optimization, generalization and error analysis will be discussed. These issues can have a profound effect on the ability of a CNN to perform a desired mapping, and an understanding of them is required to intelligently apply this computational methodology to any problem, and to problems in chemistry in particular. It is perhaps unfortunate that these methods are

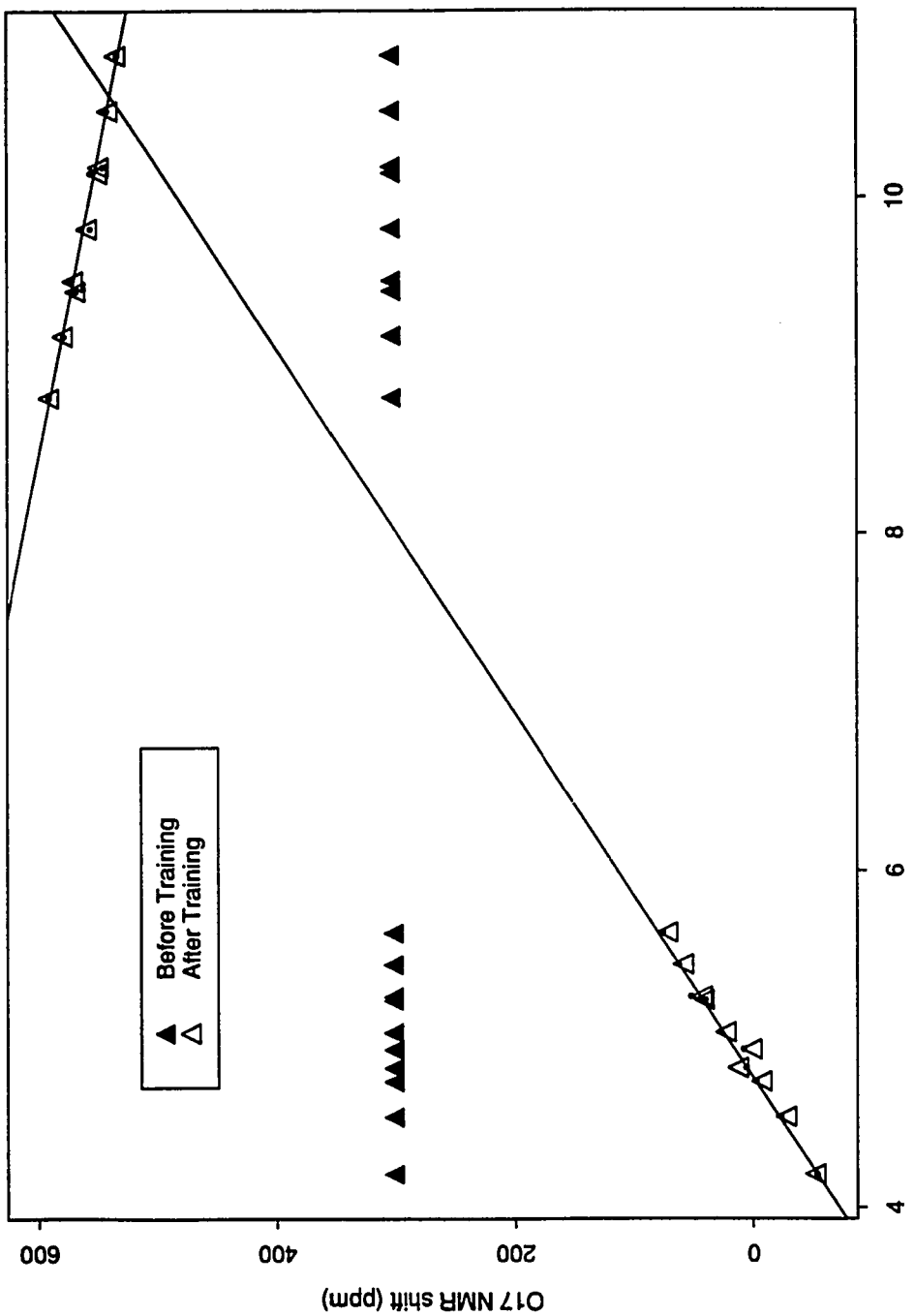


Figure 5. Plot of E-State *vs* ^{17}O NMR Shifts for Untrained & Trained Nets.

even called "neural networks." The reasons for that name derive from their origins in the field of cognitive psychology,[22] as previously discussed. The name neural networks is unfortunate because it elicits expectations of intelligence and the mysteriousness of brain function. These expectations of mysteriously intelligent behavior from artificial neural networks sometimes obscure our understanding of what their real capabilities and limitations are. It is hoped that the discussion that immediately follows and the experimental examination of some these issues in Chapter II will serve to demystify artificial neural networks and will provide insights into how they may be expected to perform on real problems.

Data Representation

Data representation is the most crucial part of any attempt to apply CNNs, especially to chemical problems. Generally, input to CNNs is in the form of a vector or list of individual components. These network inputs correspond to the set of independent variables or state variables, of the system. If an appropriately complete set of input variables is chosen, a solution to the problem is possible. If the set of input variables is insufficient to represent the system, there is little chance of any method giving a good solution. Although in many cases, such as for chemical process instrument measurements, a vector representation of the inputs is adequate, it does not capture the type of chemical information inherent in

2D or 3D molecular structures. We will present some fairly general solutions to the representation of chemical structures in Chapters III and IV, but can not claim to have found a universal solution to the representation problem. Some recent work by Kvasnicka[16,54] has addressed the representation of 2D molecular structures in an interesting but limited way, by using oriented chemical graphs as network inputs.

Given that vector-like input is required, the problem becomes one of choosing a set of appropriate feature descriptors. On the one hand, these descriptors should be as "global" as possible, that is, they should be appropriate to and obtainable for a broad range of compounds. On the other hand, the set of descriptors should be as small as possible to insure that the problem of small sample size, which occurs in many chemical studies, does not become limiting. As noted above, the number of nodes, including input nodes, determines the number of interconnections and thus, weights that must be determined during training. It is generally believed that the number of data samples should be at least three times the number of weights,[55] although even a factor of three would be considered insufficient in certain circumstances.[27] Inter-relationships among the weights can, however, lower the number of independent weights and thus, effectively increase the samples-to-weights ratio.

Another problem which arises in data representation is that of linear and non-linear correlations among descriptors. When this is the case,

several descriptors characterize essentially the same information. Linear correlations can be pinpointed with standard statistical correlation methods. Non-linear correlations, however, are more difficult to uncover.[56] A recent paper by Kramer[57] presents a detailed discussion of this problem, and describes a novel CNN solution based upon data encoding. Other related approaches have also been described in the literature.[58,59] In Chapter II, we show that correlated descriptors may not be such a serious problem for neural networks.

Network Optimization

Network optimization represents an important aspect of CNNs that must be addressed, and includes considerations of the optimal number of PEs or nodes, the nature of the error function,[60] and the nature of the transfer function. Once a particular CNN paradigm is chosen--generalized perceptrons in the present work--one of the most difficult challenges to the development of CNN applications is the determination of an appropriate network topology, *i.e.* the number of nodes and interconnections. As each node has a significant impact on the number of interconnections, and hence the number of weights that must be determined, the goal generally is to obtain the smallest net consistent with the complexity of the data. Although a number of practical schemes exist for dealing with the

important problem of optimal network architectures,[61,62,63] rigorously convergent procedures do not currently exist.

The error function is generally taken to be a sum-of-squares error function, $E = \frac{1}{2} \Sigma_i (R_i^{obs} - R_i^{net})^2$, where R_i^{obs} is the ith observed or desired value of the response function and R_i^{net} is the corresponding value produced by the CNN. In some cases the average sum-of-squares error function, $E_{ave} = (1/N)\Sigma_i (R_i^{obs} - R_i^{net})^2$ is sometimes used.[27] As the number of samples in the training set, N, increases, E_{ave} approaches the expectation value of the sum-of-squares error, which shows its connection to the stochastic approximation employed in many statistical optimization procedures.[27]

The form of the transfer function also is important in mapping applications of CNNs. In most applications sigmoid or *tanh* transfer functions (see Figure 2) are used. Lapedes and Farber[42] have presented a very clear discussion of the function mapping characteristics of such differentiable "step functions". From their discussion it is clear that while these functions may not be optimal as basis functions to represent many input-output mappings, they can, given a sufficient number of hidden nodes,[29-33] provide an adequate basis for describing a considerable variety of input-output mappings. Another aspect of sigmoid or *tanh* transfer functions is their "sharpness," which is controlled by the gain parameter, α. As seen in Figure 2, as $\alpha \rightarrow \infty$ both functions approach step functions,

while as $\alpha \rightarrow 0$ both functions approach straight lines. Thus, it is expected that for most applications requiring relatively "smooth" mappings α ought to lie in a range such that the transfer function is neither too steep nor too flat (*vide infra*).

Optimization of the weights is generally carried out by some form of backpropagation of errors procedure.[37,38] This is a gradient-based procedure and is thus beset, as are all gradient-based procedures, by the multiple minimum problem. Hence, the likelihood of becoming "trapped" within a local minimum is high, and can make obtaining a proper solution difficult. A robust modification of this algorithm, called extended delta-bar-delta (EDBD), has been developed that holds considerable promise.[64]

Several stochastic learning algorithms are also worthy of note. These can best be understood if one considers the set of weights, $\{w_i\}$, as components of a weight vector, $w = (w_1, w_2, \ldots, w_n)^T$, where T is the transpose. Determination of the weights is then a problem of searching weight space to find a w that minimizes E. Various forms of simplex,[65] Monte Carlo,[66] simulated annealing,[67] and genetic[68] algorithms have all been investigated. Andrea and Kalayeh also used an ordinary differential equation solver (ODE) to optimize the weights.[55] While these methods appear to yield well converged solutions, they generally take longer than BP or EDBD procedures. Recent work by Chen and Hecht-Nielsen[69] has, however, shown that due to permutational and sign

symmetries of the weight vector, the weight space is highly redundant, and many weight vectors yield equally good solutions. A practical importance of this to the training of CNNs is that only a relatively small "cone" in weight space need be considered, but this requires that the constraint boundaries of the cone be known. Unfortunately, the means for determining the constraint boundaries is not yet available. Nevertheless, it does indicate that weight space is not as vast as it might appear at first view and thus, future work in this area may point the way to more powerful new algorithms that will facilitate the rapid determination of well-converged weight vectors.

Generalization

One of the most important attributes of CNNs is their ability to *generalize*, that is, their ability to make reliable predictions on new data with similar accuracy to that obtained with training data. The problem of generalization is related to the problem of *overfitting*. Overfitting occurs when the size of a training dataset is comparable to the size of the weight space that "supports" the CNN. As the number of hidden layers and their associated nodes directly influences the number of weights, the complexity of a given network is limited by the size of the dataset. Under such circumstances, the training data including noise may be fit nearly exactly, but the CNN most likely will fail on new data. Thus, generalization is best

when noise is smoothed out, a situation which can be approached by obtaining more new data, by smoothing the data through averaging, and by limiting the size of the network. Examples of the latter two of these approaches are given in Chapter II.

A characteristic of CNNs that affects their ability to generalize is their propensity to interpolate rather than extrapolate.[70] An example of this behavior is shown in Chapter III, in the prediction of boiling points, and in the reaction prediction studies in Chapter IV. The best solution is to avoid requiring the CNN to extrapolate by including the broadest possible range of output values in the training set. Ideally, the training set should include examples at the boundaries of the "chemical space" that is being mapped by the neural network. If that is not possible then one must be aware that CNN predictions far beyond the training examples will have an error that shifts the result closer to the most similar training case.

Error Analysis

A critical part of the development of any CNN is an evaluation of its performance. As generalized perceptrons are based upon supervised learning, the data set is generally divided into training and test sets, which can be obtained by appropriate random-sampling procedures if the amount of data is sufficiently large.

An ideal test set is one that spans the problem space adequately to ensure that a network which performs correctly on a test set can be considered to have "solved" the ultimate problem under study.[24] In some situations, an "acceptance test set" is kept in reserve for final network validation, but this is not feasible when the amount of data available is small.

In many cases related to chemical systems, however, the presence of small data sets requires that resampling methods must be employed. The particular method chosen will depend upon the nature of the problem and the amount and type of data available. The majority of the commercially available neural network simulator software does not contain provisions for automatically applying the resampling methods, although some have a batch mode or scripting capability that allows multiple networks to be trained simultaneously. This omission greatly increases the time required to produce a validated, robust neural network method. Three resampling methods are mainly in use today, *viz.* cross-validation, leave-one-out, a variant of cross-validation, and bootstrapping. A recent book by Weiss and Kulikowski[71] provides an excellent, readable discussion of these methods and their application to error estimation in generalized perceptrons as well as in other learning-based procedures. A more mathematically-oriented treatment is provided by Efron.[72,73] An alternative approach to the small dataset problem has been dealt with by Stubbs,[74] who used a

Bayesian prediction scheme to augment the initial dataset. While this approach may not be appropriate in all cases, it certainly merits further consideration as a possible strategy for dealing with the small datasets.

The particular form of the "error function" is also important. Should all errors be treated equally? Some types of errors are more important than others. In some cases false negatives may be considerably more serious than false positives as in, for example, the diagnosis of a particular medical condition, while in other cases both types of errors may be equally serious or not very serious at all. This suggests that some weighting of the errors may be desirable or required, but determining appropriate weights can be difficult. Risk and cost functions have been used in the past, especially in Bayesian predictions,[75] but this has not generally been done in CNN studies. Nevertheless, the possibility of using them should not be precluded, and their application in specific cases may be proper and necessary.

Survey of Chemical Applications of Neural Networks

Although neural network applications have been growing at a rapid rate in general, applications in chemistry have been lagging somewhat behind as chemists learn how to apply this "new" technology. A search of The American Chemical Society's Chemical Abstracts (CA) database[76] on STN International, from 1967 through September 1992 found 312 papers

on CNNs. An additional 13 chemical applications of CNNs were found in the INSPEC engineering database,[77] five applications were found in the COMPUSCIENCE database[78] on STN, and another nine papers were found by examining the current literature directly. This survey of 339 papers is undoubtedly incomplete as new works are published weekly. Figure 6 shows a plot of the number of chemical applications of CNNs as a function of year. From the figure, it is clear that more than half the applications have occurred within approximately the last two years (206 papers or 60%), and three-fourths occurred within the last 3 years (266 papers or 78%). The earliest papers systematically applying neural networks in chemistry appear to be those of Jurs, Kowalski, Isenhour, and Reilly[79] at the University of Washington in 1969, when the first of a long series of papers appeared in the journal *Analytical Chemistry* under the series heading "Computerized Learning Machines Applied to Chemical Problems." Two earlier papers appeared in 1964 and 1966, but they were of a more theoretical than practical nature. Ironically, 1969 was also the year that Minsky and Papert's book *Perceptrons*[36] appeared. The small blip on the graph for 1969-1973 represents the enticement and subsequent shortcomings of linear learning machines[80] (perceptrons). A large rise in the number of papers followed publication of the backpropagation learning algorithm[37] in 1986, which revived the field and led to the current

resurgence in the development and application of CNNs in many fields, including chemistry.

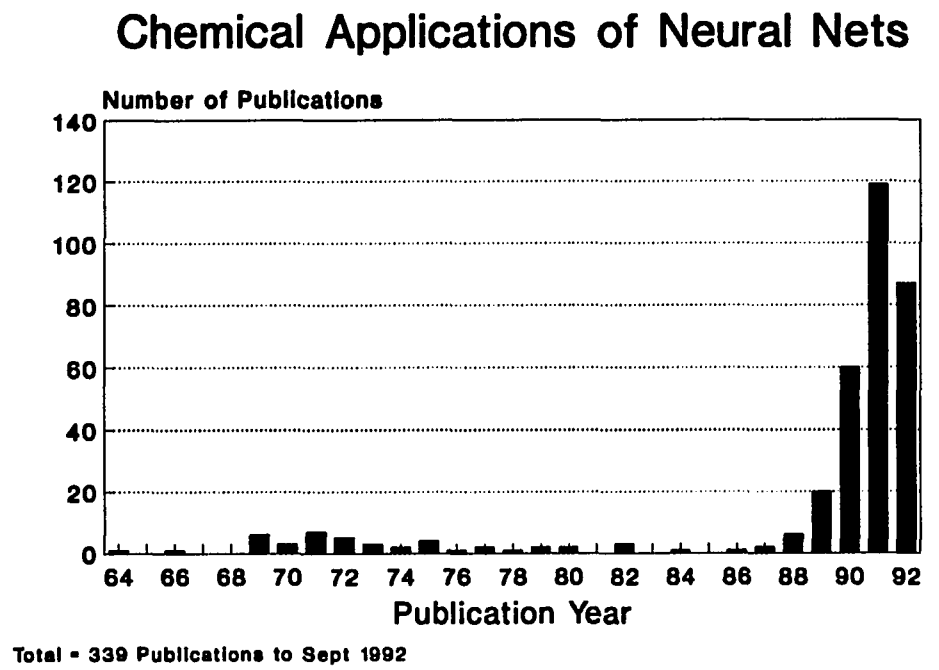


Figure 6. Histogram Showing the Number of Chemical Applications of CNN Papers Published per Year From 1964 to the Present.

Neural network applications in chemistry can be divided into roughly ten categories, as shown in Table 2. Although these ten categories are somewhat overlapping, they do reflect different areas of applications. In the following, each category, except for reactivity prediction, which was covered at the beginning of the introductory section, will be described

Table 2

Types of Published Neural Network Applications in Chemistry

Category	Total[a] # of Papers	Network Paradigm				
		Feed Forward[b]	Perceptron Linear Learn Machine	Kohonen[c] SOM LVQ CPN	Hopfield Associative Memory	Unspecified[d] & Others
Reviews & Miscellaneous	48	13	9	1	2	17
Process Control & Chemical Eng.	69	31	0	2	0	29
Analytical Chemistry	51	17	10	2	3	18
Protein Structure & Analysis	38	17	8	3	5	7
Biomolecular Informatics	22	6	6	1	0	6
QSAR & Pattern Recognition	31	12	11	3	0	4
Spectra-Structure Correlation	30	17	11	0	1	1
Property & Parameter Prediction	10	6	1	1	0	3
Reaction Prediction	7	5	0	0	1	1
Nuclear Chemistry	33	15	0	4	2	12
TOTALS	339	139	56	17	14	98

Notes. a. Total methods may exceed total papers. Several papers used more than one method.
b. Includes all multi-layer feed forward networks. Most were trained with backpropagation.
c. CounterPropagation (CPN)(5), Kohonen (3).
d. Others are genetic algorithms (2), ART (1), reinforcement learning (1), dynamic capacity allocating (DCA) (2).

briefly, emphasizing the types of applications, trends, important new methods, and the neural-network paradigms used. Four of the publications are actually dissertations, starting with (1) Jurs[81] in 1969, who studied linear learning machines for mass spectral interpretation at the University of Washington, and after a long hiatus, three chemical engineering dissertations in 1991: (2) S. Roat's (U. Tennessee) application of neural networks for nonlinear optimal control of chemical processes,[82] (3) D. Haesloop's (U. Washington) system identification and control using neural networks,[83] and (4) N. Bhat's (U. Maryland) use of backpropagation networks for control of dynamic chemical processes.[84]

Reviews, Theoretical Studies, and Miscellaneous Applications

Forty-eight papers are found in this category. Two reviews are of particular interest. Zupan and Gasteiger[70] provide an excellent overview of CNN paradigms and describe 34 chemical applications; Lacy[85] provides a brief review as an introduction to a *Symposium in Print* that he edited. Kyuma[86] reviews optical neural networks, and suggests how holographic techniques may be used to produce associative memory devices. Schmuller suggests possible CNN applications in environmental chemistry.[87] Several papers detail how neural networks can be used to implement both linear[88] and nonlinear[89] partial least squares and also nonlinear principal components analysis.[90] The advantage of neural networks in these

methods is their ability to model nonlinear relationships while attaining robust generalization properties. A novel method for the display of multivariate physicochemical properties of biologically active molecules was reported by Livingstone, Hesketh, and Clayworth[91] in the UK. They used an auto-associative backpropagation network to reduce multidimensional data to two dimensions, thus enhancing the visualization of relationships within the data. This type of application will likely become more important in the future as chemists discover the potential of neural networks for clustering and dimensionality reduction of high-dimensional chemical data.

Chemical Process Control and Chemical Engineering Applications

The largest group of applications of CNNs in chemistry is in the area of process control and chemical engineering, where there are 69 papers. Chemists and chemical engineers who deal with chemical and biochemical processes are used to describing and understanding those nonlinear processes in terms of mathematical models. The majority of the neural networks used were multilayer, feedforward nets, trained with some form of backpropagation since many of the applications in this area are concerned with real-time diagnostics and predictive control for nonlinear dynamic chemical processes. The earliest application of CNNs in chemical engineering was in 1988, where Hoskins and Himmelblau[92] gave an outline and a simple example of how chemical engineering knowledge can be

represented in an CNN. A recent issue of *Computers in Chemical Engineering*[93] is devoted to reviews of the use of neural networks for process modeling, process fault diagnosis, process variable estimation, and process control. Bugmann, Lister and Von Stockar[94] found that neural networks were useful for characterizing bioreactor processes where it was difficult to explicitly model desired properties from the measurements that were able to be made. They found that the CNN could approximate unknown functional relationships from suitable examples without needing to specify the form of the mapping. Expert systems and knowledge-based control systems are finding greater utility in process fault diagnosis and control,[95,96] but there is a bottleneck in the rate at which process knowledge can be acquired. The problem is compounded because of frequently changing process conditions, which require constant updating of the process knowledge. In this area, CNNs seem to be useful because they can extract knowledge patterns directly from plant or process operations data and can be easily updated as new operations data is obtained.

Analytical Chemistry Applications

Analytical chemistry is another area with a large number of CNN papers. Of the 51 papers in this category, the early papers mainly employed perceptrons or linear learning machines, while more recent publications generally employ generalized perceptrons trained by back-

propagation. Roughly one third (16 of 51) of the papers describe the application of CNNs for calibrating and deconvoluting spectrometer data, with the work of Glick and Hieftje being typical.[97] Another area of concentration where neural networks have seen application in analytical chemistry is in the detection, quantitation, and identification of odors and vapors from semiconductor gas sensors where there are 11 papers, examples of which are found in the work of Chang *et al.*[98] and Hoffheins.[99] The rest of the analytical applications vary from geology,[100] predicting the water content of cheese,[101] pattern recognition of chromatographic data,[102] and attempts to develop intelligent analytical instruments.[103] A recent paper by Otto, George, Schierle, and Wegscheider[104] suggests combining fuzzy logic, as a model of an analytical chemist's reasoning process, with neural networks for automatic knowledge acquisition. The goal of this integrated approach is to develop intelligent systems for the automated qualitative analysis of spectroscopic data.

Protein and Polymer Structure and Analysis Applications

Thirty-eight papers were found in this category; the majority are attempts to predict the secondary structure of proteins. Stolorz, Lapedes and Xia[105] found that an CNN did no better than other statistical methods, and concluded that the primary amino acid sequence of the protein does not contain sufficient information to improve on the secondary structure

predictions made by statistical methods. Muskal and Kim[106] had more success in predicting secondary structure when they linked two generalized perceptrons networks in series and used amino-acid composition and other data for inputs. The tertiary structure of proteins has been approached by Wolynes and coworkers[107] employing associative memory Hamiltonians to recognize the folded tertiary structure of proteins. Ferran and Ferrara[108] clustered proteins into functional families using a Kohonen self-organizing map.[109] Three of the papers included in this group describe CNN methods for mapping the potential-energy surfaces of synthetic polymers, but may have applicability for proteins as well.[110] A paper by Bohm[111] suggests that CNNs may help overcome some of the lack of knowledge about determinants of protein folding by extracting relevant information from known protein structures. The application of CNNs for protein analysis and structure prediction was recently reviewed by Hirst and Sternberg.[112]

Biomolecular Informatics Applications

Related to, but distinct from, the preceding category is the growing area of biomolecular informatics. This field is concerned with finding patterns in biological sequences and relating those sequence patterns or motifs to their biological function. The review by Hirst and Sternberg[112] analyzed nine of the 22 publications in this area. The majority of the papers focused on using perceptrons for finding promoter sites or binding

sites in DNA.[113] Some of the earliest work on finding translation initiation sites in *E. Coli* was done by Stormo and collaborators[114] in 1982. Recently Arrigo, Giuliano, Scalia, Rapallo, and Damiani[115] used a Kohonen self-organizing map[109] to identify a new sequence motif in the human insulin receptor gene. Four of the papers deal with protein sequences, with the most recent publication by a group in Hungary.[116] Claverie and Sauvaget have developed a portable software package,[117] WOBB.C, for implementing the perceptron paradigm for defining and recognizing ambiguous sequence motifs in either protein or nucleic acid sequences.

Quantitative Structure-Activity Relationships and Pattern Recognition Applications

Quantitative structure-activity relationships (QSAR) have been developed using CNNs to map sets of chemical-structure descriptors to activities in biological systems. The 31 papers found in this area show that the use of CNNs is gaining acceptance as a reasonable alternative to statistically developed models, although it does not appear to offer any significant advantages over the latter models. Several authors,[20] including Andrea[55] and Aoyama[118] have analyzed a number of issues related to the size of data sets and the reliability of the QSARs derived from CNNs. The internal models constructed by the CNNs can be interpreted as mappings or discriminants built up of linear combinations of the transfer functions

which can be considered as basis functions. Nearly all of the neural networks used in QSAR applications were of the feed-forward type, trained by backpropagation. Examples include pattern classification of flow cytometer data to identify algae,[119] prediction of biodegradation of organic benzene derivatives,[120] development of a QSAR for 256 dihydrofolate reductase inhibitors,[55] structure-odor relationships of nitrobenezenic musks,[121] and Hussain's[122] method for formulation optimization in pharmaceutical product development. Rose, Croall, and MacFie[123] used an unsupervised learning method, Kohonen SOM[109] topology-preserving mapping, to reduce a multidimensional matrix of physicochemical property data for some antifilarial compounds to a two dimensional representation. In this case principal components analysis failed to give interpretable clusters, presumably because of nonlinear features in the dataset.

Spectra-Structure Correlation Applications

Two different types of applications have employed neural networks for the correlation of spectra and chemical structure. In 28 of the 30 papers published to date in this area the input to the network has been spectral features and the desired output was the class of compound or structural fragments present in the molecule that produced that spectrum. The two remaining papers involved spectral prediction based upon the structural fragments or molecular environments present in the molecules. All but one

of the papers used backpropagation or linear learning machines, which was an early type of perceptron with a linear transfer function. The group of 28 papers that used spectra as input can be further subdivided according to the type of spectra: ten dealt with infrared (IR) spectra, six with nuclear magnetic resonance (NMR) spectra, one with ultraviolet (UV) spectra, and eleven with mass spectra (MS). Munk, Madison, and Robb[124] reported that a feedforward net was better than linear regression for identifying functional groups present in organic compounds from their IR spectra. Borggaard and Thodberg[125] have described the OMNIS program (optimal minimal neural interpretation of spectra) wherein they use principal components analysis as a preprocessing method, and then employed cross-validation as a guide to removing network connections until a minimal network that gives good generalization is obtained. An interesting paper by Kjaer and Poulsen[126] describes a feedforward network that identifies cross peaks in 2D COSY ^{1}H NMR spectra. Curry and Rumelhart[127] used a feedforward, multi-layer net to classify mass spectra according to which of 100 functional groups were present. Otto and Hoerchner[128] used a hybrid of fuzzy logic and an adaptive bidirectional associative memory type of CNN, to identify UV spectra of organic compounds by finding the nearest match between the input spectrum and a set of stored spectra. Anker and Jurs[129] used a feedforward, multi-layer net to accurately predict the ^{13}C NMR shifts of 431 keto-steroids from calculated structural descriptors.

Kvasnicka[130] also described the prediction of ^{13}C NMR shifts with a multi-layer feedforward network, but used oriented graphs as inputs. The network Kvasnicka described was structured so that it resembled the chemical structure graphs used as inputs. This is a novel approach for inputting chemical structural information, but it is difficult to extend the procedure to the general case.

Property Prediction and Parameter Estimation Applications

Compared to the other areas of chemistry, there are few applications of neural networks for property prediction and parameter estimation. Nine of the ten papers in this area were published in 1991 or 1992, indicating that this type of CNN application may be more prevalent in the future. Kito, Hattori, and Murakami[131] used a feedforward network to predict the acid strength of mixed metal oxides. Bodor and collaborators have published two studies on the prediction of properties of organic molecules using multi-layer feedforward networks, the first predicted the water solubility of a diverse set of compounds,[132] while the second predicted oxidation potentials of heterocyclic compounds.[133] Peterson[134] used a Counterpropagation Network[135] (CPN) to predict Kovats indices for substituted phenols with lower errors than were obtained using linear regression. The thin-layer chromatographic behavior of 22 benzoic acid derivatives was predicted using a feedforward neural net by Glen *et al.*.[136]

Noid, Varma-Nair, Wunderlich, and Darsey[137] inverted the usual CNN prediction process by training a multi-layer feedforward network on heat capacities of polymers. The network was then used to estimate the two parameters of the Tarasov function, which is commonly used to predict heat capacities of polymers. The accuracy obtained using the CNN estimated parameters was significantly better than that obtained by other methods.

Nuclear Chemistry and Nuclear Power Applications

The 33 applications in nuclear chemistry and nuclear power plants center on two main themes, monitoring and analyzing the status of numerous sensors in nuclear reactors and detecting and identifying sub-atomic particles from decays tracks of charged particles, with most of the authors reporting the use of feedforward networks. Guo and Uhrig[138] used a hybrid neural network which has potential applicability in other areas as well. Their hybrid net used a Kohonen SOM[109] to cluster nuclear power plant heat output data. Then the centroids of these clusters were used to train a feedforward net that could predict the rate of heat production with an accuracy of 0.1%. Opposing conclusions about the utility of CNNs for analyzing nuclear decay events are given by Cherubini & Odorico[139] who found that a Kohonen Learning Vector Quantization[109] (LVQ) network performed worse than statistical methods for identifying quark decay products, while Stimpfl-Abele[140] found that feedforward networks gave

excellent results for recognizing decays of charged tracks. Peterson[141] used CPN[135] networks to classify energy levels in Curium II and Plutonium I.

Overview

Chapter II contains the results and subsequent discussion of the application of computational neural networks to a model problem, a three dimensional response surface. An understanding of the capabilities and limitations of CNNs is sometimes obscured by the lack of well characterized sets of chemical data. In order to separate the effects of the neural network's inherent capabilities from the effects of the data itself, a model of a response surface, that retains many of the features of real problems in predictive chemistry was developed. Chapter III gives a detailed discussion of the issue of representation of chemical structures in a form suitable for input to neural networks. The successful application of the representation developed to predicting the products of electrophilic aromatic substitution reactions is presented. The chemical structure representation employed in Chapter III is further extended to other classes of reactions in a general way in Chapter IV. Chapter V discusses the results of this research and offers conclusions about the utility of neural networks for problems in chemistry. The prospects for neural networks for future chemical applications are proposed.

CHAPTER II

COMPUTATIONAL NEURAL NETWORKS: MODEL-FREE MAPPING DEVICES

Introduction

One of the fundamental principles in chemistry is that properties are a function of a compound's molecular structure. This relationship has been expressed simply by Munk[142] as:

$$properties = f(structure).$$

As Munk correctly points out, much of chemical research is devoted to determining the nature of the function *f* for a wide range of properties, with the goal of accurately predicting those properties solely from chemical structure. Another way of describing the relationship between structure and properties is shown in Figure 7. Figure 7 schematically depicts the relationship between the geometric and electronic structural features of molecules and their associated chemical properties or biological activities. The upper arrow represents the interaction of a molecular structure with its environment to produce a compound's physicochemical properties, activities and reactivities. Experimental science is concerned with determining the mapping implied by the upper arrow. The pathway

QSAR/QSPR as a Mapping Problem

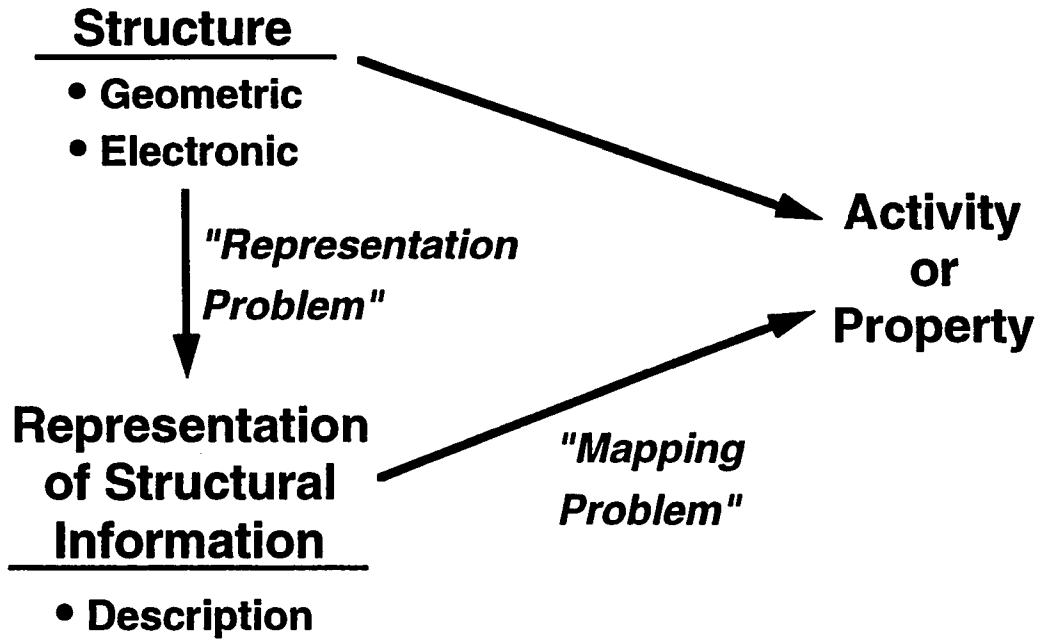


Figure 7. Mapping *vs* Representation Problem.

consisting of the two lower arrows represents the pathway that chemists employ when using computational science to model this structure-property function. The expected result of the computational modeling is the same as the experimental approach, namely to be able to accurately predict properties from molecular structure.

Computational neural nets as well as the other methodologies typically employed in quantitative structure-activity relationships (QSAR) and quantitative structure-property relationships (QSPR) studies follow the path designated by the two lower arrows. The first and more difficult step in the path involves generating a set of descriptors that characterize the relevant molecular features of the molecules being studied. This is the *Representation Problem*, which must be addressed in essentially all scientific work. CNNs generally use vector-based input. The representation problem will be addressed in Chapter III, in the context of developing a representation for predicting chemical reactions. The second step involves determining the mapping from the representation space to the prediction space. This is the *Mapping / Classification Problem*. It should be noted that determining a correct input-output mapping for a complex function is more challenging than classification. The work described here is directed towards the more difficult mapping problem. The input-output mapping can also be interpreted as a response surface, in which values of the system input variables give rise to particular system responses, and this

interpretation will be used here.

We studied the mapping ability of CNNs by addressing a number of the neural network issues raised in Chapter I within the context of a model problem. The model problem has many of the characteristics of real problems of interest in the areas of quantitative structure-activity relations (QSAR) and quantitative structure-property relations (QSPR).[55,122,123,143] The questions addressed were, given a set of compounds and their structural descriptors drawn from an unknown and unspecified distribution: (a) can a CNN develop an accurate mapping of the response surface for those compounds, (b) how important is sample size compared to the number of weights in the network, (c) what effect does the shape of the transfer function have on the mapping, (d) is a CNN mapping robust to the effects of noisy data, and (e) what is the effect on the CNN's mapping ability of correlations among the variables. Some of these questions had been studied to some degree by previous workers,[34,42,55,143] but there did not seem to be a coherent statement available about the capabilities of computational neural networks as model-free mapping devices.

Mapping Methods

Figure 8 shows the relationship between various types of mapping methods. As seen in the figure, the neural network methods have considerable overlap with the other statistically-based methods, but the

Mapping Methods

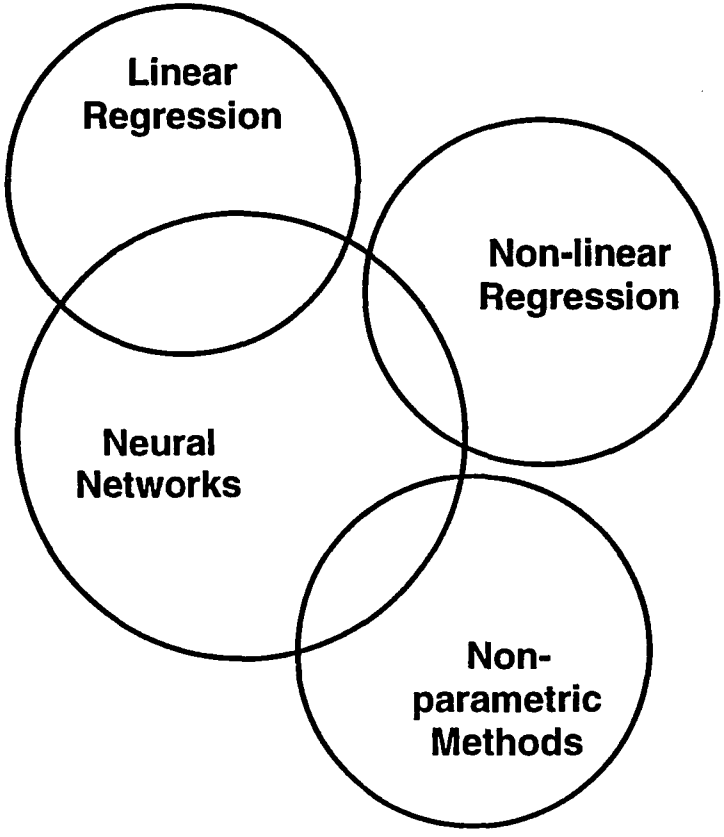


Figure 8. Mapping Methods.

CNNs also have their unique features. The main difference between neural network methods and other mathematical mapping methods is that in the traditional methods one must decide on what type of model to use and then apply that model. For example, in linear regression or linear discriminant analysis, a linear model of the form $R = aX + bY + c$ is first chosen and then the coefficients, a, b, and c are determined by the regression process to give the best fit to the data. In the CNN method, one starts with a set of very general, non-linear models initialized with randomly generated coefficients, and the network training process gradually optimizes the weights on the nonlinear functions to give a model that best reflects the underlying structure of the data. The CNN method does not require that any assumptions need to be made about the distribution of the data or about its underlying structure, except, perhaps the implicit assumption that there is some relationship between the input variables and the observed response.

Model Problem: Mapping a 3D Response Surface With a Neural Network

Typically, QSAR and QSPR studies are confined to small sets of relatively similar molecules. However, it would be advantageous to take a more *global* view of the relationship of the geometric and electronic structural features of molecules to their biological activities and to their molecular properties such as solubility, melting point, and partition

coefficients. As noted above, the relationship of biological activity to structure, for example, can be described as a *response surface*, such as the one depicted in Figure 9. In this case, the simple 3D surface is characterized by the two independent variables, x_1 and x_2, which label the coordinate axes representing molecular-feature descriptors, and $R(x_1,x_2)$ represents the response surface or function, which is given as the sum of three radially-symmetric Gaussian functions located at (2.0, 5.0), (7.0, -2.0), and (-4.0, -6.0), all with widths of 0.1 and heights of 10.0, 9.0, and 6.25, respectively. Each response surface corresponds to a particular type of biological activity measurement, or test system, while the region of each peak on the surface corresponds to a class of relatively similar, biologically active compounds—thus, $R(x_1,x_2)$ represents three classes of biologically active compounds. The situation modelled here by the simple "three-Gaussian" response surface is quite prevalent in actual studies of biologically active compounds, where different, seemingly unrelated classes of molecules may possess similar biological activity in a given test system.

A potential benefit of the model-free approach embodied in CNNs is that there is no need to specify the functional form of the "structure-activity mapping" explicitly. Within the context of the response-surface model, a number of characteristics of the mapping ability of CNNs can be investigated. Moreover, by considering a response surface that is visualizable in three dimensions it is possible to understand the role played

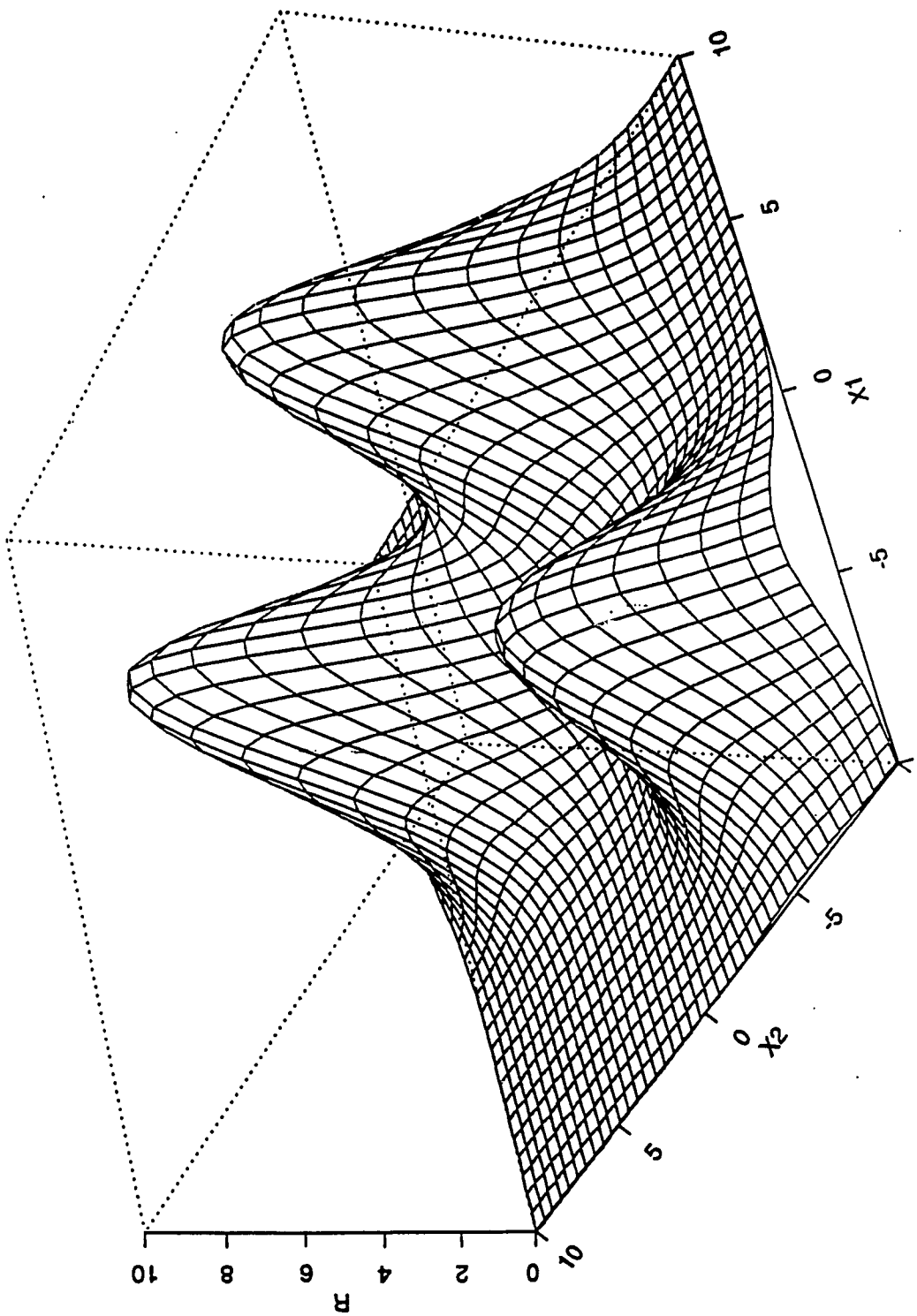


Figure 9. Response Surface.

by various factors such as sample size, noisy and missing data, and linear and non-linear correlations among the input variables.

Although it is possible to realize essentially all reasonably well-behaved mappings with a single layer of hidden units[29-32] (*vide supra*), Lapedes and Farber[42] have shown that two hidden layers are more efficient than one. Their work suggests that a "bump" function, which closely resembles a multi-dimensional Gaussian, can be modeled with five PEs arranged in two hidden layers, the first layer containing four PEs and the second layer a single PE. This relationship does not, however, necessarily extend to the "three-Gaussian" response surface studied here since the sigmoid or *tanh* transfer function of a given PE may contribute to the description of more than one Gaussian "bump." The guidelines suggested by the work of Lapedes and Farber would indicate that a feedforward network with 12 PEs in the first hidden layer and 3 PEs in the second layer would be able to accurately map the "three-Gaussian" surface studied here. We started with a network that had two hidden layers of 20 units each. This network was able to fit the surface from the training data but it had 501 weights, which would require a great deal of training data. Successively smaller networks were tried which had two hidden layers of 10 each and finally with 5 in each hidden layer. The neural network with two inputs, x_1 and x_2, two hidden layers of five PEs each, and a single output PE was found to be able to learn to fit the 3D response surface

efficiently. Each PE used the same *tanh* transfer function (see Figure 2), which upon proper rescaling yields the output or response function $R(x_1,x_2)$. Such a CNN may be designated as a 2-5-5-1 net to indicate the number of nodes in the input, hidden and output layers, respectively. The number of weights can be determined directly from this information ($2 \times 5 + 5 \times 5 + 5 \times 1 = 40$ connection weights plus $5 + 5 + 1 = 11$ bias weights, for a total of 51 weights). The network was trained using the extended-delta-bar-delta (EDBD) modification[64] to the standard, gradient-based backpropagation of error (BP) algorithm.[37,38] The NeuralWorks Professional II Plus program version of EDBD[144] was used for the response surface studies. This EDBD training procedure was found to produce a CNN that represents the response surface quite accurately (RMS error = 0.29, see Table 3) when trained on a uniform 30×30 grid of 900 sample points.

A uniform 40×40 grid of 1600 test points (see Figure 10) was used to evaluate the accuracy of the mapping produced by the 2-5-5-1 net, and the results showed that the net was capable of reproducing the true response surface to within an RMS error of 0.38. Several other network topologies containing more PEs in the hidden layers were also investigated (*vide infra*), but the 2-5-5-1 net was used in essentially all subsequent experiments due to its relative simplicity and its ability to represent the response surface accurately (*i.e.* generalize) even for relatively small sets of training data. Most importantly, it should be recognized that the testing

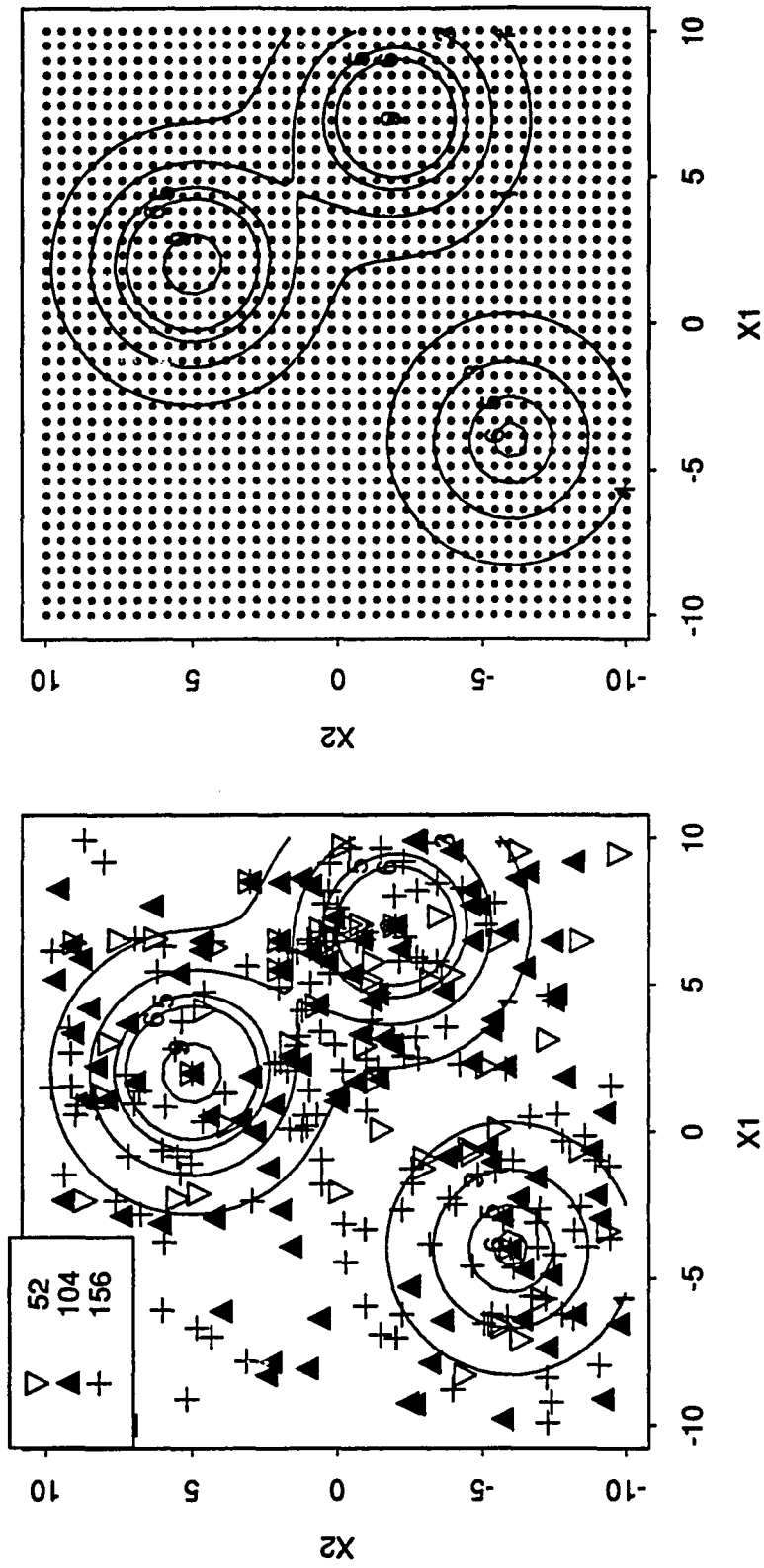


Figure 10. Data Sets of 1600, 156, 104, and 52 samples.

grid of 1600 points represents a very stringent test of the 2-5-5-1 net's ability to accurately predict the response surface over a much larger domain of the input variables than is usually the case in real QSAR or QSPR studies, where a rather limited set of test samples is typically considered.

Sample Size

For many interesting QSAR and QSPR problems only a limited set of compounds is generally available for training and testing a CNN, typically 10-50, although sometimes even as many as several hundred may be available. Since statistical considerations suggest a minimum ratio of three samples per weight[55] it is often difficult in practice to separate the effects of inadequate training datasets from possible limitations in the mapping ability of a given CNN. The response-surface model studied here, where the actual function to be modeled is known, allows one to choose data sets of arbitrary size so that issues of sample size versus number of network weights can be studied in great detail.

The original training set of 900 evenly distributed points had a samples-to-weights ratio of approximately 17:1. Three smaller training sets containing 156, 104, and 52 sample points (see Figure 10), with ratios of samples-to-weights of 3:1, 2:1 and 1:1, respectively, were chosen to investigate the effect of sample size on the ability of the 2-5-5-1 net to accurately represent the response surface. These training sets were chosen

such that they approximate distributions of compounds that might occur in real QSAR studies. When an active lead compound is found, the "chemical space" near the lead is explored thoroughly, while when an inactive analog is produced, that area of the chemical space is avoided if possible. Accordingly, for each training set, one quarter of the points were taken from a uniform random distribution in the square region surrounding each of the three peaks of active compounds, and the remaining one quarter of the sample points were taken from a uniform random distribution over the entire domain of the response function shown in Figure 9.

The results given in Table 3 show that even when the ratio of samples to weights was 1:1, the 2-5-5-1 network did a reasonable job of "learning" the response function. Visual inspection of the response surfaces depicted in Figure 11, which were obtained from sample sets of 900, 156, 104, and 52 data points, respectively, clearly shows that all four surfaces are *qualitatively* of the same shape as the test surface depicted in Figure 9. All of the major features of this surface are accounted for by the surfaces in Figure 11, even the surface obtained from the 52 point training set, which had approximately the same number of samples as weights.

Figure 12 shows the output from each of the five PEs in the first hidden layer of the 2-5-5-1 net trained on 156 data points. Each of the surfaces, which resemble a three-dimensional "shelf," are generated from suitably weighted linear combinations of the inputs that are then passed

Table 3

Characteristics of Feedforward Neural Net Descriptions of the "Three-Gaussian" Response-Surface Model

Number of Training Data Points	Character-istics of Data Points	Network Architec-ture	Samples/ Weights Ratio	RMS[e] Error Training Set	RMS[e] Error Test Set
900	–	2-5-5-1	17:1	0.29	0.38
156	–	2-5-5-1	3:1	0.35	0.49
104	–	2-5-5-1	2:1	0.48	0.78
52	–	2-5-5-1	1:1	0.35	0.94
156	noisy[a]	2-5-5-1	3:1	0.77	0.79
52	noisy[a]	2-5-5-1	1:1	0.80	1.72
52	noisy[a,b] (averaged)	2-5-5-1	1:1	0.52	1.17
156	noisy[a]	2-40-40-1	1:12	0.57	1.02
156	random[c]	3-5-5-1	~3:1	0.28	0.47
156	correl.[d]	3-5-5-1	~3:1	0.27	0.40

[a] Gaussian noise was added to the response function.

[b] Five noisy replicates at each data point were averaged and the averaged values were then used to train the CNN.

[c] A third independent variable uncorrelated with x_1 and x_2 was added.

[d] A third independent variable quadratically correlated with x_2 was added.

[e] The Root Mean Squared (RMS) error was calculated as
$RMS = (sqrt(\Sigma(R_{Target} - R_{CNNpredicted})^2)) / \text{Number of data points.}$

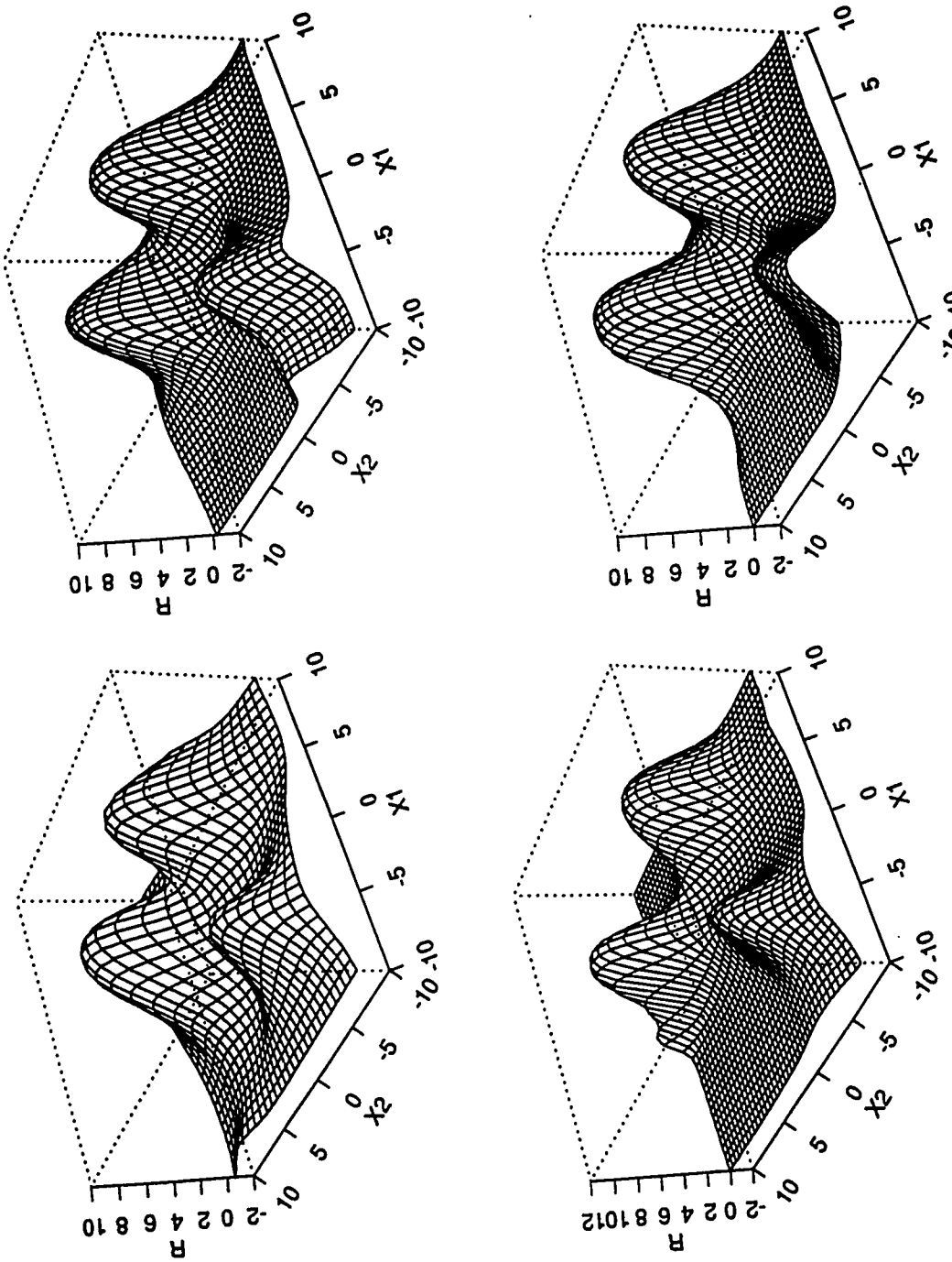


Figure 11. CNN Mapped Response Surfaces Trained on 900, 156, 104 & 52 Samples

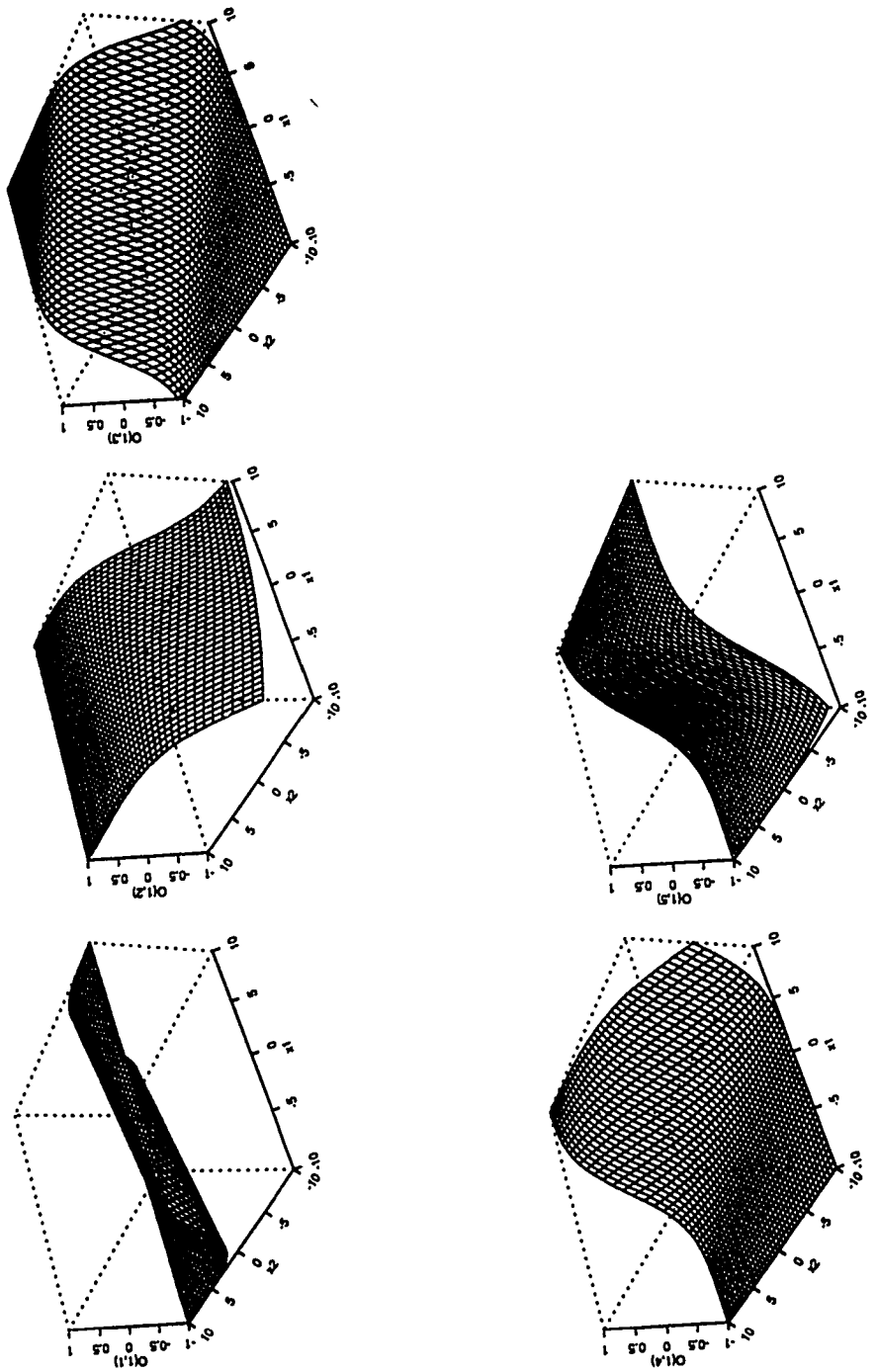


Figure 12. Outputs of Hidden Layer 1 PEs.

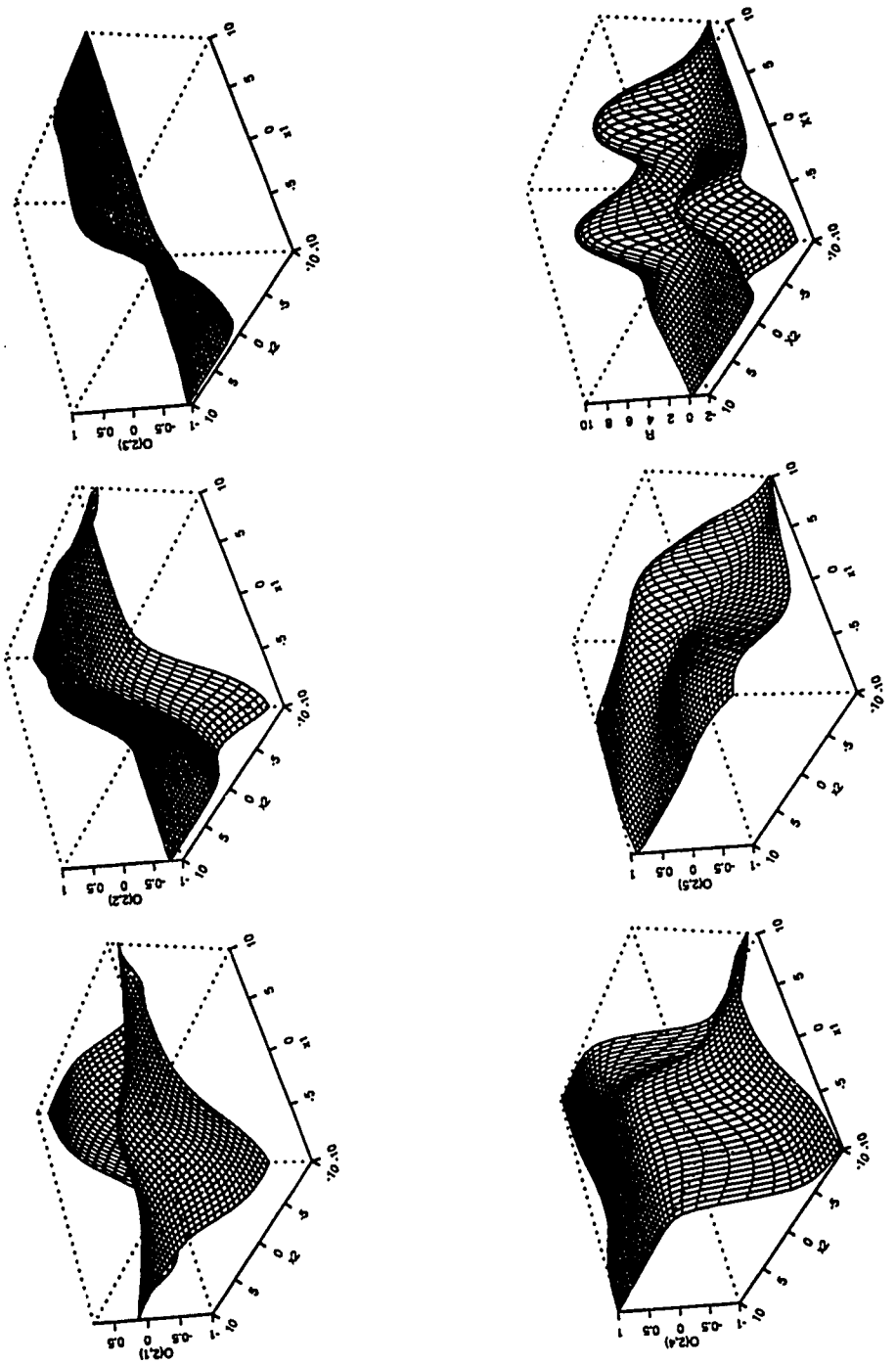


Figure 13. Outputs of Hidden Layer 2 PEs.

through the non-linear, *tanh* transfer function in each PE.[42] Figure 13 shows the output from each of the five PEs in the second hidden layer. These more complicated surfaces are obtained, as were those in the first hidden layer, from linear combinations of surfaces generated in the first hidden layer that are then passed through the non-linear *tanh* transfer function in each of the PEs in the second hidden layer. The final surface, *i.e.* the response surface of interest, is generated by the single PE in the output layer, which again appropriately combines the outputs from the PEs in the second hidden layer and outputs this value through the non-linear function of the output PE. The output is then rescaled to provide the desired response surface.

Transfer Functions

As noted earlier, the form of the transfer functions can be altered by the choice of the gain parameter, α (see Figure 2). Generally, the gain parameter is taken to be $\alpha = 1.0$, and this is the value used in essentially all the studies carried out in this work. However, two 2-5-5-1 CNNs were investigated in which the gain parameter was taken to be $\alpha = 10.0$ ("step function") and $\alpha = 0.1$ (approximately a "straight line"), respectively. In both cases, learning did not converge, which is not surprising given the shape of the three-Gaussian response surface. In some cases radial Gaussians have been used as transfer functions.[145] Based on the earlier

discussion of Lapedes and Farber regarding "bump" functions,[42] it is expected that radial Gaussians can provide a suitable "basis" for representing many input-output mappings of interest, especially surfaces similar to the response surface studied here. Nevertheless, we have chosen to carry out our work with the more traditional *tanh* transfer function for two reasons: (1) radial Gaussian transfer functions were not available in any of the neural network software that we had available, and (2) *tanh* transfer functions have been shown to be of very general utility in neural network mappings.[42]

Noisy Data

The experiments reported above were conducted in the absence of noise, but in real situations noise is present to some degree in all measurements. Accordingly, a noisy response surface depicted in Figure 14 (surface a) was generated by perturbing the smooth response surface with Gaussian noise. The noisy datasets were generated by adding Gaussian noise, with a mean of zero and a standard deviation of one, to the value of the response variable for the 900, 156, and 52 point datasets. The generation of both the smooth and the noisy response surfaces, as well as the selection of the randomly generated 156, 104 and 42 point data sets were done using the S-Plus statistical software[146] on a Sun Sparcstation 1+. Examination of the figure (Figure 14 surface a) shows that while the

general topography of the response surface remains, some of its more subtle features are obliterated.

Figure 14 (surface b) shows the response surface of a 2-5-5-1 net trained on 156 noisy points. It is clear from the figure that the general features of the response surface are reproduced here. As the number of sample points is decreased the response surface becomes distorted, as is seen in Figure 14 (surface c) for a 2-5-5-1 net trained on 52 noisy data points. In real experimental situations it is not always feasible or desirable to increase the number of data points, especially if each point represents a different compound. In such cases, standard statistical averaging procedures can help remove some of the experimental error, a procedure which is quite feasible and highly desirable in most experimental situations. This is illustrated in Figure 14 (surface d), where the shape of the response surface of a 2-5-5-1 net trained on 52 *averaged* data points is considerably improved over the surface shown in Figure 14 (surface c) and approaches the shape of the surface obtained from an identical net trained on 52 non-noisy data points (see Figure 11 surface d). Each of the averaged data points was obtained by averaging five replicates with Gaussian noise at each of the 52 points.

As shown in Table 3, a 2-40-40-1 net with 1801 weights (approximately 1 sample per 12 weights) trained on 156 noisy data points yielded an RMS error of 0.57, which was less than the value of 0.77 for the

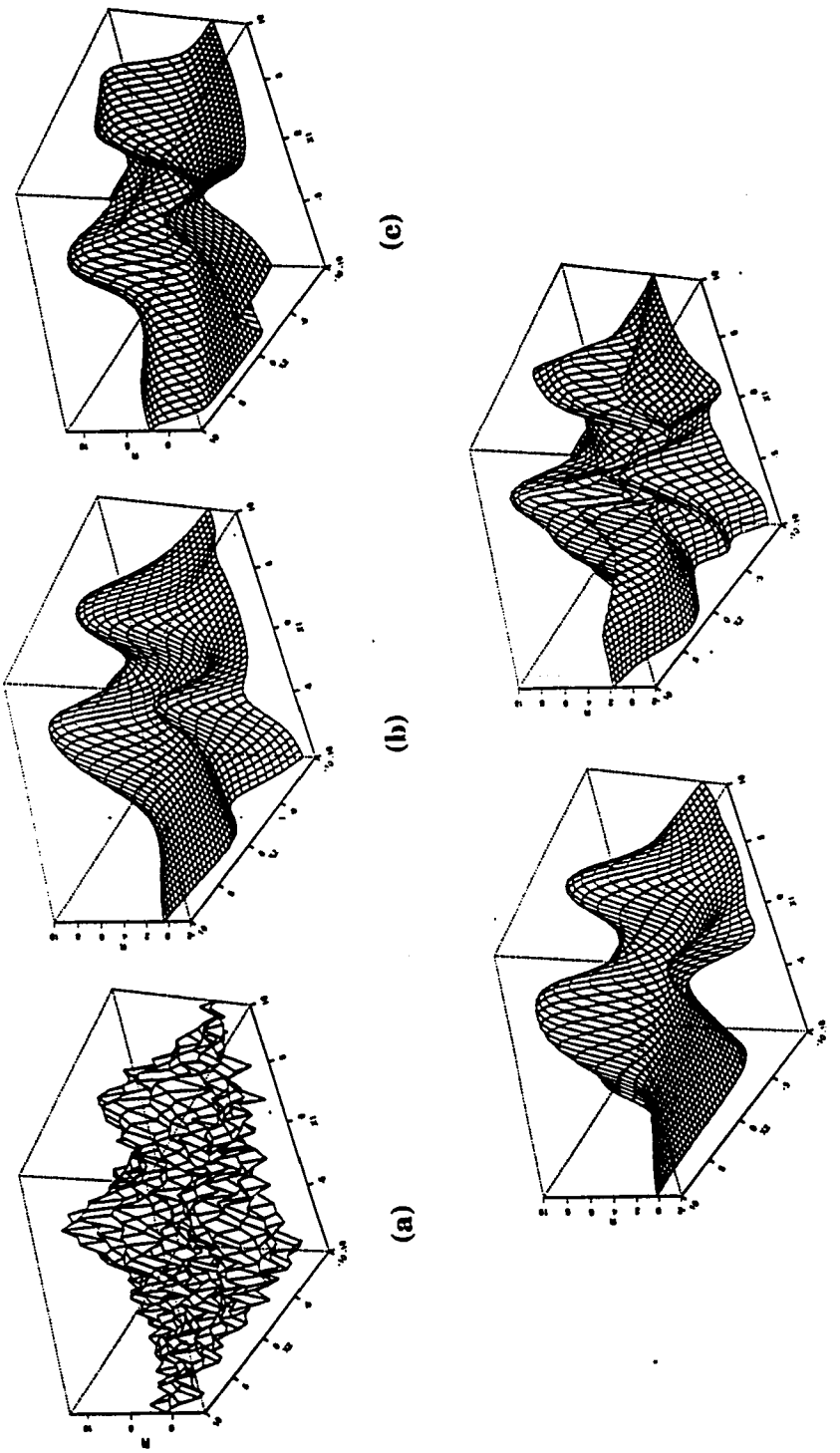


Figure 14. CNN Mapped Response Surfaces Trained on Noisy Data.

corresponding 2-5-5-1 net. On the test set, the 2-40-40-1 net had an RMS error of 1.02, which was worse than the corresponding value for the 2-5-5-1 net. This behavior is expected and indicates that the while the larger net, due to its greater complexity, was able to "learn" the noisy data better, it was less able to generalize correctly. Figure 14 (surface e) depicts the response surface obtained from the 2-40-40-1 net. The highly "ruffled" surface is clearly indicative of the net's ability to learn the noise in the data.

The fact that the simpler 2-5-5-1 net was able to model the response surface with 156 noisy data points and generalize from the test set (see Table 3) may be due to the *relative* simplicity and smoothness of the surface investigated here (see Figure 9). A more complex and highly-variable function would provide a more stringent test of a CNN's ability to generalize. More work is needed in this area to clarify the relationship of network complexity to sample size.

Representation Issues: Correlated Variables

As noted earlier, finding a suitable representation to treat a given problem or class of problems is a demanding and difficult task. Finding the *optimum* representation is an even more daunting task. Chemical systems, in particular, provide a significant challenge due to the fact that their "natural" representation is in the form of chemical graphs.[147] Vector-based

representations, as illustrated by the example given here, are employed in most CNNs and in essentially all cases investigated to date based upon generalized perceptrons. In the example considered in this work, the precise nature of the two independent, descriptor variables, x_1 and x_2, was not considered. In real applications, however, this is generally one of the major problems confronting the researcher trying to investigate QSAR, QSPR, or related problems, especially in cases where it is desired that the independent, descriptor variables provide a characterization of a significant portion of the "chemical universe." Typical issues that must be addressed by researchers regarding the choice of descriptor variables include the nature of the variables, whether they can be calculated or must be determined experimentally (and thus whether values are available for all of the compounds of interest), the number of variables required to fully characterize the system under study, and whether dependencies among the variables exist due to linear or non-linear correlations.

We have investigated the issue of correlated variables in terms of the response-surface model system described above and the results are summarized in Table 3. For example, when a third input variable, x_3, was added to the system but taken to be random (*i.e.* uncorrelated with x_1, x_2, and R), all the weights of the trained CNN connecting x_3 to the nodes in the first hidden layer were zero, and the RMS error of the net was comparable to that trained on only two input variables. When x_3 was taken

to be non-linearly correlated with x_2 (*i.e.*, $x_3 = \frac{1}{2}x_2^2$), the 3-5-5-1 net learned and generalized as well as the net trained on just x_1 and x_2. These results suggest that a CNN's ability to model a response surface is not materially altered due to extraneous input variables nor by correlations among input variables, as long as the input variables contain sufficient information that is relevant to the mapping being represented. Nevertheless, it is desirable to reduce the dimensionality of the input space as much as possible so that the complexity of the CNN does not become too great. Methods for reducing the dimensionality of the input space have been reported.[58] These methods have included CNN-based non-linear principal components analysis[57] and disabling the network inputs one at a time to find inputs that have little effect on the network's learning and generalization.[59]

Simple Chemical Relationships Mapped by a Neural Network

The results of the preceding section illustrate clearly that CNNs of the generalized perceptron type can map fairly complex relationships, such as the response surface in Figure 9, even with small samples of noisy data. Correlated and spurious inputs also did not diminish the mapping ability of the network. The following examples further reinforce this conclusion by showing how a neural network can give a better fit of the data than traditional statistical methods, without having to specify the form of the equation or without knowing the distribution of the data.

Prediction of ^{17}O NMR Shifts

One example, the correlation of E-state index with ^{17}O NMR shifts, was given in Chapter I. In that study a single 1-3-1 CNN was able to model the relationship between the E-state index and the oxygen chemical shift for two classes of compounds, whereas the linear regression methods used by Hall and Kier[51] required treating each class separately. The other example, which is described below, involves the prediction of the tumor promoting ability of a set of compounds from their hydrophobicity.

Nonlinear or Bilinear Data Mapped by CNN

Rippmann[148] reported on the structure-activity relationship between the calculated *logP* and the tumor promoting ability of a series of 25 aliphatic phorbol 12,13-diesters. Forty-two experimental measurements on the 25 compounds were reported. The compounds in the study were combinations of acetate, butyrate, pentanoate, hexanoate, heptanoate, octanoate, decanoate, undecanoate, dodecanoate, tridecanoate, tetradecanoate, hexadecanoate, and octadecanoate esters at the 12 and 13 positions of phorbol (shown in Figure 15). The author modeled the log of the relative tumor promoting ability *log(RTPA)* of the phorbol esters with the calculated *LogP* values of the phorbol esters. The data was suggested to fit either a bilinear distribution:

$$log(RTPA) = +0.441\ Log(P) - 0.738log(\beta P + 1) - 2.571$$

or a parabolic relationship:

$$log(RTPA) = -0.056\ (log\ P)^2 + 0.650\ log\ P - 2.657$$

where P is the octanol-water partition coefficient. However, the data showed some deviation from both of these "ideal" curves. An EDBD 1-3-1 network was trained on the same data and the results are shown in Figure 15. The model curve predicted by the neural network is intermediate between the author's two models. This result suggests that the neural network is able to generate a model that fits the data better, without having to force the data to fit an arbitrary model.

Summary

Generalized perceptrons can function as powerful, essentially model-free, mapping devices that can be applied to a wide range of problems in chemistry.[70] The power of these CNNs lies in their ability to represent very general mappings without the need to specify the mathematical form of the mapping explicitly. The three-dimensional response surface with two input variables studied here is relatively simple, yet, it illustrates and clarifies a number of important issues that are relevant to the applications of generalized perceptrons in general and in chemistry in particular. These issues include choice of an appropriate data representation, correlated variables, the effect of the shape of the transfer function, and evaluation of

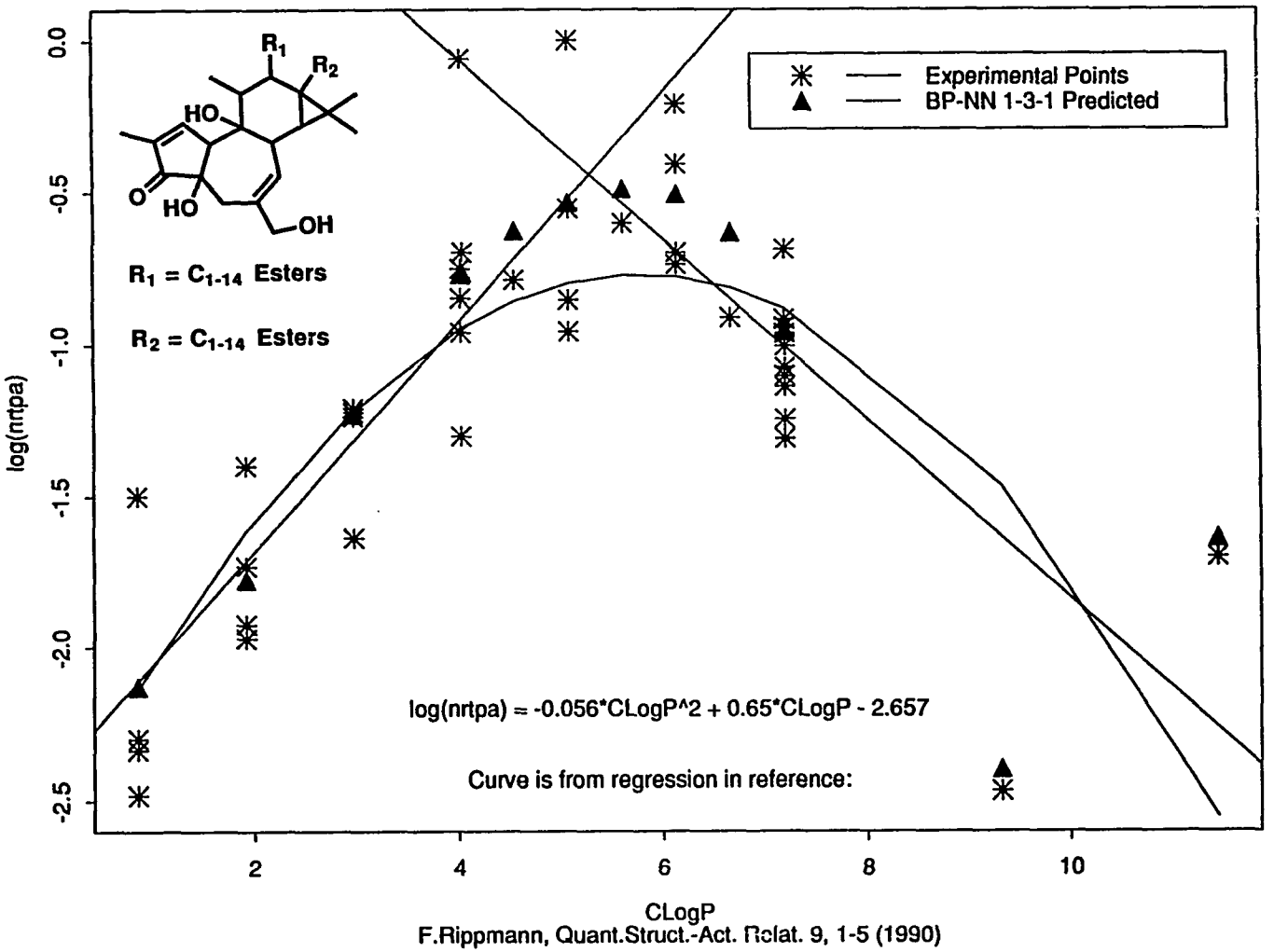


Figure 15. Phorbol Esters: Plot of Tumor-Promoting Ability *vs* CLogP.

the network's ability to fit training and test data (*i.e.* to generalize) effectively. In addition, problems due to small sample size (relative to the number of weights), and noisy data, which plague many chemical applications of CNNs, must be dealt with. However, the above example shows that even in cases where the number of samples relative to the number of weights is less than the minimum number required, *i.e.* three,[55] generalized perceptrons can still be trained to provide a reasonably accurate representation of the desired response surface (see Table 3). Other applications of generalized perceptrons to chemical reaction prediction, in Chapter III, will also show that reasonable generalization could be obtained from nets trained on a minimal number of samples, even when the number of samples was less than the number of weights. In such cases two important points should be made. First, in many chemical studies only a relatively small number of compounds may potentially exist (*i.e.*, certain molecular structures are inherently unstable and cannot be studied using normal chemical methods) and hence, predictions made in such cases can only be based on limited data. Second, knowledge of structural as well as other types of chemical information can be of assistance in choosing a "representative" training set. While such an approach admittedly introduces a bias into the learning process it, nonetheless, may provide a satisfactory means for studying chemical systems where only sparse data exists and classical statistical methods are not applicable. Statistical

methods such as "leave-one-out" or bootstrapping[71-74] can, in some cases, provide the means for addressing the "small sample problem," but the power of these methods has not been fully exploited in CNN applications to date. Additional work is needed in this area.

Once a CNN is sufficiently trained for a particular QSAR or QSPR application, it may be of interest to determine the location of extrema of interest on the response surface—whether the extrema are maxima or minima depends upon the particular application. In such cases these extremal points represent potentially active compounds (maxima) or compounds with maximal or minimal values for a given property. This more global approach to QSAR or QSPR represents an important new direction for CNNs and will provide a potent new tool for use in computer-aided molecular design.

In low-dimensional cases, such as the one illustrated here for two-dimensions, it is possible to locate extrema by evaluating the response function on a uniform grid. For higher-dimensional cases this is not feasible and other optimization methods must be used. As the response surfaces generally possess numerous extrema, gradient-based methods are doomed to failure. However, methods such as those based upon simulated annealing[67] or genetic algorithms,[68] which were discussed earlier with regard to the training of CNNs, are also applicable here, and only a slight modification of these algorithms is needed to adapt them to this problem.

CHAPTER III

NEURAL NETWORK PREDICTION OF REACTIVITY IN EAS REACTIONS

Representation of Chemical Structures for Neural Networks

Background

We argued in Chapter II that the prediction or modeling of chemical reactions is a two-stage process, with the first stage being the selection of an appropriate representation of the chemistry and the second stage being the mapping of that representation to the reaction of interest. Failures at either stage will prevent meaningful results from being obtained. Computational neural networks were shown to act as "model-free" mapping devices by their ability to fit a 3D response surface, but the meaning of the input variables used was not specified. In order to explore the mapping abilities of neural networks for computer assisted organic synthesis problems we must next develop a means for representing chemical information that is suitable for using to train a neural network. All of the authors of the papers reviewed in the QSAR and property prediction sections of Chapter I used physicochemical property descriptors for the network input. We felt that a representation more directly tied to the

molecular structure would be more general and might give results closer to what chemists would predict. The requirements for such a representation, the implementation of a connectivity-based representation, and its successful use to predict a class of reactions are the subject of this section.

General Requirements for a Representation

Chemists have developed an elegant language in chemistry that richly embodies chemical concepts and provides a framework for communicating and understanding much of organic chemistry. This language employs two-dimensional chemical structure diagrams, either singly, to communicate information about a compound, or using several structure diagrams, to convey changes that occur in chemical reactions. According to Lawler of the Institute for Scientific Information "If chemical structures are the language of chemistry, then reactions are the prose (p 57)."[149] Wilcox and Levinson[11] suggested that because of the use of chemical structure diagrams by chemists, the problem of representing chemical knowledge had been solved. Chemical structures are indeed a familiar language to chemists, and it is our conviction that the same kind of information that is available to chemists when they use structure diagrams would provide the most natural representation for neural network use. Wilcox and Levinson may be right that chemical structure diagrams are sufficient for chemist's use but chemical structure diagrams must be

converted to another form for computer storage and manipulation. The most appropriate computable chemical structure representation depends on the purpose for which it will be used, and was a major issue to resolve for this research.

All attempts to describe and represent chemical structures have some shortcomings because they are only approximations of chemical reality. The only true representation of a chemical structure is a molecule or collection of molecules themselves. A quantum mechanical description, in terms of the molecule's wave function Ψ, may mathematically embody the electronic and other properties of the molecule, but it is not a representation that is easy to use to compare related compounds. On the other hand, connection tables have been the most widely used type of computer-representation of chemical structure.[150]

Representation can be described at several levels of detail. In increasing order of accuracy are molecular formula, chemical constitution, configuration and conformation. We will be considering the first two of these: the number and types of atoms in the molecule and how they are attached. Stereochemical configuration and three-dimensional molecular conformation are necessary for a complete understanding of chemical reactivity, but are not treated here. Our approach was to prepare neural net input patterns for molecules that reflect their two dimensional structure; because chemists can infer a large amount of chemistry from

two-dimensional structure diagrams. There is not yet a general consensus among chemists on how to represent three-dimensional structures for conformationally flexible molecules which may have multiple conformations with energies within a few kilocalories of each other.

An excellent, up-to-date reference on structure representation, Chemical Structure Systems , by Ash, Warr and Willett,[151] appeared in 1991. The chapter by Barnard[150] enumerates the various types of representation methods used by chemists and chemical information scientists to store and utilize chemical information in computer systems. Connection tables, adjacency matrices, bond distance matrices, line notations such as Wiswesser Line Notation (WLN), and the more recently developed Simplified Molecular Input LinE System (SMILES) notation, geometry distance matrices, fragment lists and directed graphs have all been used as representations of molecular structure.

A crucial requirement of a representation is that it contain a sense of similarity: *ie*, similar molecules should have similar representations and similar properties.[152] In this regard, graph theoretical representations would seem to be a natural choice for an input representation, because 2-dimensional chemical structures are equivalent to labelled topological graphs.[150] However, as noted earlier, graph representations are difficult to implement in a general fashion in CNNs (*Cf* refs 16 and 54). Other requirements of a representation method are that all molecules of interest

can be represented, that all molecules give a unique or at least a consistent representation, that it be scalable to handle a range of molecular sizes, and that it contain information about functional groups or molecular neighborhoods around reacting centers. A good representation should contain approximately the same amount of information as 2D chemical structure diagrams, so that the structure could be unambiguously reconstructed from the representation.

Some of these requirements for a representation are illustrated in the following section which describes the use of graph theoretical indices for CNN prediction of boiling points. The approaches used by others (*vide supra*) mainly employed calculated physicochemical properties or substituent constants. One of the problems with these approaches is that it is not always possible to find substituent constants for uncommon substituents. Sometimes it is not possible to calculate the required physicochemical properties and thus some of the compounds or reactions must be left out of the analysis. This latter problem occurred twice in the course of using other non-connection table representations that we tried to develop. They will be discussed later in this chapter.

Graph Theoretical Index Representation for Predicting Boiling Points

The following example shows a chemical representation that is useful for other applications but has inherent limitations as an input

representation for a neural network. Johnson[153] reported a comparison of several statistical and molecular similarity-based methods for predicting the boiling points of octanes. Since the boiling point and octane number are related to the amount of branching for a set of isomeric hydrocarbons, the graph theoretical indices P2 (number of different paths of two bonds in length) and P3 (number of different paths of three bonds in length) were used as independent variables. Figure 16 shows the structures of the 18 isomeric octanes, their path counts P2 and P3, and their boiling points. Using linear regression Johnson obtained an average error of 2.6° C, while the best method, a nearest difference prediction gave an average error of 1.4° C. A BP 2-2-1 neural net was trained with backpropagation training, using the P2 and P3 values as inputs, with 2 hidden units, and the boiling point as the single output. The CNN gave an average error of 2.0° C when all 18 were use for training and testing. Leave-one-out training was next done, where the network was trained on 17 compounds and tested on the eighteenth. The leave-one-out training was repeated for all 18 of the octanes and the predictions on the compounds not in the training set were averaged to give an average error of 3.2° C. The results are in Table 4.

While the neural network results are comparable to the other methods, there are limitations in using P2 and P3 as structure descriptors for input to the CNN. First, there is error due to lack of resolution by the representation, because two compounds, 3-methylheptane and 4-methyl-

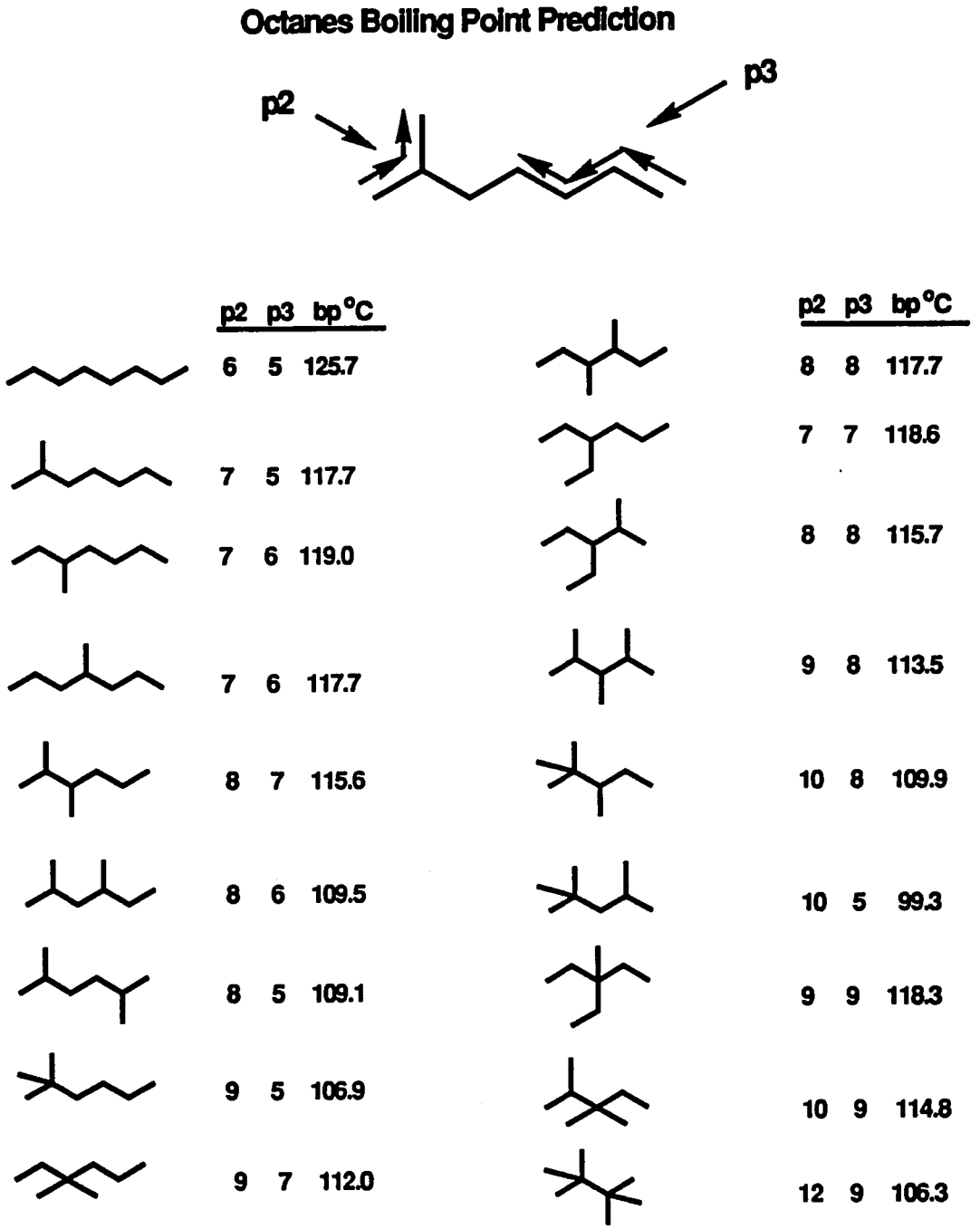


Figure 16. Octanes: Structures, Boiling Points, and Paths.

Table 4

CNN 2-2-1 Predictions of Boiling Points of Octanes from Paths P2 & P3

P2 Paths of Length 2	P3 Paths of Length 3	bp °C	BP 2-2-1* Trained on all 18 Test on 18	BP 2-2-1** Train on 17/18 Test on 18
6	5	125.7	121.5	118.8
7	5	117.7	117.2	117.0
7	6	119.0	119.4	118.1
7	6	117.7	119.4	118.8
8	7	115.6	116.8	116.5
8	6	109.5	114.5	113.8
8	5	109.1	111.1	109.5
9	5	106.9	105.3	101.9
9	7	112.0	112.0	110.9
8	8	117.7	118.1	118.6
7	7	118.6	120.2	120.8
8	8	115.7	118.1	118.9
9	8	113.5	114.9	113.2
10	8	109.9	110.4	109.1
10	5	99.3	101.6	105.4
9	9	118.3	116.6	116.0
10	9	114.8	113.7	110.8
12	9	106.3	105.2	106.5

* Trained on all 18 octanes.
** Trained on 17 of 18 octanes. Repeated for all 18.

heptane, have the same input description, P2 = 7 and P3 = 6, yet they have different boiling points, 119.0° C. and 117.7° C, respectively. Secondly, the representation is ambiguous, because there are three compounds with the same boiling point (117.7° C) but different input pairs, namely (P2,P3) of (7,5), (7,6), and (8,8), respectively. Thus the representation is not adequate, and no matter how powerful the mapping ability of the neural network, it

cannot produce an error free mapping from this representation.

Another source of error, which is not due to deficiencies in the representation, but to the nature of the neural network itself, is the propensity of the CNN to interpolate rather than extrapolate. This can be seen in the predictions for the compounds with the largest and smallest boiling points. In Table 4, n-octane, with the highest bp of 125.5°, is predicted to be 121.5° when it is included in the training set but only 118.8° when it was left out of the training set. Similarly, 2,2,4-trimethylpentane, which had the lowest bp of 99.3°, was predicted to boil at 101.6° when included in the training set and 105.4° when it was not part of the training set. In both of these cases, the errors at the extreme ends of the range of boiling points shift the predictions for the compounds at the extremes towards the middle of the range, even when those extreme compounds are used for training. This shift towards the rest of the compounds is even more pronounced when those compounds are not included in the training set.

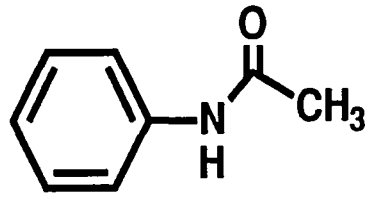
Connection Table Representations

One of the lessons learned during the work in chapter II on the response surface modeling by a CNN is that the descriptors need to be general but they do not have to be orthogonal. The neural network input requirement of having vectors or lists of numbers as input puts a constraint

on a chemical structure representation for this work. SMILES and WLN use strings of characters and typographic symbols, for which it is difficult to devise a consistent, meaningful numeric recoding scheme. Fragment lists or codes are often used in computer programs for structure searching and comparison. The problem with fragment lists is that they are not unique and contain no information about the connectivity of the fragments. Connection tables or connectivity matrices contain both atom type information and connectivity information and thus were the starting point for our representation.

A connection table is a list of the atoms and bonds in a molecule. The atoms are arbitrarily numbered and usually listed in the order of their numbering. Each row of the connection table contains the atom number of the atom, the atom type (usually the atomic symbol), and a list of the numbers of the attached atoms, with an integer (1=single bond, 2=double bond, and 3=triple bond) designating the bond order of the connection. Ordinarily the hydrogen atoms are not numbered so that the connection table can be made smaller. The knowledge of the normal valence states of atoms allows deducing the number of hydrogens attached to an atom. Sometimes the hydrogen atoms, particularly on heteroatoms, are explicitly included to indicate an uncommon valence state. Usually, the connection table has redundancy, in that it contains all of the connections for all of the heavy atoms. Figure 17 shows a connection table for acetanilide.

Acetanilide



Connectivity Matrix

		1 C	2 N	3 C	4 O	5 C	6 C	7 C	8 C	9 C	10 C
1	C	0	1	0	0	0	2	0	0	0	1
2	N	1	0	1	0	0	0	0	0	0	0
3	C	0	1	0	2	1	0	0	0	0	0
4	O	0	0	2	0	0	0	0	0	0	0
5	C	0	0	1	0	0	0	0	0	0	0
6	C	2	0	0	0	0	0	1	0	0	0
7	C	0	0	0	0	0	1	0	2	0	0
8	C	0	0	0	0	0	0	1	0	2	0
9	C	0	0	0	0	0	0	0	1	0	2
10	C	1	0	0	0	0	0	0	0	2	0

Connection Table

Atoms		Bonds		
1	C	1	2	1
2	N	2	3	1
3	C	3	4	2
4	O	3	5	1
5	C	1	6	2
6	C	6	7	1
7	C	7	8	2
8	C	8	9	1
9	C	9	10	2
10	C	1	10	1

Figure 17. Connection Table and Connectivity Matrix for Acetanilide.

Related to the connection table is the connectivity matrix or adjacency matrix. A connectivity matrix for a molecule with n atoms is represented by an $n \times n$ square matrix, where there is one row and column for each non-hydrogen atom. The elements of the matrix contain the integer bond orders of the connections, as shown for acetanilide in Figure 17. In the connectivity matrix representation, the atom type information is stored outside of matrix. The diagonal elements are zeroes, indicating that an atom does not have a bond to itself. Sometimes the bond orders are left out of the matrix, which is then called an adjacency matrix. Connectivity matrices are symmetric about the diagonal, and also contain redundant information. The number of hydrogen atoms attached to a heavy atom can be inferred by summing the number of non-zero entries in a row (or column) and subtracting from the normal valence state of the atom.

Semi-Canonical Numbering of Atoms in Connection Table

One further point needs to be made here regarding devising a form of connection table for input to a CNN. While it is customary to show the connection table with a row for each atom, the neural network input is a vector, or more correctly, a list of numbers. The order of the input numbers, like the numbering of the atoms in a molecule, is arbitrary. However, in order for the network to have any possibility of learning, the numbering of the atoms must be consistent. This means that for a set of

molecules, or reactions, the corresponding atoms must be given the same numbering in all of the molecules in the set. There are no automatic procedures available to number the corresponding atoms consistently for a set of molecules. This step must be done manually, by a chemist who uses his knowledge about the problem to resolve possible conflicts.

There are methods such as the Morgan algorithm which generate unique or so called "canonical" numberings[150] but they were designed to give a unique name to a molecule that could be used for rapid database searching. The generation of a unique numbering for a molecule depends on priority rules much like the Cahn-Ingold-Prelog (CIP) rules for stereochemistry. Just as in the CIP rules, the substitution of a higher priority atom at a site remote from the one under consideration can reverse the final designation given to that atom. What is required is for someone to devise a "semi-canonical" numbering system, which would allow the chemist to select a common atom or set of atoms, and start the numbering from there to give a consistent numbering for the whole set. Perhaps even better would be for the semi-canonical method to suggest the starting point and let the chemist accept or reject it.

We investigated the importance of a consistent numbering in the following way. A set of seven mono-substituted benzene derivatives which had three non-hydrogen atoms in the substituent was chosen. These same compounds were taken from the ones used in the reaction prediction studies

(*vide infra*), where the substituents were: -COOH, -NO_2, -CH_2OCH_3, -$COCH_3$, -CH_2CN, -$CONH_2$, -$CH_2N^+H_2CH_3$. The three substituent atoms were numbered in the 6 possible ways, to give six different input vectors per compound, each with 6 columns (the same 5 columns as discussed in the following section, with an additional column for the count of hydrogens attached to that atom) and 3 rows (18 total inputs). An 18-6-7 network was trained by backpropagation on five of the six permutations of each compound until no further improvement was seen in the root-mean-squared error. Then the network was tested on the sixth. The 7 outputs corresponded to the seven different compounds. Thus, the network was trained to recognize a compound from most of its representations. The network correctly learned all 35 training cases (7 compounds X 5 permuted input representations). In all 7 out of 7 test cases the network correctly recognized the sixth representation as being the same as the other five for that compound. The network was retrained on 4 of 6 permuted representations and tested on the other two. Here again all 28 (7 X 4) of the training cases were correctly recognized but the network was right on only 6 out of 14 (43%) test cases. This was a very simple example, with only six possible numberings of the three atoms. For averaged-sized molecules the number of possible numberings is huge and the only practical way to avoid this problem is to make sure that the atoms are numbered in a way consistent with the problem being studied.

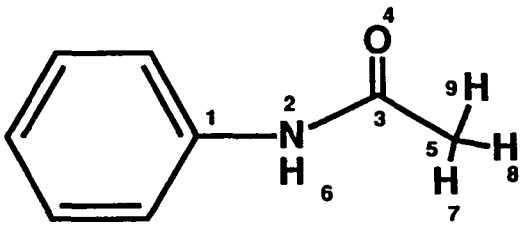
Modified Connection Table for Neural Net Input

The connection table described above was adapted to use an input representation for neural networks. Two modifications were required. The first was to devise a means of converting the atomic symbols to a numeric value. In a neural network, the input values serve as a multiplier of the weights to the hidden units in the first hidden layer. Therefore, the magnitude of the numbers should have some relationship to their importance or should be relative to a common scale. A simple way to represent atoms with a single number is by using the atomic number. The atomic number is related to the size, the electronegativity, and the number of electrons on an atom, and thus is a reasonably consistent representation. The second modification was to remove the redundancy in the connection table by only listing connections from the higher numbered atoms to the lower numbered atoms. The larger the number of inputs to a network, the more weights will have to be learned during the training, and the longer the training process will be. A potential problem with removing the redundancy is that it is much more difficult to determine how many non-hydrogen neighbors an atom has.

The modified connection table was designed to be as compact as possible while still containing all of the relevant chemical information. As discussed above, the numbering of the atoms is determined by the specific

reaction or property to be mapped by the neural network. Given the appropriate numbering, there is one row in the connection table for each non-hydrogen atom. The rows are ordered by the numbers given to the atoms: the first row corresponds to the first atom, the second row to the second atom, and so forth until all of the necessary atoms are assigned to a row. The determination of which atoms are necessary will be illustrated in the work that follows on predicting the products of electrophilic aromatic substitution reactions. Each row of the connection table contains five entries. These are illustrated in Figure 18. The first column holds the atomic number of the atom. The second and third columns contain the assigned numbers of the atom, and the atom to which that atom is attached, respectively. The fourth column has the bond order of the bond and the fifth column has the charge on the atom. Specifying the charge allows hydrogen atoms to be ignored by using charge to indicate differences from normal valence, such as for ammonium ions. A requirement of the neural network is that the input vectors or lists must be the same length for all of the examples in the data set. Reactants and products are often of varying sizes, so in order to encompass this range of molecular sizes, the connection table is made as large as the largest compound. Smaller compounds have the remaining rows filled in with zeroes.

Connection Table Representation



	Atomic Number	Atom 2 of Bond	Atom 1 of Bond	Bond Type	Charge
One Row	7	2	1	1	0
for Each	6	3	2	1	0
Non-Hydrogen	8	4	3	2	0
Atom in the	8	5	3	1	0
Substituent	0	0	0	0	0

Input Vector

7 2 1 1 0 6 3 2 1 0 8 4 3 2 8 5 3 1 0 0 0 0 0 0 0

Figure 18. Connection Table Representation Suitable for Neural Net Input.

Other Representations

Charge Vector

While it was felt that a connection-table derived representation would contain more relevant chemical information and thus generate better neural net predictions than would a representation based on calculated properties, we also looked at calculated partial charges for comparison. Partial charges may be useful in some cases, particularly in reactions where charged intermediates or transition states occur. Instead of representing the substituents directly, the effect of the substituent was represented by calculated charges. MOPAC,[162] a semi-empirical quantum mechanics program was used to calculate the Mulliken partial charges on each atom of interest. A vector of the calculated charges was used as the input for predicting electrophilic aromatic substitution reactions. Figure 19 shows the charge vector for the example compound acetanilide.

Graph Transforms

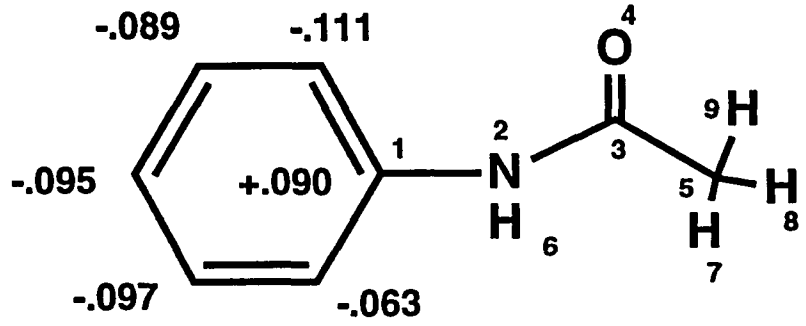
Zou, Johnson, and Tsai[17] developed a set of graph-theoretic transforms for predicting chemical and metabolic reactions. In fact, they chose to develop their transforms based upon one of the reactions, electrophilic aromatic substitution (EAS), reported in this dissertation. The transforms are much like substructures which are required for the reaction

to occur. Some of the features or substructures that they found to be useful in predicting EAS reactions were: (a) halo-substituted aromatic, (b) aromatic ring with a charged substituent, (c) branching alpha to an aromatic ring, and (d) heteroatom with non-bonded electrons attached to aromatic ring. These transforms were used as ten inputs, indicating the presence or absence in that reactant of each of the ten transforms. An example for acetanilide is shown in Figure 19. Note that the input vector in Figure 19 is all zeroes. Acetanilide was one of the compounds that did contain any of the ten graph-theoretic transforms.

Randic Indices

Randic has developed a set of topological indices or invariants that have proven useful for predicting properties.[154] These topological indices are calculated directly from the connection table. Six values, which are the weighted paths P1, P2, P3, P4, P5, and P6 (Figure 19), were calculated for each molecule[155] (Data kindly provided by Randic, Nov 14, 1990). Unfortunately, the Randic procedure cannot calculate these indices for charged molecules, so only 27 of the 35 training compounds were present in this set and only 10 of 13 test compounds.

Other Representations



Charge Vector

+.090, -.063, -.095, -.097, -.089, -.111

Graph Transforms 0 0 0 0 0 0 0 0 0 0 0 0

Randic Indices

P1	P2	P3	P4	P5	P6
1.7583	0.3877	0.723	0.0149	0.0026	0.0002

Figure 19. Other Representations for Acetanilide.

CNN Prediction of Electrophilic Aromatic Substitution Reactions

Selection and Background of the Problem

Electrophilic aromatic substitution (EAS) involves the substitutionof a hydrogen atom on an aromatic ring, by an electrophile such as nitric acid, to give ortho, meta or para substituted products which correspond, as shown in Figure 20, to the attachment of the electrophile at positions 2, 3, and 4. Since EAS is well studied and since there is a reasonable amount of data available for this reaction, it was chosen as a model for investigating neural network applications to chemical reaction prediction. The ratio of isomers formed depends mainly on the nature of the substituent X and to a lesser degree on the electrophile Y. Substituents may be divided into two classes, ortho-para directors, which tend to be electron donors, and meta directors, which tend to be electron acceptors. Resonance effects of the substituent can reinforce or oppose these inductive effects and thereby affect the product ratio. Steric hindrance by large substituents also can affect the reaction by blocking the adjacent ortho positions.

Figure 20 shows the mechanism for electrophilic substitution. The electrophile first adds to one of the aromatic carbons to give a carbocation which has the positive charge delocalized over three carbon atoms. Then the proton is eliminated to give the substituted product. The reaction is

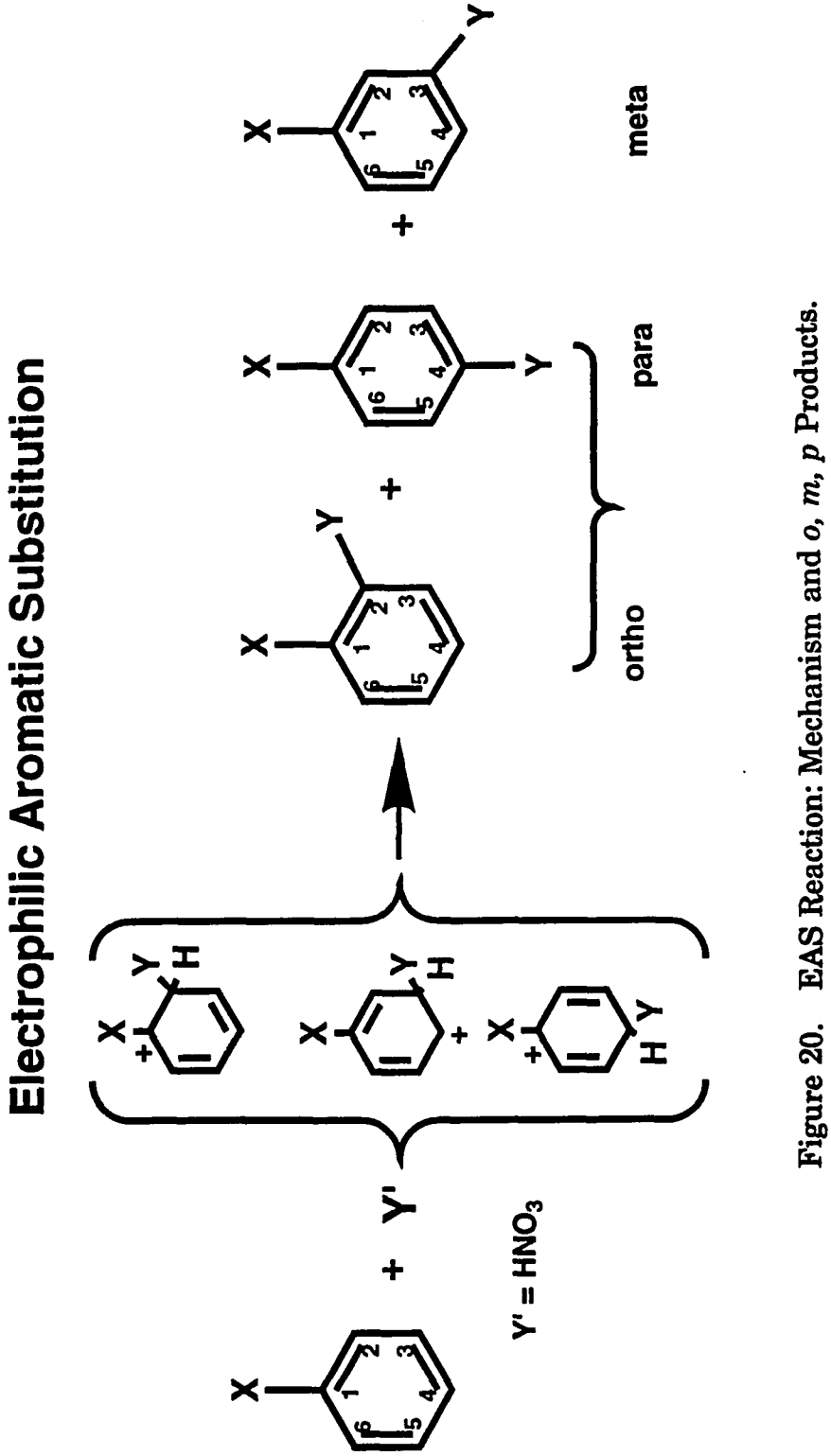


Figure 20. EAS Reaction: Mechanism and *o*, *m*, *p* Products.

generally irreversible and kinetically controlled, with the first or addition step being rate-determining. Since the aromatic ring contributes the two electrons needed to form the bond to the incoming electrophile, the orientation of the reaction, when there is already a substituent on the ring, depends on the relative electron availability at the unsubstituted sites.

Organic chemistry students are taught rules which they can use to predict the major products in EAS reactions, by considering the electron withdrawing or attracting character of the substituents in mono-substituted benzenes. These rules enable chemists to predict the major products in EAS by simply looking at the substituent and classifying it as either ortho-para or meta directing. Generally no attempt is made to predict percentages of each product formed. It is usually implied that the products formed are only ortho-para or only meta, but upon examining the chemical literature it is apparent that many intermediate cases exist which give mixtures of all three products. In these cases, it would be useful to have a prediction of the ortho-para to meta product ratio.

In the investigation of the applicability of neural network methods in general and in chemistry in particular, two issues must be addressed at the outset: the choice of the network paradigm and the choice of the data representation. A back propagation network was chosen as the network architecture based on its pattern mapping ability (*vide supra*). The data representation question is a key theme of the present work. The correct

data representation is crucial to the predictive ability of a network. As no prior knowledge of chemistry is encoded in the network, it must extract relevant features from the training examples in order to be capable of making predictions. Consequently, we investigated the representations in the previous section for their suitability for encoding the relevant molecular structure information into numeric form for CNN applications in chemistry.

Methods

Data

Product ratios for electrophilic substitution reactions of mono-substituted benzenes taken from the literature were used to train and test the neural network.[156-161] The same electrophile, nitric acid, was used in all of the reactions except for phenol and phenoxide, where t-butyl hypochlorite was used. The compounds were split into a training set of 32 compounds and a test set of 13 compounds, shown in Figure 21. The test set was chosen to be representative of both the types of substituents and the range of product ratios. Nine of the thirteen test compounds were from the literature, the other four were constructed by making trivial structural changes in four of the training set compounds to produce homologs, for example, methyl benzoate was constructed from ethyl benzoate.

	Substituent	% Meta		Substituent	% Meta		Substituent	% Meta
Ph	$-NH_2$	0		$-C\equiv C\cdot COOH$	8	**	$-CH_2N^+H_2CH_3$	60
	$-CH=CHCOOH$	0	**	$-CH_2NH_2$	10		$-CCl_3$	64
	$-F$	0		$-CH_2OCH_3$	12		$-COOCH_2CH_3$	68
	$-CHMe_2$	0		$-CH_2SO_2O^-$	14	**	$-COOCH_3$	68
	$-O^-$	0	*	$-CH_2CN$	14		$-N^+H_2CH_3$	70
*	$-OH$	0		$-CH_2Cl$	16	*	$-CONH_2$	70
*	$-CH_2CH_3$	0		$-CH_2F$	18		$-COCH_3$	72
	$-Cl$	1	**	$-CH_2COOH$	22		$-CHO$	72
*	$-Br$	1		$-CH_2SO_2NH_2$	31		$-N^+HMe_2$	78
	$-I$	2		$-CHCL_2$	34		$-COOH$	80
	$-NHCOCH_3$	2	*	$-SiMe_3$	40		$-CN$	82
	$-OCH_3$	2	*	$-NH_3+$	42		$-NO_2$	93
	$-CH=CHNO_2$	2		$-CH_2SO_2Cl$	51	*	$-S^+Me_2$	95
	$-CH_2CH_2OCH_3$	3	*	$-CH_2NO_2$	56		$-CF_3$	100
	$-CH_3$	4		$-SO_3^-$	60		$-SO_2CH_3$	100

*Test Set **Test Set Homologs [t] Data from references 156-161

Figure 21. EAS Literature Ranked by Percent of Meta Product.[156-161]

Connection Table Representation

A compact connection table, shown for the acetamido substituent in Figure 18, was developed as an input representation for the neural net. Since the aromatic ring is the same for all of the reactants, only information on the substituents is included. Atoms, excluding hydrogens, were numbered from the aromatic ring out in a breadth first manner, with atom number 1 being the atom in the aromatic ring where the substituent was attached. The connection table was kept as concise as possible while preserving the connectivity information by citing only the bonds from higher numbered atoms to lower numbered atoms. There is one row for each non-hydrogen atom in the substituent. The first column contains the atomic number of the higher numbered atom in a bonded pair. The atomic number provides a chemically relevant means of differentiating atom types with a single number since, as noted in the previous section, a neural network requires numeric values for its input data. Bonds are specified with the higher numbered atom in the second column and the lower numbered atom in the third column. The fourth column indicates the bond order: 1 for single bonds, 2 for double bonds, and 3 for triple bonds. The charge on the atom, either -1, 0 or +1, is placed in the fifth column. Specifying the charge on the atom allows hydrogen atoms to be ignored by indicating differences from the normal valence with the charge. This

results in a very compact 5 X 5 connection table. For substituents with less than 5 atoms, the remaining rows are filled out with zeroes to ensure that all the input patterns have the same number of values.

Charge Vector Representation

Instead of representing the substituent directly, the effect of the substituent on the aromatic ring was represented by the charge at each carbon atom on the ring (Figure 19). The carbons were numbered starting at the atom where the substituent was attached and proceeding clockwise around the ring. MOPAC,[162] a semi-empirical quantum mechanics program, was used to calculate the Mulliken partial charges on each of the atoms. The data from MOPAC was used to create a vector of 6 charge values, one for each carbon atom in the aromatic ring, which was used to train and test the network.

Network Simulator

The ANSIM neural network simulator program was used for these EAS reaction prediction studies. ANSIM is a multi-paradigm, general purpose network simulator, formerly available from Science Applications International Corporation, (SAIC), San Diego, CA. A nice feature of ANSIM is that it allows displaying the layers as if they were two-dimensional. In this case, the 5 x 5 connection table could be displayed using the program's

graphics as a 5 by 5 grid of units. This made the structure of the input layer appear to parallel the construction of the EAS connection table, even though the inputs were really in one 25-component input vector. ANSIM version 2.30, which requires Microsoft Windows 2.0 or later, was run on an IBM-compatible 20 MHz 80386 microcomputer equipped with an 80387 math coprocessor. The math coprocessor is necessary since the neural network program makes extensive use of floating point calculations. SAIC has stopped selling and supporting ANSIM and now only provides neural network consultation services, so subsequent work was done using different CNN simulator programs. ANSIM has internal data scaling features, which were used to scale the data into the ANSIM-required range of -0.5 to +0.5. The ANSIM rescaling feature was used to re-scale the data from the network back to the range 0% to 100% used for the reaction product.

Network Configuration

The best network paradigm and configuration for a particular problem must be found empirically. However, some decisions can be made on the basis of the characteristics of the various network types. A three layer feed-forward backpropagation (BP) network was chosen because of its pattern mapping properties.[26] The BP net constructs an internal representation of the task, in this case, predicting the products of the EAS reaction, which can be applied to new reactants not in the training set. The

network which used the connection table as input (Figure 22) had 25 processing units in the input layer, a hidden layer of 5 units, and an output layer of two units. The 25 input units correspond to the 5 x 5 connection table representation. Five hidden units gave the best network performance. A larger number of hidden units caused the net to fit the input examples very well but gave poor generalization for the test set of reactants. Fewer hidden units gave poor predictions for both the training set and the test set. The two output units represent the fraction of ortho-para product and the fraction of meta product produced in the reaction. When the partial charges on the aromatic ring were used as input, six input units were needed. Ten hidden units were found to give the best predictions in this case. Two output units were also used for the charge vector network.

Network Training and Testing

Training a network is an iterative process that involves repeatedly presenting the training examples, adjusting the weights according to the learning rule, and modifying the learning parameters. Thirty-two of the reactants were used to train the network and the remaining 13 compounds were kept for testing. Starting with a set of connection strengths randomly chosen in the range [-0.3 to +0.3], the weights were changed by the supervised BP algorithm to minimize the root mean squared (RMS) error between the net's predicted output values and the experimental ones.

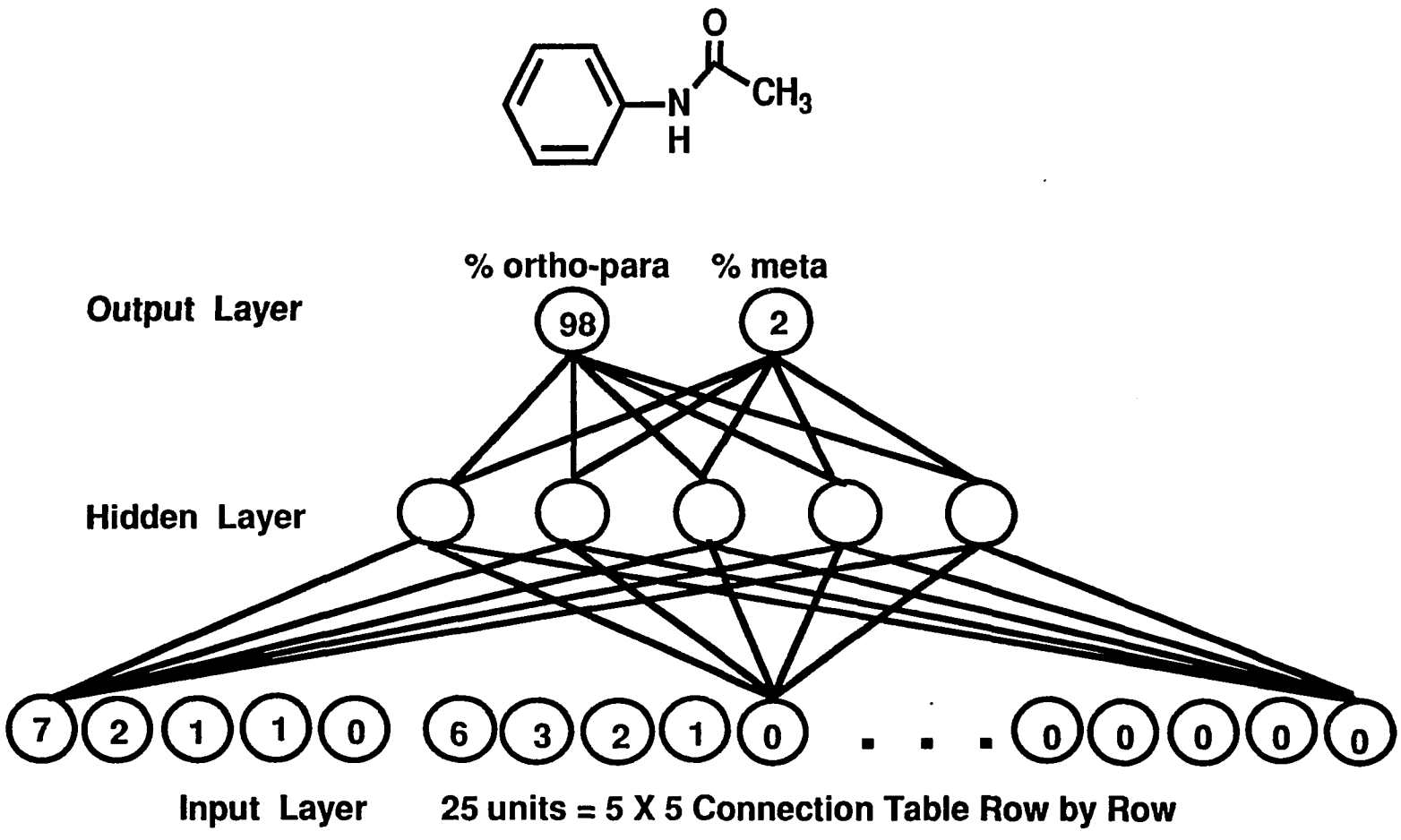


Figure 22. Network Used to Predict EAS Reactions From Connection Table.

Typically it required presenting the entire set of training examples for more than 100,000 repetitions to obtain a set of weights that reduced the RMS error to less than 0.05. The learning rate η, which is the proportion of the error that is used to adjust the weights, was reduced from 0.1 to 0.005 over the course of training.

It is not possible to determine that the set of weights obtained by the BP procedure represents the global minimum on the error surface for the system. Two approaches were combined in the training regimen to attempt to avoid getting trapped in a local minimum. In the first approach, random values were added to the weights after the total (RMS) error had stabilized. This allows the network to explore other parts of the error surface. Training was continued until the RMS error reached another plateau. This process of "damaging" the weights by adding a random component and retraining the net was repeated several times until no further improvement in the RMS error was obtained. The other technique to improve network performance was to add random noise to the input patterns and to let the noise gradually decay to zero. This process can make the network more robust to the effects of noisy data and may also serve to help avoid some local minima. Both of these approaches were repeated several times during the training process to obtain the best network predictions.

Unlike the training process, which can take a very long time, testing the network can be done rapidly in a single pass through the data. In the

testing phase a set of input data is presented to the network without providing the target output values and without changing the weights. The output values computed by the net are determined by the set of weights that developed during the training process. Network predictions for both the training and test sets were obtained separately so that the ability of the net to learn how the patterns of input features correlate with the product ratios in the training data could be measured as well as the ability of the net to make generalizations about new cases on which it had not been trained.

Comparison With Non-Neural Net Prediction Methods

The resulting neural networks were compared to predictions made by the widely used chemical expert system CAMEO[163] and by three experienced synthetic organic chemists. CAMEO applies mechanistic rules to make predictions about a broad range of organic reactions, of which electrophilic substitution is only one. CAMEO, like the organic chemists it emulates, predicts either ortho-para products or meta products but does not quantitate those predictions. So CAMEO predicts either 0% meta or 100% meta products. Reactions which yield less than 20% or more than 80% of meta product will have small errors while reactions which form between 20% and 80% meta product will have correspondingly larger errors. The three organic chemists were given the percentages of ortho, para, and meta

products for the 32 training set reactants and asked to predict the percent of the three products for the 13 unknowns from the test set. Based on the range of predictions by the chemists and also on the variation in product ratios due to experimental conditions a prediction was deemed correct if it was within 20% of the average of the experimental values. The accuracy of the prediction for the fraction of meta product was used as a basis for determining the overall performance of a network.

Results For Connection Table and Charge Vector CNNs

Figure 23 shows the magnitude of the errors in the network predictions for the amount of meta product for the training set. The vertical axis displays the 32 training compounds and the horizontal bars correspond to the size of the errors between the predicted percent of meta product and the experimental value. The best connection table network, with a 25 unit input layer, a 5 unit hidden layer, and 2 unit output layer was trained to a total RMS error of 0.022. All 32 of the training compounds were correctly predicted with an average error per compound of 0.3%. The best charge vector network, with 6 input units, 10 hidden units, and 2 output units (final RMS error 0.078) also correctly predicted 32 of 32 training compounds but with an average error of 5.2%. CAMEO predicted only 22/32 of the training set correctly, with an average error of 18%. CAMEO is at a disadvantage in this type of comparison because it predicted

all meta or no meta product, while the CNNs gave quantitative predictions.

The more difficult problem is that of making predictions on the unknowns in the test set. As shown in Figure 24, the connection table network correctly predicted 10/13 reactions of the test compounds with an average error of 12%. The charge vector network predicted only 8/13 correctly with an average error of 20%. CAMEO, partly due to the all or nothing nature of its predictions, was correct in 7/13 cases. The same 13 reactants were given to three synthetic organic chemists who made correct predictions for 10/13 reactants with an average error of 15%.

Of particular interest is the reactant with a trimethylsilyl (TMS) group attached to the aromatic ring, which was reported experimentally to yield 40% of meta product.[157] None of the training examples contained any silicon atoms so the network was forced to make a generalization for this substituent. The connection table network gave a reasonable prediction of 63% meta even though it had not been trained on any examples of silyl substituents. The charge vector network gave a very poor prediction of 0% meta while CAMEO predicted 33% meta. The compound with a TMS substituent was the only one of the 45 in the data set for which CAMEO predicted all 3 products: ortho, para, and meta. It appears that CAMEO has a rule that handles a TMS group as an exception. The three chemists did not do as well as the network on this reactant. They predicted that the TMS substituent would yield only 2% of meta substitution. When asked

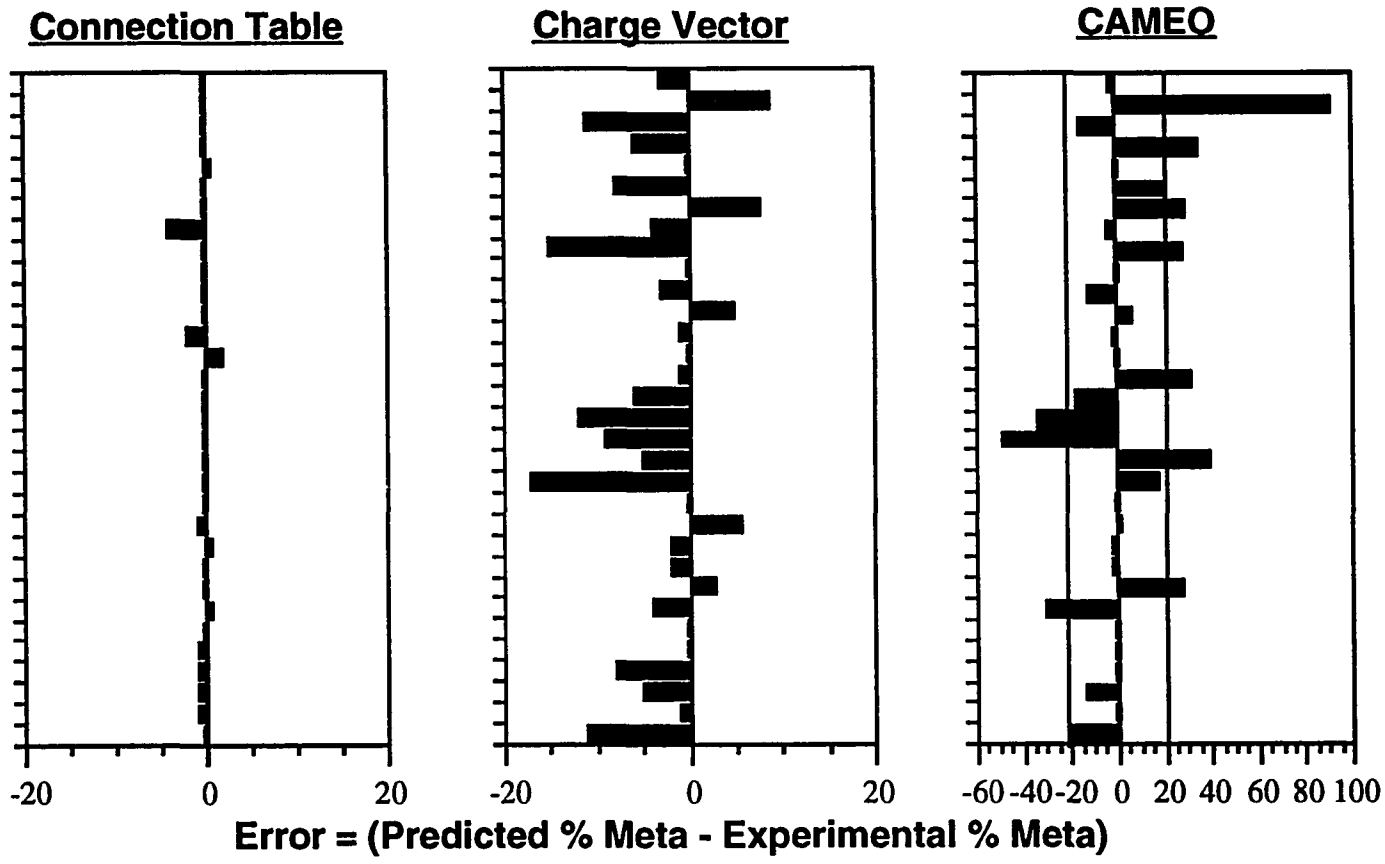


Figure 23. EAS Reaction. CNN Results on Training Set Reactions.

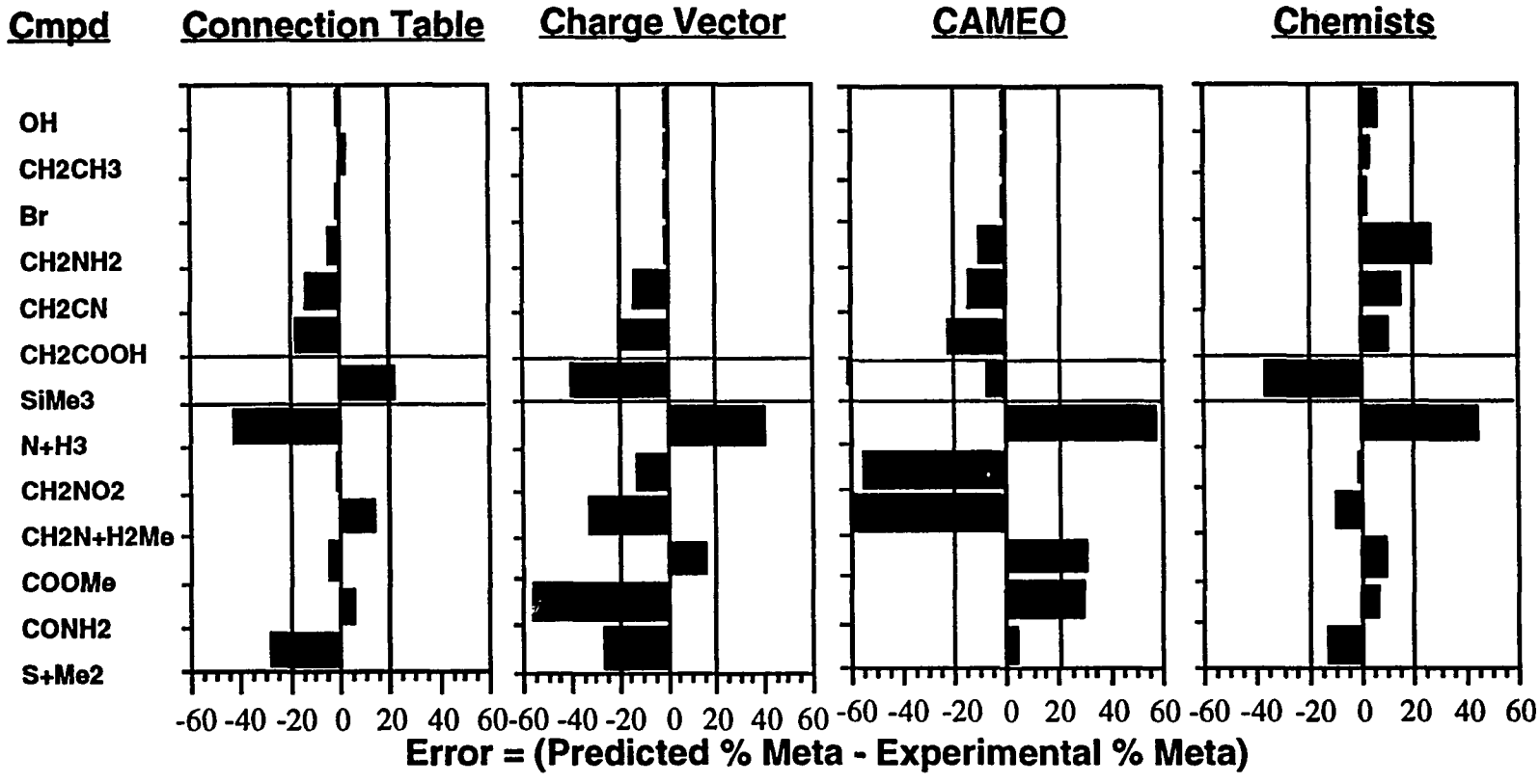


Figure 24. EAS Reaction. CNN Results on Test Set Reactions.

why they made this prediction the chemists said that a silicon substituent was not part of their "training set" and so they used methyl or t-butyl as a model. These both produce less than 10% of meta compound, so neither is a good model for the reaction with a TMS substituent.

The two networks compare well with the CAMEO expert system program and with the chemists in the number of correct predictions. CAMEO predicted 69% of the training set correctly and 54% of the test cases, to within a 20% error. The connection table network was somewhat better than either the charge vector network or the CAMEO program and performed about as well as the chemists. The connection table network predicted 100% of the training set correctly and 77% of the unknowns while the charge vector was correct for 100% of the training set and 62% of the test set. The chemists correctly predicted 77% of the test set.

Results for Graph Transforms and Randic Indices

Graph Transforms

Zou, Johnson and Tsai[17] provided descriptors for their graph transforms for a set of 54 monosubstituted benzene compounds shown in Table 5. They determined the presence (1) or absence (0) of ten transforms. The N-Net 210 neural network simulator program,[46] run on a 80386/80387 IBM PS/2 Model 70 microcomputer was used to develop a CNN for

predicting electrophilic aromatic substitution reactions of these 54 compounds. N-Net contains an unsupervised CNN for clustering as well as backpropagation and functional link networks.[47] A functional link net is a variation of the perceptron where the input layer is enhanced with the cross-products of the inputs. It is a feedforward network trained using the delta rule.

After several attempts to train functional link networks and backpropagation networks, none of which would converge to give good results on the training set, the unsupervised clustering network of N-Net was used to cluster the data. The results of the clustering, along with the result of the functional link network predictions are shown in Table 5. Several facts emerge from this clustering that explain the difficulty the network had in even learning these EAS reactions when the graph-theoretic transforms were used as the input representation. Seventeen of the compounds had input vectors of all zeroes. This indicates that those seventeen compounds had *none* of the transforms as features. Several other groups of compounds had identical input vectors, but within the group, the range of meta products experimentally observed varied widely. In one case, the substituents $-CH_2OCH_3$, $-CH_2F$, $-CH_2NO_2$, and $-CH_2N^+(CH_3)_3$ all had inputs of 0010000000, but the amounts of meta product ranged from 88% down to 12%. For this group, the network predicted 56% meta product for all four compounds, in effect averaging the amounts of product for the

Table 5

Graph Theoretic Transforms for EAS Reaction and CNN Results

Substituent	Graph-Theoretic Transforms										Exp. % meta	CNN Pred % meta
-NHPh	0	0	0	0	0	0	0	0	0	0	100	97
-NH2	0	0	0	0	0	0	0	0	0	0	100	97
-Ph	0	0	0	0	0	0	0	0	0	0	100	97
-F	0	0	0	0	0	0	0	0	0	0	100	97
-CH(CH3)2	0	0	0	0	0	0	0	0	0	0	100	97
-OH	0	0	0	0	0	0	0	0	0	0	100	97
-O-	0	0	0	0	0	0	0	0	0	0	100	97
-CH2CH3	0	0	0	0	0	0	0	0	0	0	100	97
-Cl	0	0	0	0	0	0	0	0	0	0	99	97
-Br	0	0	0	0	0	0	0	0	0	0	99	97
-NHCOCH3	0	0	0	0	0	0	0	0	0	0	98	97
-OCH3	0	0	0	0	0	0	0	0	0	0	98	97
-CH2CH2OCH3	0	0	0	0	0	0	0	0	0	0	97	97
-CH3	0	0	0	0	0	0	0	0	0	0	96	97
-C(CH3)3	0	0	0	0	0	0	0	0	0	0	92	97
-CH2COOCH2CH3	0	0	0	0	0	0	0	0	0	0	89	97
-CH2CN	0	0	0	0	0	0	0	0	0	0	86	97
-CH=CHCOOH	0	0	0	0	0	0	1	0	0	0	100	96
-CH=CHNO2	0	0	0	0	0	0	1	0	0	0	98	96
-CH=CHSO2Cl	0	0	0	0	0	0	1	0	0	0	98	96
-CCCOOCH2CH3	0	0	0	0	0	0	1	0	0	0	94	96
-CCCOOH	0	0	0	0	0	0	1	0	0	0	92	96
-CH2OCH3	0	0	1	0	0	0	0	0	0	0	88	56
-CH2F	0	0	1	0	0	0	0	0	0	0	82	56
-CH2NO2	0	0	1	0	0	0	0	0	0	0	45	56
-CH2N+(CH3)3	0	0	1	0	0	0	0	0	0	0	12	56
-CH2SO2O-	0	0	1	0	0	0	0	0	0	1	86	71
-CH2Cl	0	0	1	0	0	0	0	0	0	1	84	71
-CH2P+(CH3)3	0	0	1	0	0	0	0	0	0	1	81	71
-CH2SO2NH2	0	0	1	0	0	0	0	0	0	1	69	71
-CH2SO2OCH3	0	0	1	0	0	0	0	0	0	1	68	71
-CHCl2	0	0	1	0	0	0	0	0	0	1	66	71
-CH2SO2Cl	0	0	1	0	0	0	0	0	0	1	49	71
-SO3-	0	1	0	0	0	0	0	0	0	0	61	60

Table 5--Continued

Substituent	Graph-Theoretic Transforms										Exp % meta	CNN Pred % meta
-N+H3	1	0	0	1	0	0	0	0	0	0	58	57
-Si(CH3)3	0	1	1	0	1	0	1	1	1	1	40	39
-CCl3	0	0	1	0	1	0	0	1	0	1	36	35
-COOCH2CH3	0	0	1	0	1	0	1	0	0	0	32	23
-CONH2	0	0	1	0	1	0	1	0	0	0	30	23
-COCH3	0	0	1	0	1	0	1	0	0	0	28	23
-COOH	0	0	1	0	1	0	1	0	0	0	20	23
-NO2	0	0	1	0	1	0	1	0	0	0	7	23
-NH2+CH3	1	0	0	0	0	1	0	0	0	0	30	17
-NH+(CH3)2	1	0	0	0	0	1	0	0	0	0	22	17
-N+(CH3)3	1	0	0	0	0	1	0	0	0	0	0	17
-CHO	0	0	1	0	0	0	1	0	0	0	28	22
-CN	0	0	1	0	0	0	1	0	0	0	18	22
-I+Ph	1	1	0	0	0	1	0	0	0	0	18	7
-S+(CH3)2	1	1	0	0	0	1	0	0	0	0	5	7
-P+(CH3)3	1	1	0	0	0	1	0	0	0	0	0	7
-SO2CH3	0	1	1	0	1	0	1	1	0	0	0	0
-CF3	0	0	1	0	1	0	0	1	0	0	0	0

compounds that had the same input. As in the case of the boiling point prediction in the representation section of this chapter, the representation did not sufficiently distinguish between dissimilar compounds. Because of the inherent ambiguity in the transform representation, where the same input maps to several different output values, the transforms were not considered further as an input representation. The fact that the clustering indicated where there were problems with the data points out the value in carefully analyzing the data before applying CNN mapping methods.

Randic Indices

Table 6 shows the Randic weighted path indices for the training and testing sets of aromatic compounds used to train a NeuralWorks Professional II Plus EDBD network. The network had 6 inputs, 5 hidden units in a single hidden layer, and 2 outputs (%ortho + %para products and % meta product). The network was trained for 100,000 training cycles, until the error of the training set did not improve. As noted in the representation section, only the uncharged compounds were able to have their paths calculated, so there were 27 compounds in the training set. The trained network was then tested on the test set of 10 compounds. Three of the test compounds that were used in the test set for the connection table and charge vector representations were charged, and thus their paths could not be calculated by the Randic method. The results on the training set showed that 25 of 27 (92.5%) training compounds were predicted to within the 20% error criteria. The average error on the training set was 3.4%. For the 10 test compounds, only 5 or 50% had their EAS reaction product ratios correctly predicted. The average error of these test compounds, was 22.9%, significantly worse than the connection table or charge vector representations.

Table 6

Randic Indices and Results for EAS Reactions

Compound	Randic Indices						Exp % Meta	Pred % Meta	Error
	P1	P2	P3	P4	P5	P6			
-NH2	1.3434	.3176	.0587	.0107	.0020	.0001	.000	.0105	-.0105
-CH=CHCOOH	1.9664	.4262	.0874	.0168	.0037	.0003	.000	.0088	-.0088
-F	1.3248	.3082	.0572	.0105	.0019	.0001	.000	.0071	-.0071
-CHMe2	1.5853	.3704	.0789	.0141	.0025	.0002	.000	-.0010	.0010
-Cl	1.2845	.2881	.0538	.0099	.0018	.0000	.009	.0041	.0049
-I	1.2347	.2631	.0497	.0092	.0017	.0000	.018	.0023	.0158
-NHCOCH3	1.7583	.3877	.0723	.0149	.0026	.0002	.020	.0247	-.0047
-OCH3	1.4443	.3271	.0666	.0121	.0022	.0001	.020	.0774	-.0574
-CH=CHNO2	1.9495	.4229	.0867	.0166	.0033	.0003	.020	.0512	-.0312
-CH2CH2OCH	1.6998	.3670	.0739	.0141	.0026	.0002	.030	.0324	-.0024
-CH3	1.3556	.3237	.0597	.0109	.0020	.0001	.040	.0172	.0228
-CC-COOH	2.0302	.4373	.0948	.0183	.0037	.0004	.080	.0507	.0293
-CH2OCH3	1.5748	.3507	.0710	.0136	.0024	.0001	.120	.4133	-.2933
-CH2Cl	1.4268	.3323	.0661	.0120	.0022	.0001	.155	.0941	.0609
-CH2F	1.4552	.3373	.0686	.0124	.0022	.0001	.180	.1826	-.0026
-CH2SO2NH2	1.7634	.3954	.0700	.0142	.0025	.0002	.314	.3174	-.0034
-CHCl2	1.4905	.3449	.0710	.0128	.0023	.0001	.338	.3270	.0110
-CH2SO2Cl	1.7388	.3867	.0696	.0140	.0025	.0002	.508	.5058	.0022
-CCl3	1.5484	.3602	.0748	.0134	.0024	.0001	.645	.6478	-.0027
-COOCH2CH3	1.8702	.4018	.0864	.0159	.0028	.0002	.684	.7084	-.0244
-COCH3	1.5748	.3507	.0710	.0136	.0024	.0001	.720	.4133	.3067
-CHO	1.5441	.3476	.0754	.0135	.0024	.0001	.720	.7177	.0023
-COOH	1.6302	.3776	.0813	.0145	.0026	.0002	.802	.8105	-.0085
-CN	1.6248	.3557	.0809	.0144	.0026	.0002	.820	.8161	.0039
-NO2	1.6109	.3703	.0787	.0141	.0025	.0002	.932	.9421	-.0101
-CF3	1.6248	.3857	.0809	.0144	.0026	.0002	1.000	1.002	-.0020
-SO2CH3	1.6235	.3597	.0715	.0129	.0023	.0001	1.000	.9970	.0030
-OH	1.3333	.3125	.0579	.0106	.0019	.0001	.000	.0089	-.0089
-CH2CH2	1.4771	.3412	.0705	.0127	.0023	.0001	.000	.2573	-.2573
-Br	1.2500	.2708	.0509	.0095	.0017	.0000	.012	.0027	.0093
-CH2NH2	1.4684	.3397	.0698	.0126	.0023	.0001	.100	.2277	-.1277
-CH2CN	1.7552	.3709	.0745	.0154	.0027	.0002	.140	-.1242	.2642
-CH2COO	1.7607	.3925	.0746	.0154	.0027	.0002	.220	-.1086	.3286
-SiMe3	1.5518	.3434	.0696	.0126	.0023	.0001	.398	.4681	-.0701
-CH2NO2	1.7435	.3892	.0737	.0151	.0027	.0002	.550	-.1197	.6697
-COOCH3	1.7443	.3876	.0852	.0154	.0027	.0002	.684	.2249	.4591
-CONH2	1.6366	.3799	.0818	.0146	.0026	.0002	.700	.7998	-.0998

Summary

Results comparing several types of representation of chemical information, connection tables, charge vectors, graph-theoretic transforms, and Randic indices, indicate that the more general connection table representation provides better predictions. The charge vector focuses on a specific aspect of the reactants, the distribution of charge in the aromatic ring, which is only one of the factors in the relative reactivity of the ortho, para, and meta positions. The graph-theoretic transforms, which worked for a non-CNN application,[17] did not distinguish compounds in the CNN application that had very different product ratios in the EAS reaction. The Randic indices also had limitations and did not give good generalization. The more general connection table representation is a richer source of information for the neural network. The connection table provides a means of comparing the atoms on the basis of atomic number, the number and types of connections, and other properties such as charge or number of hydrogens. The ability of the network to make reasonably good quantitative predictions is an advantage compared to the expert system which only predicts the major tendency of the reaction.

The ability of neural networks to predict a chemical reaction as well as chemists or an expert system has been demonstrated for the EAS reaction. The neural net was able to learn by example to make predictions

for cases on which it had not been trained. Quantitative predictions were made by the network in contrast to the expert system which made qualitative predictions. Also the neural network did not require formulating rules about reactivity in order to make useful predictions but was able to form an internal model of the reaction by extracting information directly from examples of the reaction. The method used for representing the chemical information for the neural network is a major factor in determining the predictive ability of the network. A more descriptive, less specific representation derived from connectivity information gave better results than the representation that employed specific charge information.

CHAPTER IV

EXTENSION OF CNN PREDICTION TO OTHER REACTIONS

Introduction

In Chapter III we reported on the prediction of ortho-para/meta product ratios in electrophilic aromatic substitution reactions using a backpropagation neural network based on a simplified connection-table representation of the reactants. Although that method was very successful, it was limited to the particular class of reactions under study. In order to deal with a wider variety of chemical reactions using neural networks, a more general representation is required. In this chapter we report on the development of a suitable representation based upon a modification of the BE-matrix used by Dugundji and Ugi to describe chemical reactions.[164] A backpropagation neural network using this representation was examined for its ability to predict three different types of chemical reactions: (1) Markovnikov addition of hydrogen halides to alkenes, (2) Diels-Alder and retro-Diels-Alder cycloadditions, and (3) Saytzeff E1 elimination to form olefins. For all three of these types of reactions chemists have developed rules to predict which of the two or more regioisomers would be formed. What the network must then do is develop an internal representation of the

rules in order to predict the regiochemistry of the reactions.

The simplest case involved a test of whether a neural net could "discover" Markovnikov's rule for the addition of hydrogen halides, HX, to unsymmetrical olefins. Markovnikov's rule states that the hydrogen atom of HX adds to that carbon which bears the greater number of hydrogens.[165] Figure 25 shows the addition of HBr to simple alkenes. The second reaction involved the prediction of the regiochemistry of Diels-Alder cycloaddition reactions of unsymmetric dienes with unsymmetric dienophiles.[166] As depicted in Figure 26, 1-substituted dienes react with substituted dienophiles to give 80 to 100% of the "ortho" isomer while 2-substituted dienes give predominantly "para" isomer.[167] The network was also tested for its ability to predict the disconnections required in the corresponding retro-Diels-Alder reaction. Beta elimination of HX to form alkenes, shown in Figure 27, was chosen as the third class of reactions on which to apply the neural network method. The regiochemistry of the double bond in the elimination product can be predicted by Saytzeff's rule, namely, that a compound which can eliminate in two ways to give two different olefins normally gives as the major product the more highly conjugated olefin.[168]

Ugi B-E Matrix Representation of Reactions

In their elegant work, Dugundji and Ugi[164] treated constitutional

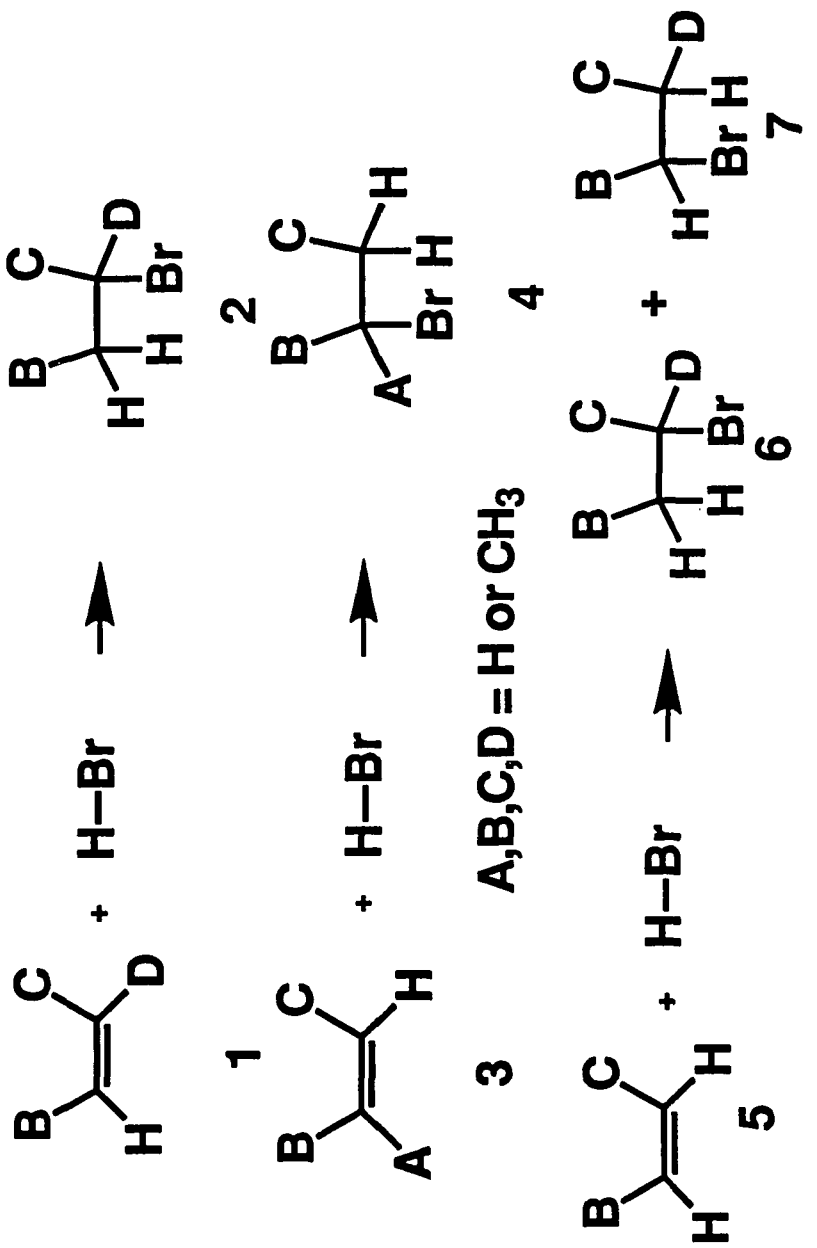


Figure 25. Markovnikov Addition of Hydrogen Bromide to Alkenes.

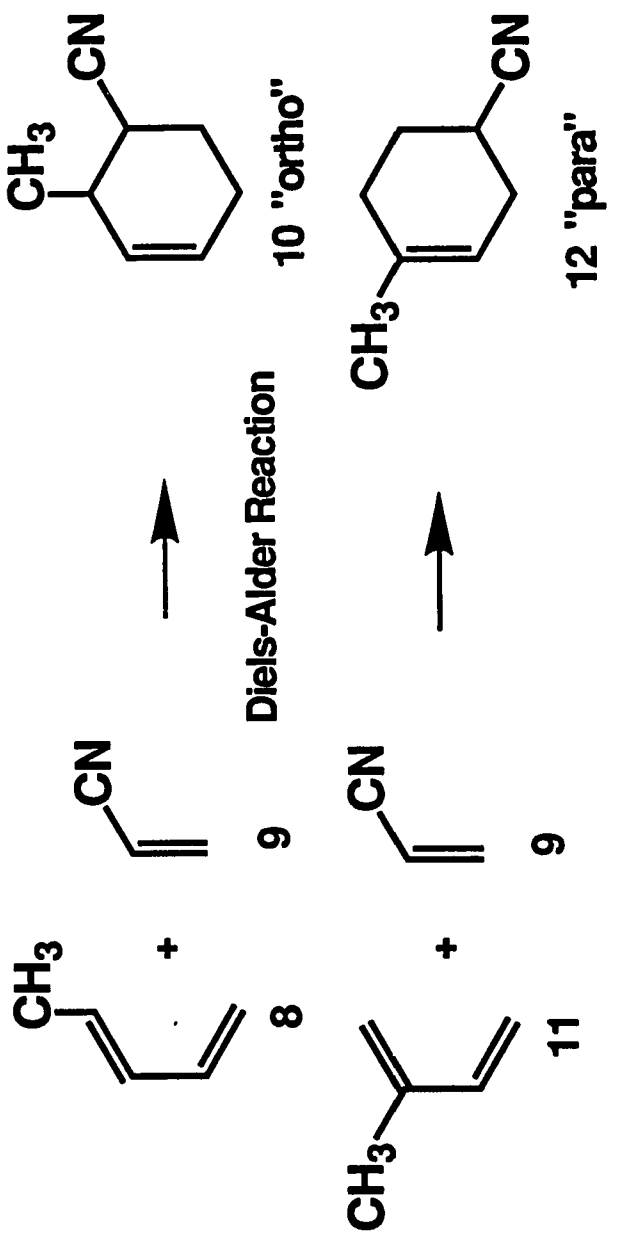


Figure 26. Diels-Alder and Retro Diels-Alder Reactions.

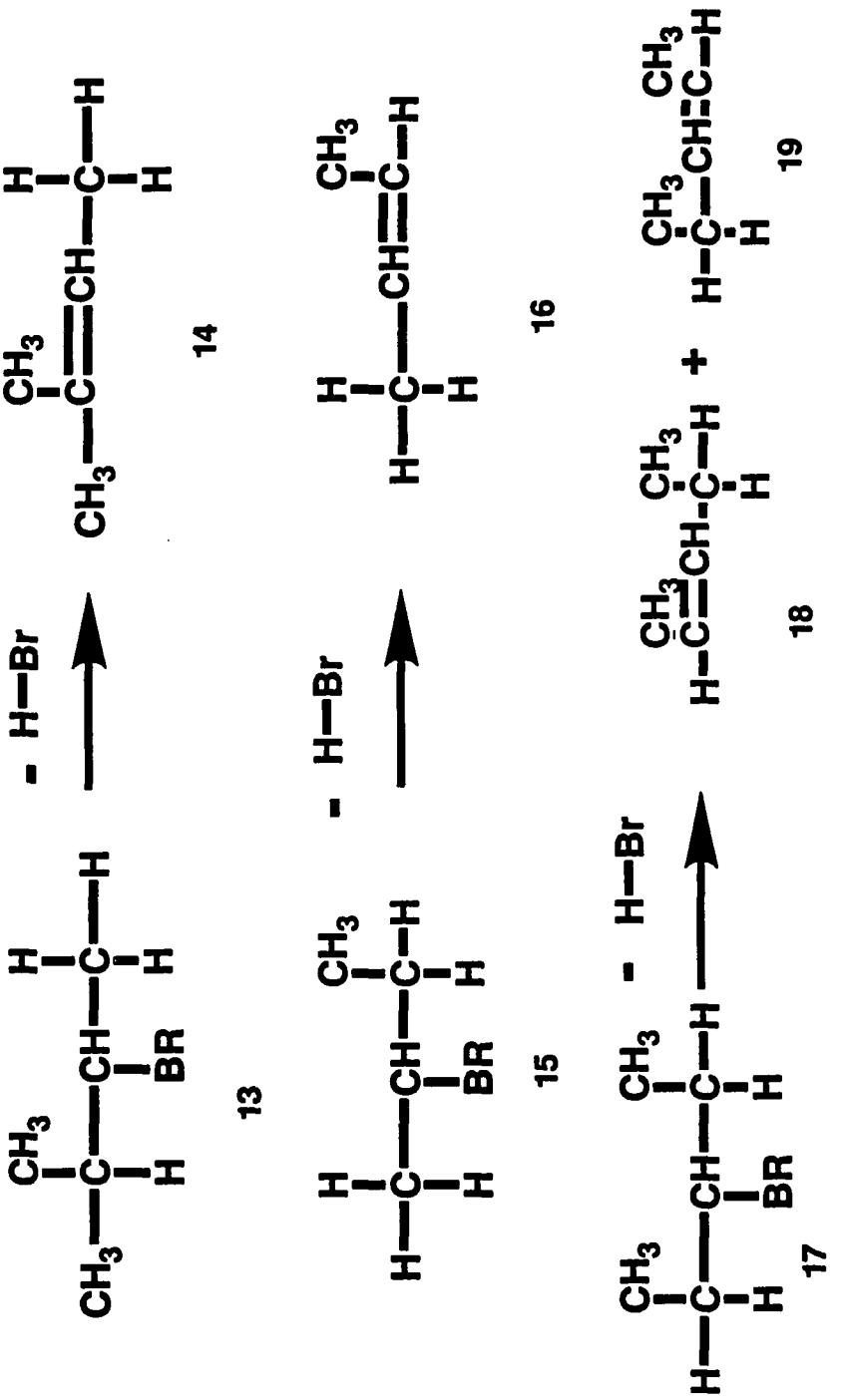
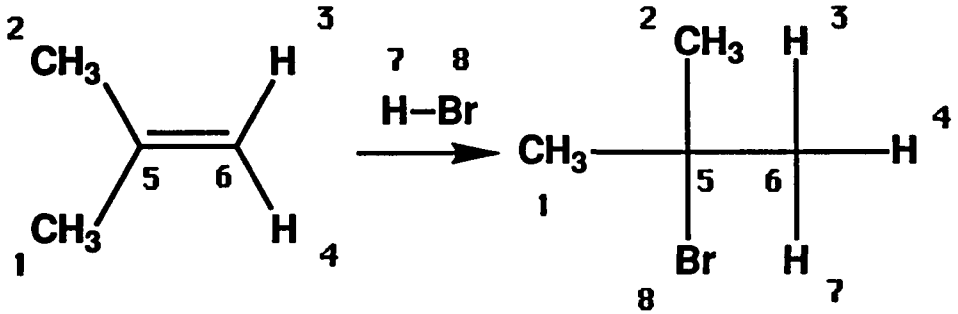


Figure 27. E1 (Saytzeff's Rule) Elimination to Form Alkenes.

chemistry as isomerizations of ensembles of molecules. In their model, chemical reactions are defined as the redistribution of valence electrons from the ensemble of molecules that represent the reactants to the isomeric ensemble of molecules that represent the products. The set of covalent bonds corresponding to an ensemble of molecules is conveniently represented as an $n \times n$ BE-matrix, where n is the total number of atoms in the ensemble of molecules. The BE-matrix is a symmetric atom-connectivity matrix where each off-diagonal element b_{ij} $(i \neq j)$ indicates the formal bond order between an indexed pair of atoms A_i and A_j, and the diagonal elements b_{kk} contain the number of unshared valence electrons in atom A_k.

The storage of information on the atom types outside of the BE-matrix allows the Dugundji-Ugi approach to specify successive levels of generality for reactions.[169] However, in order for a neural network to differentiate compounds with the same connectivity pattern but different atomic compositions, the atom-type information must be contained in the reaction representation itself. Using a slight modification of the BE-matrix, where the atomic number of atom A_k replaces the number of unshared electrons in the diagonal element b_{kk}, it is possible to differentiate each individual compound. Another advantage of this matrix representation is that all the reactants can be represented within the same data structure. A potential disadvantage is that the BE-matrix must include a



Reactants

```
60001000
06001000
00100100
00010100
11006200
00112600
00000011
000000135
```

Products

```
60001000
06001000
00100100
00010100
11006101
00111610
00000110
000010035
```

CNN INPUT 661166135 000000110000112000000000001

CNN OUTPUT 000000110000111100000100010

Figure 28. BE-Matrix Representation of Reactions.

stoichiometric accounting of all of the participants in the reaction.[170]

Figure 28 shows the matrix representations for the reaction of 2, 2-dimethyl-ethylene with hydrogen bromide. The modified BE-matrix representations for the reaction at the top of Figure 28 are shown in the middle of the figure and the resulting form of the input and output vectors used in the neural network are shown at the bottom of the figure. Since the matrix is symmetric only the lower triangle was used as the input for the neural network. The first n elements are the diagonal elements b_{kk} which contain the atomic numbers of atoms 1 to n (n = 8 in this example). The remaining elements are the lower triangle, b_{ij} ($i > j$), row by row, up to but not including the diagonal element. For the output vector the lower triangle minus the diagonal was used, b_{ij} where $i > j$. The diagonal, which contains the atom-type information, is only needed for input to the network: it is unnecessary for the network to predict the atoms since they do not change during the course of the reaction.

Methods

Neural Network Simulator

The ANSIM neural network simulator program was also used for these reaction prediction studies. ANSIM version 2.30, which requires Microsoft Windows 2.0 or later, was run on an IBM-compatible 20 MHz

80386 microcomputer equipped with an 80387 math coprocessor. The ANSIM internal data scaling features were used to scale the data into the ANSIM-required range of -0.5 to +0.5, and to re-scale the data from the network back to the original range.

Neural Network Configuration

A backpropagation neural network with two hidden layers was employed for these reaction predictions. This is a difference from the networks used for the EAS reaction prediction in Chapter III. Backpropagation networks have been shown to excel at pattern recognition and feature detection,[171] and they indirectly implement non-linear statistical analysis methods in an adaptive and non-parametric manner to generate an internal representation of the reaction being studied. A separate neural network was used for each reaction type, with the exception of the Diels-Alder and retro-Diels-Alder reactions, which were treated in the same network. Figure 29 shows the neural network used for Markovnikov addition. The input layer consists of a vector containing atom-type and connectivity information. The size of the input layer was determined by the number of atoms in the reactants and ranged in size from 36 units for the Markovnikov reaction network, 66 units for the Saytzeff rule prediction network, and 120 units for the Diels-Alder network.

Figure 30 summarizes the information on the network sizes used.

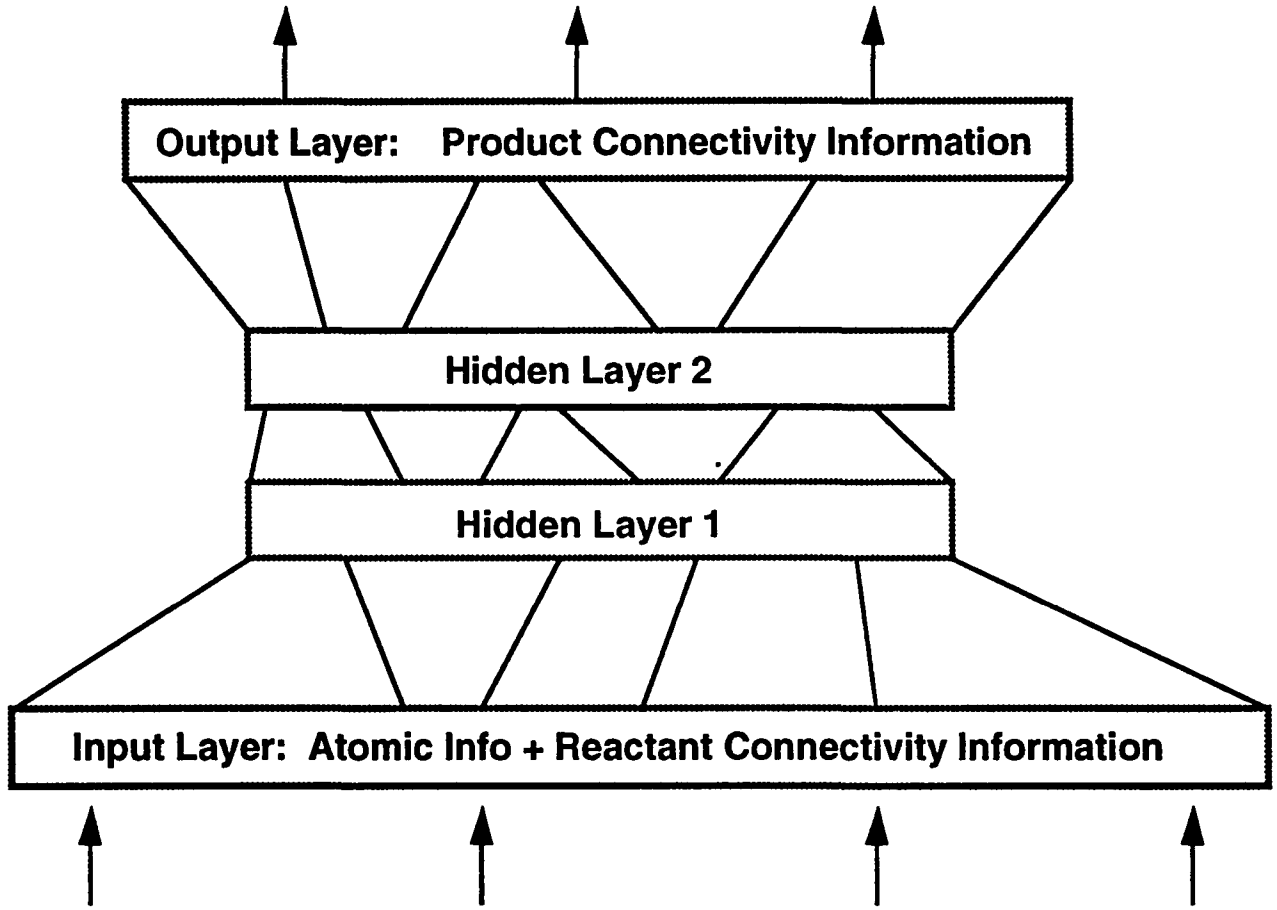


Figure 29. Neural Network for Reaction Prediction With BE-Matrices.

Network	Number of Nodes in Layers				Training cycles	RMS Error
	Input	Hidden1	Hidden2	Output		
Markovnikov	36	8	8	28	8100	0.142
Saytzeff	66	24	24	55	7200	0.112
Diels-Alder	120	36	36	105	9000	0.008

Reaction	Training Set	Test Set
Markovnikov	18 / 18 Correct	6 / 7 Correct
Saytzeff	74 / 74 Correct	8 / 9 Correct
Diels-Alder	14 / 14 Correct	4 / 4 Correct
Retro Diels-Alder	16 / 16 Correct	2 / 2 Correct

Figure 30. BE-Matrix Network Sizes and Results for Reaction Prediction.

Two hidden layers were used due to reports[33,42] that showed that a network with two hidden layers can compute any mapping to arbitrary accuracy. For the current set of predictions a network with a single hidden layer performed as well as one with two hidden layers. However, in the network with two hidden layers the number of connections was nearly halved compared to having the same number of hidden units arranged in a single layer. Since the training time of the network increases proportionally to the number of connections, nets with two hidden layers were used. The size of the hidden layers varied depending on the size of the input and output layers. The output layer contained only connectivity information and required 28, 55, or 105 units respectively for the three cases studied.

Neural Network Training

Data on the reactions were split into two sets, with 75 to 90 percent of the reactions used to train the networks and the remaining 10 to 25 percent of the examples used to test the fully trained networks. In this way, the ability of the network to generalize from the training set could be measured. A learning rate of 0.01 was used during the training phase and the best results were obtained when random noise was added to the input patterns. The random noise was allowed to decay to zero over 1000 presentations of the training data set. When the network was trained

without adding noise to the inputs it usually converged to a state in which the predicted product was the average of all the possible products for the training set. Comparatively, training with input noise gave distinct products for each reaction. Training was continued until the root mean squared (RMS) error over the outputs was minimized. Several iterations of training, damaging the weights, and retraining were required to achieve the lowest RMS error.

Results

Markovnikov Addition Reactions

Eighteen examples of Markovnikov reactions from Table 7 were used to train a network which had 36 inputs, two hidden layers of 8 units each, and 28 output units. The network correctly predicted the position of addition of the halogen for all of the training examples. The seven test reactions, except for the last one in the table, were correctly predicted. In the case of the last test reaction, the reactant did not contain a double bond, but the network predicted addition would occur anyway. This may reflect the fact that there were no examples in the training set where the compounds did not react, and the network tried to apply its internally developed rule inappropriately. As networks seem to be more able to interpolate than extrapolate,[33] it is important that the training data span

Table 7

CNN Results for Markovnikov Reaction

Training Set						
R1	R2	R3	R4	X	Expected Products	CNN Pred. Products
Me	Me	Me	Me	Br	2 + 4	2 + 4
Me	Me	Me	Me	Br	2 + 4	2 + 4
Me	Me	Me	H	Br	2	2
Me	Me	H	Me	Br	2	2
Me	Me	H	H	Br	2	2
Me	H	Me	H	Br	2 + 4	2 + 4
Me	H	Me	H	Br	2 + 4	2 + 4
Me	H	H	Me	Br	2 + 4	2 + 4
Me	H	H	Me	Br	2 + 4	2 + 4
Me	H	H	H	Br	2	2
H	Me	Me	Me	Br	4	4
H	Me	H	Me	Br	2 + 4	2 + 4
H	Me	H	Me	Br	2 + 4	2 + 4
H	H	Me	Me	Br	4	4
H	H	Me	H	Br	4	4
H	H	H	Me	Br	4	4
H	H	H	H	Br	2 + 4	2 + 4
H	H	H	H	Br	2 + 4	2 + 4
Test Set						
R1	R2	R3	R4	X	Expected Products	CNN Pred. Products
Me	H	Me	Me	Br	4	4
H	Me	H	H	Br	2	2
H	Me	Me	H	Br	2 + 4	2 + 4
H	Me	Me	H	Br	2 + 4	2 + 4
Me	Me	H	H	Cl	2	2
H	H	H	Me	Cl	4	4
Me	Me	Me	Me	Br	None	2 + 4

the "reaction space" adequately. This result also points out the need for

integration of neural network methods with expert systems which can help determine the appropriate network to apply to a particular reaction class.[18]

Diels-Alder Cycloaddition and Retro Diels-Alder Reactions

Thirty training examples were used to develop a neural network suitable for predicting Diels-Alder reactions. The training examples were divided between Diels-Alder and retro-Diels-Alder reactions. The network contained 120 input units, two hidden layers of 36 units each, and 105 output units. The network first had to determine if the input pattern corresponded to the reactants for the forward or reverse Diels-Alder reaction and then, if it was a forward reaction, what was the orientation of the major product. All 30 of the training examples (see Table 8) were correctly predicted by the network and as were the 6 test reactions.

Saytzeff Elimination Reactions

For the Saytzeff reaction a network with 66 inputs, two hidden layers of 24 units each, and 55 outputs was trained on 74 examples of E1 elimination reactions (Table 9). All 74 of the training set reactions were predicted correctly. The network predicted eight of the nine test set reactions correctly. The remaining test reaction did not have any hydrogens on carbons adjacent to the bromine and thus could not undergo dehydrohalogenation. But, the network predicted 80% of the elimination

product 19. Again, this points out the need to train the network with both successful reactions and cases where no reaction occurs.

Summary

All three of the neural networks were able to use examples of the reactions to develop an internal representation of the rules which determine the regiochemistry of the reactions. In the case of the Diels-Alder network, both forward and reverse reactions were included in the training set. The resulting network correctly predicted not only the regiochemistry of the forward reaction but also the appropriate disconnections in the reverse reaction. This result indicates that a suitably trained neural network could be useful not only for predicting reaction products but also in retrosynthetic analysis to select possible precursors. In addition, the results presented also show the usefulness of the modified BE-matrix representation for describing chemical reactions for input to neural networks.

Table 8

CNN Results for Diels-Alder and Retro-Diels-Alder Reactions

Training Set						
Rxn Type	R1	R2	R3	R4	Expected Products	CNN Pred. Products
Retro DA	H	Ph	H	CHO	Retro DA	Retro DA
DA	COOH	H	Ph	H	Ortho	Ortho
Retro DA	H	Me	H	COOMe	Retro DA	Retro DA
Retro DA	H	Ph	H	COOH	Retro DA	Retro DA
DA	Me	H	CHO	H	Ortho	Ortho
DA	Me	H	CN	H	Ortho	Ortho
DA	Me	H	COOMe	H	Ortho	Ortho
DA	Me	H	Ph	H	Ortho	Ortho
Retro DA	H	Me	H	COOH	Retro DA	Retro DA
DA	Ph	H	CHO	H	Ortho	Ortho
Retro DA	H	Me	H	CHO	Retro DA	Retro DA
Retro DA	H	Ph	H	CN	Retro DA	Retro DA
Retro DA	H	COOH	H	Ph	Retro DA	Retro DA
DA	Ph	H	COOH	H	Ortho	Ortho
DA	H	COOH	H	Ph	Para	Para
Retro DA	Ph	H	COOH	H	Retro DA	Retro DA
Retro DA	Ph	H	CN	H	Retro DA	Retro DA
Retro DA	Ph	H	CHO	H	Retro DA	Retro DA
DA	H	Me	H	CN	Para	Para
Retro DA	Me	H	Ph	H	Retro DA	Retro DA
DA	H	Me	H	COOH	Para	Para
DA	H	Me	H	COOMe	Para	Para
Retro DA	H	Me	H	Ph	Retro DA	Retro DA
Retro DA	Me	H	COOH	H	Retro DA	Retro DA
DA	H	Me	H	Ph	Para	Para
Retro DA	Me	H	CN	H	Retro DA	Retro DA
DA	H	Ph	H	CHO	Para	Para
Retro DA	Me	H	CHO	H	Retro DA	Retro DA
DA	H	Ph	H	CN	Para	Para
Retro DA	COOH	H	Ph	H	Retro DA	
Test Set						
Rxn Type	R1	R2	R3	R4	Expected Products	CNN Pred. Products
DA	Me	H	COOH	H	Ortho	Ortho
Retro DA	Me	H	COOMe	H	Retro DA	Retro DA
DA	Ph	H	CN	H	Ortho	Ortho
DA	H	Me	H	CHO	Para	Para
Retro DA	H	Me	H	CN	Retro DA	Retro DA
DA	H	Ph	H	COOH	Para	Para

Table 9

CNN Results for Saytzeff Reactions

Training Set								
R1	R2	R3	R4	R5	R6	X	Expected Products	CNN Predicted Products
H	H	H	H	H	H	Br	14 + 16	14 + 16
H	H	H	H	H	H	Br	14 + 16	14 + 16
H	H	H	H	H	C	Br	16	16
H	H	H	H	C	H	Br	16	16
H	H	H	H	C	C	Br	16	16
H	H	H	C	H	H	Br	16	16
H	H	H	C	H	C	Br	16	16
H	H	H	C	C	C	Br	14	14
H	H	C	H	H	H	Br	14	14
H	H	C	H	H	C	Br	14 + 16	14 + 16
H	H	C	H	H	C	Br	14 + 16	14 + 16
H	H	C	H	C	H	Br	14 + 16	14 + 16
H	H	C	H	C	H	Br	14 + 16	14 + 16
H	H	C	H	C	C	Br	16	16
H	H	C	C	H	H	Br	14 + 16	14 + 16
H	H	C	C	H	H	Br	14 + 16	14 + 16
H	H	C	C	H	C	Br	16	16
H	H	C	C	C	H	Br	16	16
H	C	H	H	H	H	Br	14	14
H	C	H	H	H	C	Br	14 + 16	14 + 16
H	C	H	H	H	C	Br	14 + 16	14 + 16
H	C	H	H	C	H	Br	14 + 16	14 + 16
H	C	H	H	C	H	Br	14 + 16	14 + 16
H	C	H	H	C	C	Br	16	16
H	C	H	C	H	C	Br	16	16
H	C	H	C	C	H	Br	16	16
H	C	H	C	C	C	Br	14	14
H	C	C	H	H	H	Br	14	14
H	C	C	H	H	C	Br	14	14
H	C	C	H	C	H	Br	14	14
H	C	C	H	C	C	Br	14 + 16	14 + 16
H	C	C	H	C	C	Br	14 + 16	14 + 16
H	C	C	C	H	H	Br	14	14
H	C	C	C	H	C	Br	14 + 16	14 + 16
H	C	C	C	H	C	Br	14 + 16	14 + 16

Table 9--Continued

Training Set								
R1	R2	R3	R4	R5	R6	X	Expected Products	CNN Predicted Products
H	C	C	C	C	H	Br	14 + 16	14 + 16
H	C	C	C	C	H	Br	14 + 16	14 + 16
H	C	C	C	C	C	Br	14	14
C	H	H	H	H	H	Br	14	14
C	H	H	H	C	H	Br	14 + 16	14 + 16
C	H	H	H	C	H	Br	14 + 16	14 + 16
C	H	H	H	C	C	Br	16	16
C	H	H	C	H	H	Br	14 + 16	14 + 16
C	H	H	C	H	H	Br	14 + 16	14 + 16
C	H	H	C	H	C	Br	16	16
C	H	H	C	C	H	Br	16	16
C	H	H	C	C	C	Br	14	14
C	H	C	H	H	H	Br	14	14
C	H	C	H	C	H	Br	14	14
C	H	C	H	C	C	Br	14 + 16	14 + 16
C	H	C	H	C	C	Br	14 + 16	14 + 16
C	H	C	C	H	H	Br	14	14
C	H	C	C	H	C	Br	14 + 16	14 + 16
C	H	C	C	H	C	Br	14 + 16	14 + 16
C	H	C	C	C	H	Br	14 + 16	14 + 16
C	H	C	C	C	H	Br	14 + 16	14 + 16
C	H	C	C	C	C	Br	14	14
C	C	H	H	H	H	Br	14	14
C	C	H	H	H	C	Br	14	14
C	C	H	H	C	H	Br	14	14
C	C	H	H	C	C	Br	14 + 16	14 + 16
C	C	H	H	C	C	Br	14 + 16	14 + 16
C	C	H	C	H	H	Br	14	14
C	C	H	C	H	C	Br	14 + 16	14 + 16
C	C	H	C	H	C	Br	14 + 16	14 + 16
C	C	H	C	C	H	Br	14 + 16	14 + 16
C	C	H	C	C	H	Br	14 + 16	14 + 16
C	C	H	C	C	C	Br	14	14
C	C	C	H	H	H	Br	16	16
C	C	C	H	H	C	Br	16	16
C	C	C	H	C	H	Br	16	16
C	C	C	C	H	H	Br	16	16
C	C	C	C	H	C	Br	16	16
C	C	C	C	C	H	Br	16	16

Table 9--Continued

Test Set								
R1	R2	R3	R4	R5	R6	X	Expected Products	CNN Predicted Products
H	H	H	C	C	H	Br	16	16
H	H	C	C	C	C	Br	14	14
H	C	H	C	H	H	Br	14 + 16	14 + 16
H	C	H	C	H	H	Br	14 + 16	14 + 16
C	H	H	H	H	C	Br	14 + 16	14 + 16
C	H	H	H	H	C	Br	14 + 16	14 + 16
C	H	C	H	H	C	Br	14	14
C	C	C	H	C	C	Br	16	16
C	C	C	C	C	C	Br	None	16

CHAPTER V

CONCLUSIONS AND FUTURE DIRECTIONS

Results and Conclusions

We focused our neural network research on the two premier requirements for any computer-assisted organic chemistry method: (1) the ability of the methodology to map complex nonlinear structure-reactivity relationships and (2) the representation of chemical information in a form suitable for the methodology. The ability of computational neural networks to act as model-free mapping devices was clearly demonstrated by modeling a 3D response surface. The CNN was able to produce an accurate approximation of the underlying function that generated the response surface, merely by training on examples of the data at various points on the response surface. We were able to make a significant contribution to the much more difficult representation problem by devising two related connectivity-matrix derived chemical structure representations. Since no prior chemical knowledge is programmed into a CNN, all of the information necessary to compute a desired structure-property mapping must be embedded in the representation. The first of the representations, based on a compact connection table, was used to train computational neural

networks to predict product ratios in electrophilic substitution reactions as well as chemists and better than a rule-based expert system. The second representation, an extension of the Dugundji-Ugi BE-matrix, was sufficiently general to enable neural networks to learn elementary, but important chemical reactivity rules relating to regiochemistry, directly from examples of the three classes of reactions.

Mapping Properties

Generalized perceptrons can function as powerful, essentially model-free, mapping devices that can be applied to a wide range of problems in chemistry. The power of these CNNs lies in their ability to represent very general mappings without the need to specify the mathematical form of the mapping explicitly. And although the model three-dimensional response surface with two input variables studied here is relatively simple, it permitted the clarification of several important issues that are relevant to the applications of generalized perceptrons in general and in chemistry in particular. CNNs composed of generalized perceptrons were able to accurately map the response surface, even with only a small number of samples and in the presence of noise. Correlated variables, and random, irrelevant variables had little effect on the mapping ability of the CNN.

On the other hand, the shape of the transfer function had a much larger effect on whether the network would even converge to a solution.

The shapes of the surfaces generated by the hidden units, as displayed in Figures 12 and 13, reinforce the understanding of neural networks as function approximation methods or mapping devices. Being able to visually see the surface or function that the hidden units generate removes much of the mystery from what happens in the "hidden units". The shapes of the surfaces in Figure 12 look very much like three dimensional analogs of the *tanh* used as the transfer function. Aoyama and Ichikawa[172] showed that a neural net could learn the simple linear function $y = x$, and the bilinear function $y = |x - 10|$, which looks like the letter V. However, there was always a small amount of error because the CNN was fitting straight line(s) with a curve, the sigmoid function in their case. Perhaps a CNN which had a number of different types of transfer functions in its PEs would give the best overall mapping. This proposed universal computational artificial neural network (UCANN) might include linear, *tanh*, parabolic, *sine*, and *cosine* transfer functions as well as radial basis functions in its PEs.

The network learning rule uses essentially local information between pairs of processing elements to cause a global decrease in the error function by modifying the inter-neuron weights. These weights are the coefficients of a nonlinear function of nonlinear functions. A nearly optimum set of weights or coefficients then leads to an approximation of the true function that is both enabled by and also limited by the shape of the transfer function. Essentially, the choice of the transfer function is the choice of

the basis functions that the network has available to build a model. What the learning rule provides is an unbiased ("model-free") way to construct the mapping.

Molecular Representation for Reaction Prediction

The ability of neural networks to predict a chemical reaction as well as chemists or an expert system was demonstrated for the EAS reaction. The neural net was able to learn by example to make predictions for cases on which it had not been trained. Quantitative predictions were made by the network in contrast to the expert system which made qualitative predictions. Also the neural network did not require formulating rules about reactivity in order to make useful predictions but was able to form an internal model of the reaction by extracting information directly from examples of the reaction. The method used for representing the chemical information for the neural network is a major factor in determining the predictive ability of the network. A more descriptive, less specific representation derived from connectivity information gave better results than the representation that employed specific atomic-charge information. Other representations were tried that have had reasonable success in the applications of chemical graph theory but were not found to be suitable for CNN application. Specifically, the graph-theoretic transforms and Randic indices were not discriminating enough to differentiate compounds with

similar structures but different reactivity. This is not a indictment of graph-theoretic structural descriptors in general, and may be only a result of the particular graph-theoretic representations tried.

All three of the neural networks trained with BE-matrices were able to use examples of the reactions to develop an internal representation of the rules which determine the regiochemistry of the reactions. In the case of the Diels-Alder network, both forward and reverse reactions were included in the training set. The resulting network correctly predicted not only the regiochemistry of the forward reaction but also the appropriate disconnections in the reverse reaction. This result indicates that a suitably trained neural network could be useful not only for predicting reaction products but also in retrosynthetic analysis to select possible precursors. In addition, the results presented here also show the usefulness of the modified Dugundji-Ugi BE-matrix representation for describing chemical reactions for input to neural networks. In my opinion, the development by Dugundji and Ugi of the BE-matrix as part of an overall mathematical model of constitutional chemistry[164,169,170] is a major contribution to the understanding of the logical structure of organic chemistry. Hendrickson[173] recently (mid 1992) reconciled three systems of reaction description, which included Hendrickson's and Ugi's. Further explorations are needed to test the generality of this representation in neural network applications to chemistry. Extensions of the BE-matrix representation that include

non-connectivity related chemical information such as electronegativities, polarizabilities, or bond energies, might also prove useful and should be investigated.

Challenges and Opportunities for the Future

Many new applications of CNNs remain to be discovered and exploited in chemistry and related fields. The rapidly growing literature in these areas—it has increased more than ten-fold in the last three years—attests to their vitality. Moreover, the breadth and variety of the applications are increasing as well. Whether CNNs will fulfil their promise in chemistry at this time still, however, remains an open question. Clearly, CNNs can be used to build complicated models directly from data. What is not clear is whether the models they generate are significantly better than could be constructed by other means. However, even if the answer to the above question is no in some cases, the power of CNNs to address a wide and growing variety of problems in many fields strongly suggests that CNNs warrant further serious study as problem solving tools and as interesting entities in their own right.

CNNs of even modest complexity can possess large numbers of weights that must be determined during network training. For example, a generalized perceptron with only two input nodes, two hidden layers of five nodes each, and a single output node possesses 51 weights. Generally,

it is assumed that in order to achieve statistically reliable predictions it is necessary to have at least three samples per weight (*Cf.* Refs 27, 55). This would mean that for statistically reliable predictions, the simple multilayer net described above would require a training set of about 130 samples. This represents a serious impediment to the use of CNNs in many types of problems of interest in chemistry, where the size of the dataset may be relatively small. If, for example, a QSAR study is undertaken, the amount of data needed to train a reliable CNN is substantial. Thus, the amount of synthesis needed to develop a dataset of suitable size may be such as to obviate the need for a QSAR analysis — the problem, essentially, will have already been solved. Fortunately, as we have shown in our work, cases do exist in which reliable results have been achieved, even with noisy data, when the samples-to-weights ratio is less than three. Nevertheless, a deeper understanding of the this issue would certainly be of great value in future work, especially in chemistry.

Neural networks may be useful as an adjunct to expert systems in cases where concrete rules describing reactivity can not be easily formulated. Databases of chemical reactions could provide sufficient examples to train a set of neural nets to predict a number of reactions. The connection-table approach described here needs further exploration to determine how to best represent chemical information in order for a neural network to make useful predictions about chemical reactions. It seems

doubtful that there would exist a completely universal structure representation that would allow extension of this neural network approach to all types of reactions. One of the difficulties of the connectivity-based representation is the requirement for a semi-canonical numbering scheme.

Even in light of the problems noted above, CNNs, due to their ability to find complex, nonlinear mappings, appear to have a future in chemistry but not, perhaps, in their "pure" form. Recently, considerable work has been carried out on a number of hybrid systems that incorporate a variety of technologies including fuzzy logic, genetic algorithms, and artificial intelligence along with neural networks.[174] Although applications of hybrid systems to date are primarily in the business area, as the behavior of these systems becomes better understood, applications in other areas including chemistry will no doubt begin to occur with increasing frequency. Recently, in fact, several papers which use hybrid neural-network methods have appeared. Otto[104] used fuzzy logic and neural nets to design intelligent analytical instruments; Guo and Uhrig[138] used Kohonen SOM's to cluster data about nuclear power plant heat output and then trained a generalized perceptron on the cluster centroids; and Mavrovouniotis and Chang[175] described the use of hierarchies of generalized perceptrons. Ham, Cohen & Cho[176] used a hybrid net consisting of a generalized perceptron and counter-propagation network to detect biological substances in complex aqueous solutions using IR spectral data as input. Otto & Hoerchner[128]

used a fuzzy neural net, where the neural-net component was an unsupervised adaptive bidirectional associative memory network, for finding the best match for the UV spectra of organic compounds. D'Antone[177] described a neural net which functioned as part of an expert system; the network was trained to recognize malfunctions in a particle detector, and the expert system suggested how to remedy the instrument failure.

The amount of research in and applications of CNNs has increased at a phenomenal rate during the last half of the 80's. Even in chemistry, as seen in Figure 6, CNN applications have increased significantly in both variety and numbers over the last several years. Their future in chemistry appears to be bright, particularly in analytical chemistry, chemical engineering, and biomolecular informatics. However, there are a number of issues that present significant challenges to some of the other chemical applications of CNNs. We have constructively dealt with two of the most important ones in this work: (1) how best to represent chemical information in a format suitable for neural-network applications and (2) how to deal with the relatively meager amount of data available on many chemical systems of interest.

Currently, CNNs are only in the early linear phase of the "S-curve" of neural-network technology, and in chemistry only the first few steps along this path have been taken. However, the future appears to hold great

promise, and many exciting new applications of neural nets in chemistry remain to be discovered.

Final Comments

A final comment to put computational neural network methodology in perspective draws on a quote from physicist John Denker[178] "neural networks are the second best way of doing just about anything"(p 10). Denker's comment was further elaborated by Hertz, Krogh, and Palmer[179] who wrote

> The *best* way is to find and use the right rules or the optimum algorithm for each particular problem, but this can be inordinately expensive and time-consuming. There is plenty of scope for a second best approach based on learning by example (p 10).

REFERENCES

1. Barone, R.; Chanon, M. Computer-Aided Organic Synthesis (CAOS). In *Computer Aids to Chemistry*; Vernin, G.; Chanon, M., Ed.; Ellis Horwood Limited: Chichester, UK, 1986; Chapter 1, p 19-102.

2. Noordik, J. H. Computer Assisted Organic Synthesis. *Recl. Trav. Chim. Pays-Bas,* **1992**, *111(4),* unnumbered page preceding page 239.

3. Ott, M. A.; Noordik, J. H. Computer Tools for Reaction Retrieval and Synthesis Planning in Organic Chemistry. A Brief Review of Their History, Methods, and Programs. *Recl. Trav. Chim. Pays-Bas*, **1992**, *111(4)*, 239-246.

4. Corey, E.J.; Long, A.K.; Rubenstein, S.D. Computer-Assisted Analysis in Organic Synthesis. *Science* **1985**, 408-418.

5. Bauer, J; Herges, R.; Fontain, E.; Ugi, I. IGOR and Computer-Assisted Innovation in Chemistry. *Chimia* **1985**, *39*, 43-53.

6. Gasteiger, J.; Hutchings, M. G.; Christoph, B.; Gann, L.; Hiller, C.; Low, P.; Marsili, M.; Saller, H.; Yuki, K. A New Treatment of Chemical Reactivity: Development of EROS, an Expert System for Reaction Prediction and Synthesis Design. *Topics Curr. Chem.* **1987**, *137*, 19-74.

7. Hendrickson, J. B. Organic Synthesis in the Age of Computers. *Angew. Chem. Int. Ed. Engl.*, **1990**, *29*, 1286-1295.

8. Jorgensen, W. L.; Laird, E. R.; Gushurst, A. J.; Fleischer, J. M.; Gothe, S. A.; Helson, H. E.; Paderes, G. D.; Sinclair, S. CAMEO: A Program for the Logical Prediction of the Products of Organic Reactions. *Pure Appl. Chem.*, **1990**, *62(10)*, 1921-32.

9. Gasteiger, J.; Ihlenfeldt, W. D.; Rose, P. A Collection of Computer Methods for Synthesis Design and Reaction Prediction. *Recl. Trav. Chim. Pays-Bas*, **1992**, *111*, 270-290.

10. CASREACT Chemical Reactions Database, Chemical Abstracts Service, American Chemical Society, Columbus, OH, 1985-1992.

11. Wilcox, C. S.; Levinson, R. A. A Self-Organized Knowledge Base for Recall, Design, and Discovery in Organic Chemistry. In *ACS Symp. Ser 306 Artificial Intelligence Applications in Chemistry*: American Chemical Society: Washington, DC, 1986; p. 209-230.

12. Schulz, K. P.; Hofmann, P.; Gasteiger, J. Use of an Associative Memory System for Reactivity Prediction. In *Software-Entwickl, Chem. 2, Proc. Workshops "Comput. Chem." Meeting Date 1987*; Gasteiger, J. Ed.; Springer: Berlin, FRG, 1988; p. 181-196.

13. Elrod, D.W.; Maggiora, G. M.; Trenary, R. G. Predicting Chemical Reactions With a Neural Network. In *Computing in the 90's. The First Great Lakes Computer Science Conference Proceedings. (October 1989)*; Sherwani, N. A.; De Doncker, E.; Kapenga, J. A., Ed.; Springer-Verlag: Berlin, Germany, 1991, *Lect N. Comp V. 507*, p 392-398.

14. Elrod, D. W.; Maggiora, G. M.; Trenary, R. G. Applications of Neural Networks in Chemistry. 1. Prediction of Electrophilic Aromatic Substitution Reactions. *J. Chem. Inf. Comput. Sci.* **1990**, *30(4)*, 477-84.

15. Elrod, D. W.; Maggiora, G. M.; Trenary, R. G. Applications for Neural Networks in Chemistry. 2. A General Connectivity Representation for the Prediction of Regiochemistry. *Tetrahedron Comput. Methodol.* **1990**, *3*, 163-74.

16. Kvasnicka, V.; Pospichal, J. Application of Neural Networks in Chemistry. Prediction of Product Distribution of Nitration in a Series of Monosubstituted Benzenes. *THEOCHEM* **1991**, *81(3-4)*, 227-42.

17. Zou, Y.; Johnson, M. A.; Tsai, C.C. Modeling Aromatic Nitration Reactions Using Graph-Theoretic Transforms. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 442-450.

18. Luce, H. H.; Govind, R. Neural Network Applications in Synthetic Organic Chemistry. I. A Hybrid System Which Performs Retrosynthetic Analysis. *Tetrahedron Comput. Methodol. 1990, 3(3-4)*, 143-61.

19. Marie, S.; Nicolle-Adam. A.; Villemin, D. MAROCO: A Learning Machine on Object-Based Representations in Organic Chemistry. in *Proceedings of the First International Conference on Knowledge Modeling and Expertise Transfer*, 22-24 April 1991, Sophia-Antipolis, France; IOS: Amsterdam, 1991, p 115-29.

20. Maggiora, G. M.; Elrod, D. W.; Trenary, R. G. Computational Neural Nets as Model-Free Mapping Devices. *J. Chem. Inf. Comput. Sci.* (In Press: **1992**, *32*).

21. Elrod, D. W.; Maggiora, G. M. Artificial Neural Networks: A New Computational Paradigm with Applications in Chemistry. To appear in the *Proceedings of Online Information 92*, London, UK, Dec. 8, 1992.

22. Rumelhart, D. E.; McClelland, J. L. *Parallel Distributed Processing, Vols. I,II*; MIT Press: Cambridge, Massachusetts, 1986.

23. Hertz, J.; Krogh, A.; Palmer, R. G. *Introduction To The Theory of Neural Computation*; Addison-Wesley Publishing Company: Redwood City, California, 1991.

24. Hecht-Nielsen, R. *Neurocomputing*; Addison-Wesley Publishing Company: Redwood City, California, 1990.

25. Simpson, P.K. *Artificial Neural Systems: Foundations, Paradigms, Applications, and Implementations*; Pergamon Press: New York, 1990.

26. Lippmann, R.P. An Introduction to Computing with Neural Nets; *IEEE ASSP Magazine* **1987**, *April*, 4-22.

27. White, H. Learning in Artificial Neural Networks: A Statistical Perspective; *Neural Comp.* **1989**, *1*, 425-464.

28. Simpson, P.K. *Artificial Neural Systems: Foundations, Paradigms, Applications, and Implementations*; Pergamon Press: New York, 1990, Chapter 5.

29. Kolmogorov, A.N. On the Representation of Continuous Functions of Several Variables by Superposition of Continuous Functions of One Variable and Addition; *Dokl. Akad. Nauk SSSR* **1957**, *114*, 953-956.

30. Cybenko, G. Approximation by Superpositions of a Sigmoidal Function; *Math. Contr. Sig. Sys.* **1989**, *4*, 303-314.

31. Hecht-Nielsen, R. Theory of the Backpropagation Neural Network. In *Proc. Int. Joint Conf. Neural Networks, Vol. I*; IEEE Press: New York, 1989.

32. Stinchcombe, M.; White, H. Universal Approximation Using Feedforward Networks with Non-Sigmoid Hidden Layer Activation Functions. In *Proceedings of the International Joint Conference on Neural Networks (Washington, D.C., June 1989); 1989; Vol.I, pp. 607-611.*

33. Chester, D.L. Why Two Hidden Layers are Better Than One. In *Proceedings of the International Joint Conference on Neural Networks (Washington, D.C., January 1989); 1989; Vol.I, pp. 265-268.*

34. Kosko, B. *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*; Prentice Hall: Englewood Cliffs, NJ, 1992.

35. Favlow, S.J., Ed. *Self-Organizing Methods in Modeling: GMDH Algorithms*; Marcel Dekker: New York, 1984.

36. Minsky, M.; Papert, S. *Perceptrons*; MIT Press: Cambridge, MA, 1969.

37. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning Representations by Back-Propagating Errors, *Nature* **1986**, *323*, 533-536.

38. Werbos, P. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences, Ph.D. dissertation, Harvard University, Cambridge, MA, 1974.

39. Hertz, J.; Krogh, A.; Palmer, R. G. *Introduction To The Theory of Neural Computation*; Addison-Wesley Publishing Co: Redwood City, CA, 1991, p 102-104.

40. Klimasauskas, C. C. Neural Networks: A Short Course. From Theory to Application. *PC AI*, **1988**, *4*, 26-30.

41. Denning, P. J. Neural Networks. *Am. Scientist*, **1992**, *80*, 426-9.

42. Lapedes, A.; Farber, R. How Neural Networks Work. In *Neural Information Processing Systems*, Denver 1987; Anderson, D.Z., Ed.; American Instit. of Physics: New York, 1988; 442-456.

43. ANSIM neural network Simulator program, Science Applications International Corporation, San Diego, CA, 1988.

44. Microsoft Windows program, Microsoft Corporation, One Microsoft Way, Redmond, WA 98052-6399, 1988.

45. International Business Machines Corporation, Boca Raton, FL.

46. N-Net 210 neural network simulation program, AI-Ware Corp., Cleveland, OH, 1989.

47. Pao, Y. H. *Adaptive Pattern Recognition and Neural Networks*; Addison-Wesley: Reading, MA, 1989.

48. Nets neural network simulator program version 2.0, developed at NASA Johnson Space center, Houston, TX, by P. Baffes. Available from COSMIC, 382 East Broad St., University of Georgia, Athens, GA 30602, 1989.

49. Sun Microsystems Computer Corporation, Mountain View, CA.

50. NeuralWorks Professional II Plus neural network simulator program, NeuralWare, Inc, Pittsburgh, PA, 1990.

51. Hall, L. H.; Mohney, B.; Kier, L. B. The Electrotopological State: Structure Information at the Atomic Level for Molecular Graphs. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 76-82.

52. Rumelhart, D. E.; McClelland, J. L. *Parallel Distributed Processing, Vol. I*; MIT Press: Cambridge, Massachusetts, 1986, p 330.

53. Jones, W. P.; Hoskins, J. Back-Propagation. *Byte*, **October 1987**, 155-162.

54. Kvasnicka, V. An Application of Neural Networks in Chemistry. Prediction of ^{13}C NMR Chemical Shifts; *J. Math. Chem.* **1991**, *6*, 63-76.

55. Andrea, T. A.; Kalayeh, H. Applications of Neural Networks in Quantitative Structure-Activity Relationships of Dihydrofolate Reductase Inhibitors. *J. Med. Chem.* **1991**, *34*, 2824-2836.

56. Sharaf, M.A.; Illman, D.L.; Kowalski, B.R. *Chemometrics*; Wiley-Interscience: New York, 1986.

57. Kramer, M.A. Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. *AIChE J.* **1991**, *37*, 233-243.

58. Hertz, J.; Krogh, A.; Palmer, R. G. *Introduction To The Theory of Neural Computation*; Addison-Wesley Publishing Company: Redwood City, California, 1991, Chapter 6, p 115-162.

59. Hussain, A.; Kane, A. A Neural Network Approach for Identification of Relevant Structural Descriptors in QSAR Analysis; *submitted to Pharm. Res.* **1992**.

60. Hertz, J.; Krogh, A.; Palmer, R. G. *Introduction To The Theory of Neural Computation*; Addison-Wesley Publishing Company: Redwood City, California, 1991, Chapter 2, p 11-41.

61. Mézard, M.; Nadal, J.-P. Learning in Feedforward Layered Networks: The Tiling Algorithm; *J. Phys. A* **1989**, *22*, 2191-2204.

62. Marchand, M.; Golea, M.; Ruján, P. A Convergence Theorem for Sequential Learning in Two-Layer Perceptrons; *Europhys. Letts.* **1990**, *11*, 487-492.

63. Frean, F. The Upstart Algorithm: A Method for Constructing and Training Feedforward Neural Networks; *Neural Comp.* **1990**, *2*, 198-209.

64. Minai, A. A.; Williams, R. D. Acceleration of Back-Propagation Through Learning Rate and Momentum Adaptation. In *Proceedings International Joint Conference on Neural Networks (January, 1990)*; Caudill, M., Ed.; Lawrence Erlbaum Assoc.: Hillsdale, NJ, 1990; p. 676-679.

65. Fletcher, R. *Practical Methods of Optimization, Vol.1*; Wiley-Interscience: New York, 1980.

66. Baba, N. A New Approach to Finding the Global Minimum of the Error Function of Neural Networks; *Neural Networks* **1989**, *2*, 367-373.

67. Szu, H.; Hartley, R. Fast Simulated Annealing. *Phys. Letts.* **1987**, *1222*, 157-162.

68. Montana, D.J.; Davis, L. Training Feedforward Networks Using Genetic Algorithms. In *Eleventh International Joint Conference on Artificial Intelligence (Detroit, 1989)*; Sridharan, N.S., Ed.; Morgan-Kaufmann: San Mateo, California, 1989; pp. 762-767.

69. Chen, A.M.; Hecht-Nielsen, R. On the Geometry of Feedforward Neural Network Weight Spaces. In *Second International Conference on Artificial Neural Networks*; IEE: London, 1991; pp. 1-4.

70. Zupan, J.; Gasteiger, J. Neural networks: a new method for solving chemical problems or just a passing phase? *Anal. Chim. Acta*, **1991**, *248(1)*, 1-30.

71. Weiss, S. M.; Kulilowski, C. A. *Computer Systems That Learn*; Morgan-Kaufman: San Mateo, California, 1991.

72. Efron, B.; Gong, G. A Leisurely Look at the Bootstrap, the Jackknife, and Cross-Validation. *Amer. Stat.* **1983**, *37*, 36-48.

73. Efron, B. *The Jackknife, the Bootstrap and Other Resampling Plans*; SIAM: Philadelphia, 1982.

74. Stubbs, D. F. Multiple Neural Network Approaches to Clinical Expert Systems. *SPIE (International Society for Optical Engineering) Proceedings (April, 1990)*; **1990**, *1294*, 433-441.

75. Duda, R.O.; Hart, P.E. *Pattern Classification and Scene Analysis*; Wiley-Interscience: New York, 1973.

76. CA File database, Chemical Abstracts Service, American Chemical Society, Columbus, OH, 1967-1992.

77. INSPEC Database, IEEE, available on STN International, Chemical Abstracts Service, American Chemical Society, Columbus, OH, 1969-1992.

78. COMPUSCIENCE Database, FIZ Karlsruhe, Germany, available on STN International, Chemical Abstracts Service, American Chemical Society, Columbus, OH, 1982-1992.

79. Jurs, P. C.; Kowalski, B. R.; Isenhour, Thomas L.; Reilley, C. N. Computerized Learning Machines Applied to Chemical Problems. Convergence Rate and Predictive Ability of Adaptive Binary Pattern Classifiers. *Anal. Chem.*, **1969**, *41(6)*, 690-5.

80. Isenhour, T. L.; Jurs, P. C. Learning Machines. *Comput. Chem. Instrum.*, **1973**, *1*, 285-330.

81. Jurs, P. C. Binomial Distribution Statistics Applied to Minimizing Activation Analysis Counting Errors. An Analog Computer Controlled Gamma-Ray Spectrometer for Comparative Activation Analysis. Computerized Learning Machines for the Interpretation of Low Resolution Mass Spectrometry Data. 1969, Ph.D. dissertation, University of Washington, Seattle, WA, 95 pp.

82. Roat, S. D. The Application of Neural Networks and System Cultivation with a Nonlinear Optimal Control Algorithm for the Chemical Process Industry. 1991, Ph.D. dissertation, Univ. Tennessee, 261 pp.

83. Haesloop, D. G. System Identification and Control using Neural Networks with Combined Linear and Non-Linear Mapping Functionality. 1991, Ph.D. dissertation, Univ. Washington, 184 pp.

84. Bhat, N. D. System Identification and Control of Dynamic Chemical Processes Using Backpropagation Networks. 1991, Ph.D. dissertation, Univ. Maryland, 262 pp.

85. Lacy, M. E. Neural Network Technology and its Application in Chemical Research. *Tetrahedron Comput. Methodol.*, **1990**, *3(3-4)*, 119-28.

86. Kyuma, K. (1991) Optical Neural Networks - a Review. *Nonlinear Opt.*, **1991**, *1(1)*, 39-49.

87. Schmuller, J. Neural Networks and Environmental Applications. In *ACS Symp. Ser. 431, Expert Systems for Environmental Applications*; Hushon, J., Ed; American Chemical Society: Washington, D.C., 1990; 52-68.

88. Holcomb, T. R.; Morari, M. PLS/Neural Networks. *Comput. Chem. Eng.*, **1992**, *16(4)*, 393-411.

89. Qin, S. J.; McAvoy, T. J. Nonlinear PLS Modeling Using Neural Networks. *Comput. Chem. Eng.*, **1992**, *16(4)*, 379-91.

90. Kramer, M. A. Nonlinear Principal Component Analysis Using Autoassociative Neural Networks. *AIChE J.*, **1991**, *37(2)*, 233-43.

91. Livingstone, D. J.; Hesketh, G.; Clayworth, D. Novel Method for the Display of Multivariate Data Using Neural Networks. *J. Mol. Graphics*, **1991**, *9(2)*, 115-18.

92. Hoskins, J. C.; Himmelblau, D. M. Artificial Neural Network Models of Knowledge Representation in Chemical Engineering. *Comput. Chem. Eng.*, **1988**, *12(9-10)*, 881-90.

93. *Computers in Chemical Engineering*, **1992**, *16(4)*.

94. Bugmann, G.; Lister, J. B.; Von Stockar, U. Characterizing Bubbles in Bioreactors Using Light or Ultrasound Probes: Data Analysis by Classical Means and by Neural Networks. *Can. J. Chem. Eng.*, **1991**, *69(2)*, 474-80.

95. Venkatasubramanian, V.; Chan, K. A Neural Network Methodology for Process Fault Diagnosis. *AIChE J.*, **1989**, *35(12)*, 1993-2002.

96. Joseph, B.; Wang, F. H.; Shieh, D. S. S. Exploratory Data Analysis: a Comparison of Statistical Methods with Artificial Neural Networks. *Comput. Chem. Eng.*, **1992**, *16(4)*, 413-23.

97. Glick, M.; Hieftje, G. M. Classification of Alloys with an Artificial Neural Network and Multivariate Calibration of Glow-Discharge Emission Spectra. *Appl. Spectrosc.*, **1991**, *45(10)*, 1706-16.

98. Chang, S. M.; Iwasaki, Y.; Suzuki, M.; Tamiya, E.; Karube, I.; Muramatsu, H. Detection of Odorants Using an Array of Piezoelectric Crystals and Neural Network Pattern Recognition. *Anal. Chim. Acta*, **1991**, *249(2)*, 323-9.

99. Hoffheins, B. S. Using Sensor Arrays and Pattern Recognition to Identify Organic Compounds, Report ORNL/TM-11310; DE90013299, 67 pp. NTIS: Energy Res. Abstr. 1990, 15(17), Abstr. No. 39677.

100. Rogers, S. J.; Fang, J. H.; Karr, C. L.; Stanley, D. A. Determination of Lithology From Well Logs Using a Neural Network. *AAPG Bull.*, **1992**, *76(5)*, 731-9.

101. Bos, A.; Bos, M.; Van der Linden, W. E. Artificial Neural Networks as a Tool for Soft-Modeling in Quantitative Analytical Chemistry: the Prediction of the Water Content of Cheese. *Anal. Chim. Acta*, **1992**, *256(1)*, 133-44.

102. Long, J. R.; Mayfield, H. T.; Henley, M. V.; Kromann, P. R. Pattern Recognition of Jet Fuel Chromatographic Data by Artificial Neural Networks With Back-Propagation of Error. *Anal. Chem.*, **1991**, *63(13)*, 1256-61.

103. Zupan, J. Can an Instrument Learn From Experiments Done by Itself? *Anal. Chim. Acta*, **1990**, *235(1)*, 53-63.

104. Otto, M.; George, T.; Schierle, C.; Wegscheider, W. Fuzzy logic and Neural Networks - Applications to Analytical Chemistry. *Pure Appl. Chem.*, **1992**, *64(4)*, 497-502.

105. Stolorz, P.; Lapedes, A.; Xia, Y. Predicting Protein Secondary Structure Using Neural Net and Statistical Methods. *J. Mol. Biol.*, **1992**, *225(2)*, 363-77.

106. Muskal, S. M.; Kim, S. H. Predicting Protein Secondary Structure Content. A Tandem Neural Network Approach. *J. Mol. Biol.*, **1992**, *225(3)*, 713-27.

107. Friedrichs, M. S.; Goldstein, R. A.; Wolynes, P. G. Generalized Protein Tertiary Structure Recognition Using Associative Memory Hamiltonians. *J. Mol. Biol.*, **1991**, *222(4)*, 1013-34.

108. Ferran, E.A.; Ferrara, P. Topological Maps of Protein Sequences. *Biological Cybernetics*, **1991**, *65(6)*, 451-8.

109. Kohonen, T. *Self-organization and Associative Memory*; Springer: Berlin, 1988.

110. Sumpter, B. G.; Noid, D. W. Potential Energy Surfaces for Macromolecules. A Neural Network Technique. *Chem. Phys. Lett.*, **1992**, *192(5-6)*, 455-462.

111. Bohm, G. Protein Folding and Deterministic Chaos: Limits of Protein Folding Simulations and Calculations. *Chaos, Solitons and Fractals*, **1991**, *1(4)*, 375-82.

112. Hirst, J. D.; Sternberg, M. J. E. Prediction of Structural and Functional Features of Protein and Nucleic Acid Sequences by Artificial Neural Networks. *Biochemistry*, **1992**, *31(32)*, 7211-18.

113. Uberbacher, E. C.; Mural, R. J. Locating Protein-Coding Regions in Human DNA Sequences by a Multiple Sensor-Neural Network Approach. *Proc. Natl. Acad. Sci. USA.*, **1991**, *88(24)*, 11261-5.

114. Stormo, G. D.; Schneider, T. D.; Gold, L.; Ehrenfeucht, A. Use of the Perceptron Algorithm to Distinguish Translational Initiation Sites in *E. coli. Nucleic Acids Res.*, **1982**, *10(9)*, 2997-3011.

115. Arrigo, P.; Giuliano, F.; Scalia, F.; Rapallo, A.; Damiani, G. Identification of a New Motif on Nucleic Acid Sequence Data Using Kohonen's Self-Organizing Map. *Comput. Appl. Biosci.*, **1991**, *7(3)*, 353-7.

116. Ladunga, I.; Czako, F.; Csabai, I.; Geszti, T. Improving Signal Peptide Prediction Accuracy by Simulated Neural Network. *Comput. Appl. Biosci.*, **1991**, *7(4)*, 485-7.

117. Claverie, J. M.; Sauvaget, I. WOBB.C: A Portable Software Package for Defining and Searching Ambiguous Sequence Patterns. *Protein Sequences Data Anal.*, **1991**, *4(2)*, 119-21.

118. Aoyama, T.; Ichikawa, H. Reconstruction of Weight Matrices in Neural Networks--a Method of Correlating Outputs With Inputs. *Chem. Pharm. Bull.* **1991**, *39(5)*, 1222-1228.

119. Smits, J. R. M.; Breedveld, L. W.; Derksen, M. W. J.; Kateman, G.; Balfoort, H. W.; Snoek, J.; Hofstraat, J. W. Pattern Classification With Artificial Neural Networks: Classification of Algae, Based Upon Flow Cytometer Data. *Anal. Chim. Acta*, **1992**, *258(1)*, 11-25.

120. Zitko, V. Prediction of Biodegradability of Organic Chemicals by an Artificial Neural Network. *Chemosphere*, **1991**, *23(3)*, 305-12.

121. Chastrette, M.; De Saint Laumer, J. Y. Structure-Odor Relationships Using Neural Nets. *Eur. J. Med. Chem.*, **1991**, *26(8)*, 829-33.

122. Hussain, A. S.; Yu, X.; Johnson, R. D. Application of Neural Computing in Pharmaceutical Product Development. *Pharm. Res.*, **1991**, *8(10)*, 1248-52.

123. Rose, V. S.; Croall, I. F.; MacFie, H. J. H. An Application of Unsupervised Neural Network Methodology (Kohonen Topology-Preserving Mapping) to QSAR Analysis. *Quant. Struct.-Act. Relat.*, **1991**, *10(1)*, 6-15.

124. Munk, M. E.; Madison, M. S.; Robb, E. W. Neural Network Models for Infrared Spectrum Interpretation. *Mikrochim. Acta*, **1991**, *2(1-6)*, 505-14.

125. Borggaard, C.; Thodberg, H. H. Optimal Minimal Neural Interpretation of Spectra. *Anal. Chem.*, **1992**, *64(5)*, 545-51.

126. Kjaer, M.; Poulsen, F. M. Identification of 2D Proton NMR Antiphase Cross Peaks Using a Neural Network. *J. Magn. Reson.*, **1991**, *94(3)*, 659-63.

127. Curry, B.; Rumelhart, D. E. MSnet: A Neural Network Which Classifies Mass Spectra. *Tetrahedron Comput. Methodol.*,**1990**, *3(3-4)*, 213-37.

128. Otto, M.; Hoerchner, U. (1990) Application of Fuzzy Neural Network to Spectrum Identification, *Software Dev. Chem. 4, Proc. Workshop Comput. Chem.*, Meeting Date 1989, Ed: Gasteiger, J.; Springer: Berlin, 1990; p 377-84.

129. Anker, L. S.; Jurs, P. C. Prediction of Carbon-13 Nuclear Magnetic Resonance Chemical Shifts by Artificial Neural Networks. *Anal. Chem.*, **1992**, *64(10)*, 1157-64.

130. Kvasnicka, V. An Application of Neural Networks in Chemistry. Prediction of ^{13}C NMR Chemical Shifts. *J. Math. Chem.* **1991**, *6*, 63-76.

131. Kito, S.; Hattori, T.; Murakami, Y. Estimation of the Acid Strength of Mixed Oxides by a Neural Network. *Ind. Eng. Chem. Res.*, **1992**, *31(3)*, 979-81.

132. Bodor, N.; Harget, A.; Huang, M. J. Neural Network Studies. 1. Estimation of the Aqueous Solubility of Organic Compounds. *J. Am. Chem. Soc.*, **1991**, *113(25)*, 9480-3.

133. Brewster, M. E.; Huang, M. J.; Harget, A.; Bodor, N. Reactivity of Biologically Important Reduced Pyridines. 10. Neural Network Studies. 2. Use of a Neural Net to Estimate Oxidation Energies for Substituted Dihydropyridines and Related Heterocycles. *Tetrahedron*, **1992**, *48(17)*, 3463-72.

134. Peterson, K. L. Counter-Propagation Neural Networks in the Modeling and Prediction of Kovats Indexes for Substituted Phenols. *Anal. Chem.*, **1992**, *64(4)*, 379-86.

135. Hecht-Nielsen, R. *Neurocomputing*; Addison-Wesley Publishing Company: Redwood City, CA, 1990, p 147-153.

136. Glen, R. C.; Rose, V. S.; Lindon, J. C.; Ruane, R. J.; Wilson, I. D.; Nicholson, J. K. Quantitative Structure-Chromatography Relationships: Prediction of TLC behavior Using Theoretically Derived Molecular Properties. *J. Planar Chromatogr.-Mod. TLC*, **1991**, *4*, 432-8.

137. Noid, D. W.; Varma-Nair, M.; Wunderlich, B.; Darsey, J. A. Neural Network Inversion of the Tarasov Function Used for the Computation of Polymer Heat Capacities, *J. Therm. Anal.*, **1991**, *37(10)*, 2295-300.

138. Guo, Z.; Uhrig, R. E. Use of Artificial Neural Networks to Analyze Nuclear Power Plant Performance. *Nucl. Technol.*, **1992**, *99(1)*, 36-42.

139. Cherubini, A.; Odorico, R. Discrimination of p.hivin. p.fwdarw. t.mchlt.t Events by a Neural Network Classifier. *Z. Phys. C: Part. Fields*, **1992**, *53(1)*, 139-48.

140 Stimpfl-Abele, G. Recognition of Decays of Charged Tracks With Neural Network Techniques. *Comput. Phys. Commun.*, **1991**, *67(2)*, 183-92.

141. Peterson, K. L. Classification of Curium II and Plutonium I Energy Levels Using Counterpropagation Neural Networks. *Phys. Rev. A*, **1991**, *44(1)*, 126-38.

142. Munk, M.E. Symposium on Spectral Simulation. *J. Chem. Inf. Comput. Sci.*, **1992**, *32*, 263.

143. Aoyama, T.; Suzuki, Y.; Ichikawa, H. Neural Networks Applied to Quantitative Structure/Activity Relationships; *J. Med. Chem.* **1990**, *33*, 2583-2590.

144. *Neural Computing: NeuralWorks Professional II/Plus and NeuralWorks Explorer*; NeuralWare, Inc.: Pittsburgh, PA; 1991, p NC-177 - NC-193.

145. Moody, J.; Darken, C. Learning with Localized Receptive Fields. In *Proceeding of the 1988 Connectionist Summer School* (Pittsburgh, 1988); Touretzky, D.S.; Hinton, G.E.; Sejnowski, T.J., Eds.; Morgan-Kaufmann: San Mateo, California, 1988; pp. 133-143.

146. S-Plus statistical software, Version 3.0, Release 1 for Sun SPARC, Statistical Sciences, Inc. Seattle, WA, 1991.

147. Trinajstic, N. *Chemical Graph Theory, Vols. I,II*; CRC Press: Boca Raton, Florida, 1983.

148. Rippmann, F. Hydrophobicity and Tumor Promoting Ability of Phorbol Esters. *Quant. Struct.-Act. Relat.*, **1990**, *9*, 1-5.

149. Lawler, B. quoted in Bawden, D. Chemical Reaction Information. In *Chemical Structure Systems. Computational Techniques for Representation, Searching, and Processing of Structural Information.*; Ash, J. E.; Warr, W. A.; Willett, P., Ed.; Ellis Horwood: New York, 1991; Chapter 2, p 57.

150. Barnard, J. M. Structure Representation and Searching. In *Chemical Structure Systems. Computational Techniques for Representation, Searching, and Processing of Structural Information.*; Ash, J. E.; Warr, W. A.; Willett, P., Ed.; Ellis Horwood: New York, 1991; Chapter 1, p 9-56.

151. *Chemical Structure Systems. Computational Techniques for Representation, Searching, and Processing of Structural Information.*; Ash, J. E.; Warr, W. A.; Willett, P., Ed.; Ellis Horwood: New York, 1991.

152. Maggiora, G. M.; Johnson, M. A. Introduction to Similarity in Chemistry. In *Concepts and Applications of Molecular Similarity*; Johnson, M. A.; Maggiora, G. M., Ed.; Wiley-Interscience: New York, 1990; Chapter 1, p 1-13.

153. Johnson, M. A. Molecular Similarity Analysis Methods for Fuel Design. Submitted for publication, *The Design of Fuel Molecules*, Rouvray, D., Ed.

154. Randic, M. Design of Molecules with Desired Properties. A Molecular Similarity Approach to Property Optimization. In *Concepts and Applications of Molecular Similarity*; Johnson, M. A.; Maggiora, G. M., Ed.; Wiley-Interscience: New York, 1990; Chapter 5, p 77-145.

155. Randic, M. Dept. of Math and Computer Sci., Drake U., Des Moines, Iowa. Data calculated and provided to DWE in Nov., 1990.

156. Carey F.A.; Sundberg, R.J. *Advanced Organic Chemistry*; 2nd ed., Part A, Plenum Press: New York, 1984; pp 481-503.

157. De La Mare, P.D.B.; Ridd, J.H. *Aromatic Substitution: Nitration and Halogenation*; Academic Press: New York, 1959; pp 80-93, 236-237.

158. Harvey, D.R.; Norman, R.O.C. The Ortho:Para-Ratio in Aromatic Substitution. *J. Chem. Soc.*, **1961**, 3606-3611.

159. Hoggett, J. G.; Moodie, R. B.; Penton, J. R.; Schofield, K. *Nitration and Aromatic Reactivity*; Cambridge University Press: London, 1971; pp. 166-183. 166-183.

160. Norman, R. O. C.; Taylor, R. *Electrophilic Substitution in Benzenoid Compounds*; Elsevier Publishing Company: Amsterdam & New York, 1965; pp 72-85.

161. Patai, S. *The Chemistry of the Amino Group*; Wiley-Interscience: London, 1968; pp 250 - 257.

162. MOPAC, A General Molecular Orbital Program Package V4.0, QCPE, Department of Chemistry, Indiana University, Bloomington, IN, 47405.

163. CAMEO program, 1988 version. Available from CAMEO Group c/o Prof. W. L. Jorgensen, Sterling Chemistry Laboratory, Yale University, New Haven, CT 06511.

164. Dugundji, J.; Ugi, I. An Algebraic Model of Constitutional Chemistry as a Basis for Chemical Computer Programs. *Top. Curr. Chem.*, **1973**, *39*, 19-64.

165. Norman, R. O. C. *Principles of Organic Synthesis*; Chapman and Hall Ltd.: London; 2nd Ed., 1978; p. 83-85.

166. Titov, Y. A. Orientation in Diene Synthesis and its Dependance on Structure. *Russ. Chem. Rev.* **1962**, *31*, 267-283.

167. Feuer, J.; Herndon, W. C.; Hall, L. H. A Perturbational MO Method Applied to Diels-Alder Reactions with Unsymmetrical Dienes and Dienophiles: Prediction of the Major Product. *Tetrahedron*, **1968**, *24*, 2575-2582.

168. Norman, R. O. C. *Principles of Organic Synthesis*; Chapman and Hall Ltd.: London; 2nd Ed., 1978; p. 120-122.

169. Ugi, I.; Wochner, M.; Fontain, E.; Bauer, J.; Gruber, B.; Karl, R. Chemical Similarity, Chemical Distance, and Computer-Assisted Formalized Reasoning by Analogy. In *Concepts and Applications of Molecular Similarity*; Johnson, M. A.; Maggiora, G. M., Eds., Wiley: New York, 1990; Chapter 9, p 239-271.

170. Dengler, A.; Fontain, E.; Knauer, M.; Stein, S.; Ugi, I. Competing Concepts in CAOS. *Recl. Trav. Chim. Pays-Bas*, **1992**, *111*, 262-269.

171. Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. Learning Internal Representations by Error Propagation. In *Parallel Distributed Processing*; Rumelhart, D. E.; McClelland, J. L., Eds.; MIT Press: Cambridge, MA, 1986; p. 318-362.

172. Aoyama, T.; Ichikawa, H. Basic Operating Characteristics of Neural Networks When Applied to Structure-Activity Studies. *Chem. Pharm. Bull.*, **1991**, *39(2)*, 358-366.

173. Hendrickson, J. B. Descriptions of Reactions: Their Logic and Applications. *Recl. Trav. Chim. Pays-Bas*, **1992**, *111*, 323-334.

174. Klimasauskas, C. C. Hybrid Technologies: More Power for the Future. In *Advanced Technology for Developers*, 1, August 1992, High-Tech Communications: Sewickley, PA, 1992, 17-20.

175. Mavrovouniotis, M. L.; Chang, S. Hierarchical Neural Networks, *Comput. Chem. Eng.*, **1992**, *16(4)*, 347-69.

176. Ham, F.M.; Cohen, G.M.; Cho, B. Improved Detection of Biological Substances Using a Hybrid Neural Network and Infrared Absorption Spectroscopy. In *Conference Proceedings: IJCNN-91-Seattle: International Joint Conference on Neural Networks*, **1991**, *1*, 227-32.

177. D'Antone, I. A Neural Network in an Expert Diagnostic System. *IEE. Trans. Nucl. Sci.*, **1992**, *39(2, Pt. 1)*, 58-62.

178. Denker, J. quoted in Hertz, J.; Krogh, A.; Palmer, R. G. *Introduction To The Theory of Neural Computation*; Addison-Wesley Publishing Company: Redwood City, California, 1991, p 10.

179. Hertz, J.; Krogh, A.; Palmer, R. G. *Introduction To The Theory of Neural Computation*; Addison-Wesley Publishing Company: Redwood City, California, 1991, p 10.