University of Alberta

Artificial Neural Network – Advanced Theories and Industrial Applications

By

Qing James Zhang ©

A thesis submitted to the Faculty of graduate Studies and Research in partial fulfillment of the requirements for the degree of Doctoral of Philosophy

in

Environmental Engineering

Department of Civil and Environmental Engineering

Edmonton, Alberta

Fall, 2002

0-612-81291-X

University of Alberta

Library Release Form

Name of Author: Qing James Zhang

Title of Thesis: Artificial Neural Network - Advanced Theories and Industrial Applications

Degree: Doctor of Philosophy

Year this Degree Granted: 2002

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

13203 149 Avenue
Edmonton, Alberta
Canada, T6V 1L2

Date submitted: Sep 18, 2002

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled Artificial Neural Network – Advanced Theories and Industrial Applications submitted by Qing James Zhang in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Environmental Engineering.

Ian Buchanan
Dr. I. D. Buchanan

Daniel W. Smith
Dr. D. W. Smith

DC Sego
Dr. D. C. Sego

Dr. S. J. Stanley

Dr. J. Leonard

Dr. M. G. El-Din

Dr. M. J. Rodriguez

Thesis Approval Date: 13 Sept 2002

Abstract

This doctoral thesis presents the advances in three major research fronts of artificial neural network (ANN): a graphical mapping technique to interpret the internal activities of ANN model, an ANN hierarchical modeling protocol enhanced with a pattern recognition technique, and the actual industrial application and implementation of ANN technology. The advances in these three aspects of ANN technology interact and benefit each other.

The ability to interpret and manipulate internal workings of neural network is a major breakthrough in the ANN theoretical research. The author proves that most of the feed-forward neural networks are functions. Thus, many properties of functions can be applied to analyze neural network behaviors. Based on this proof, a graphical mapping technique was proposed to interpret the internal activities of ANN model. With this technique, it is possible to study the impact of noisy data on ANN modeling, and several key features of ANN models such as memorization, robustness, and sensitivity from the perspective of artificial neurons and their connection weights.

Based on the new knowledge of ANN models, a systematic modeling approach is proposed. Pattern recognition is integrated into the ANN modeling approach to provide additional capacities to analyze the source data in a noisy or complex study domain. The pattern recognition technique classifies the source data into many small modular domains to allow much more accurate modeling within smaller domains. This modeling approach

creates an ANN modeling system with a hierarchical structure, modular components, and an important built-in ability to automatically detect new features of a study domain, and thus prevent meaningless prediction on the out-of-bound data and maintain the integrity of the model prediction.

Finally, the improved ANN modeling approach is used to build two industrial applications: one in water treatment industry and the other one in the oil sand treatment process. The success of these applications not only complements the theoretical research, but also provides precious experience on how to implement ANN modeling system on-line for the industrial applications.

Acknowledgement

The author would like to gratefully acknowledge several public agencies and people from these agencies for their financial support to the completion of this thesis. The research is funded in part by EPCOR through Mr. David Rector from the Rossdale Water Treatment Plant, by the American Water Works Association Research Foundation through its research grants, by the National Science and Engineering Research Council through the National Science and Engineering Research Council Postgraduate Scholarship, by the Alberta Research Council though Dr. Ron Sawatzky and Dr. Dean Wallace, and by the University of Alberta through various scholarships such as the University of Alberta Dissertation Fellowship.

The author would like to thank Dr. Stephen J. Stanley for his effort to put all the funding together, for his guidance and support of the research, and being a wonderful friend of the family. The author would also like to thank Dr. Daniel W. Smith and Dr. Gordon R. Finch for their interest and support of the research and many wonderful suggestions and recommendations in my candidacy examination.

The author also extends my sincerest thanks to the following people:

1. Mr. David Rector for being an original and non-stop advocate for the ANN research in water treatment.

2. Mr. Riyaz Shariff and Ms. Audrey A. Cudrak from EPCOR Water Services for not only being the co-authors of the paper (Chapter II), but also their wonderful day to day support in data collection, ANN modeling refinement, ANN model on-line implementation, and tireless review and improvement of the paper.

3. Dr. Ron Sawatsky, Dr. Dean Wallace and Mr. Mike London from Alberta Research Council for being wonderful co-authors of the paper (Chapter IV), their contribution of the inside knowledge of the oil sand treatment process, and the wonderful experience in the heated discussion of the pattern recognition modeling approach.

4. Mr. Gary Kovacik for his active participation of the oil sand treatment modeling project.

Finally, I would like to thank my wife Ge Gail Li, and my parents, Sumei Huang and Rongkun Zhang for their wonderful and constant support. I was able to devote most of my time on studying and writing this thesis without interference. Without their support, there would be no thesis.

Table of Contents

CHAPTER I INTRODUCTION

CHAPTER II OVERVIEW OF THE ARCHITECTURE AND MODELING PROTOCOL OF ARTIFICIAL NEURAL NETWORK

CHAPTER III INTERNAL WORKINGS OF ANN MODELS

CHAPTER IV DESIGN ARTIFICIAL NEURAL NETWORK SYSTEMS FOR COMPLEX DOMAINS — A PATTERN RECOGNITION APPROACH

CHAPTER V UPGRADING ANN MODEL FOR REAL-TIME APPLICATION

- A CASE STUDY TO IMPLEMENT ANN RAW WATER COLOUR FORECASTING

MODEL IN THE ROSSDALE WATER TREATMENT PLANT

CHAPTER VI ARTIFICIAL NEURAL NETWORK MODELING OF OIL SANDS

EXTRACTION PROCESSES

List of Tables

List of Figures

List of Symbols and Abbreviations

Symbol	Definition
1D	one-dimensional
2D	two-dimensional
ANN	artificial neural network
$f()$	a function
LEE	low energy extraction
mM	milli Mole
$P_1, P_2, \ldots, P_n$	n data patterns
R^2	the coefficient of multiple determination
$S()$	summation function
W_o	the weight on the output connection
$W_1, W_2, \ldots, W_n$	weights on the input connections
W_t	the threshold of the artificial neuron
$X_1, X_2, \ldots, X_n$	n outside stimuli
$X_1', X_2', \ldots, X_n'$	n transitional stimuli inside neural network
X^m	the mth pattern in the cluster
$\|X-Y\|_2$	Euclidean distance between two patterns X and Y
$Y_1, Y_2, \ldots, Y_n$	outputs from n neurons
Z_n	the center of the nth data cluster

Chapter I Introduction

Although artificial neural networks (ANN) technique first emerged in early 1940s, its research and development progressed slowly for the next fifty years. Since 1990s, with the significant improvement of computing power in personal computers, the applications of ANN modeling have been dramatically increased in pace and spread into many fields of finance, science and engineering.

In the water treatment industry, there are many recent ANN applications burgeoning with encouraging results. The most recent successful applications of ANN include: water demand and consumption forecasting (Zhang, *et al*, 2001; Stark, *et al*, 2000), ANN direct process control (Baxter, *et al*, 2001; Zhang, *et al*, 1999), prediction of water main breaks (Sacluti, *et al*, 2000), modeling of coagulant dosages (Mirsepassi, *et al*, 1995; Gagnon, *et al*, 1997; Han, et al, 1997), modeling of filtration performance (Tupas, *et al*, 2000; Conlin, *et al*, 1997), chlorination dosage and control (Rodriguez, *et al*, 1996; Rodriguez, *et al*, 1997), modeling of enhanced coagulation (Stanley, *et al*, 1997; Stanley *et al*, 2000; Baxter *et al*, 1999), and water quality parameter prediction (Maier, *et al*, 1996; Zhang, *et al*, 1997). ANN Modeling has become one of the best among available technologies for modeling water treatment processes and applications.

Even though there are many successful applications of ANN, there is a perception that artificial neural networks are black box type models. This perception has negatively impacted user confidence and the general acceptance of ANN technology. The lack of understanding of the internal activities of ANN models has slowed the advance of ANN modeling techniques and their practical applications. Thus, it is very important to find a way to open up the ANN model to understand the ANN modeling activities at its fundamental level.

This thesis describes the author's attempt to accomplish such a task. A graphical mapping technique will be introduced to open the ANN model and analyze the activities inside the model. With this new knowledge of ANN inner activities, a modular hierarchical modeling structure is proposed to model the complex and noisy domains. This new modeling approach applies the pattern recognition technique to divide the complex domain into smaller and cohesive modules, models the modules individually, and thus results in much higher modeling accuracy for complex study domains. Subsequent chapters of the thesis demonstrate two ANN applications that benefited from this new modeling technique.

To properly introduce the graphical mapping technique, Chapter 2 provides an overview of the structure and the functionality of basic components of artificial neural network. Most of the discussion is focused on the artificial neuron since it is the building block of neural networks. The discussion is further extended to cover the basic principles of ANN learning especially the back-propagation algorithm, because it is the most common learning algorithm used in the ANN applications. The last section of the chapter provides a quick review of the ANN modeling protocol. Generally, the modeling protocol evolves with our understanding of the inner working of an ANN model. A thorough understanding of the ANN architecture and learning algorithm helps to explain each step of the modeling protocol, and provides opportunities to improve upon.

The concept of graphical mapping is presented in detail in Chapter 3. First, it is critical to prove that all feed-forward neural networks can be treated as functions. Based on this proof, the graphical mapping technique is then used to illustrate the principles of how to manipulate the values of connection weights and thresholds to change the behaviors of artificial neurons.

To further demonstrate the capacity of the graphical mapping technique, several case studies were performed to analyze the collective behaviors of neural network such as the impact of noise, memorization, robustness, sensitivity, and boundary extrapolation. The purpose of the case studies is to understand these features fundamentally at the level of the artificial neuron. With these new understandings, it is possible to directly manipulate the internal structure of neural network to suit special modeling purpose.

An improved ANN modeling strategy is proposed in Chapter 4 to handle noisy or complex domains. Noisy domains are domains in which noise and true data are so well mixed it is difficult to separate them with statistical approaches. Complex domains relate to domains in which several very distinct groups or sub-domains exist under the original study domain. For both types of study domains, it is difficult to obtain reasonably accurate modeling results with a single neural network model. Thus, there is a need for using multiple neural networks to accurately model the whole domain. To integrate multiple ANN models into a tight and fully functional system, the author proposes a pattern recognition approach.

The pattern recognition approach uses a data classifier to separate the original complex domain into smaller and more cohesive domains. This allows the development of a specific neural network model for each modular sub-domain. In actual application, the data classifier detects which sub-domain the incoming data belong to and subsequently redirect the incoming data to the ANN model designated for that sub-domain for a prediction. The data classifier is also designed to recognize new features, which are significantly different from any known features of the existing domain in a fresh data set. This automatic recognition system can be used to alarm the data that are outside the existing domain, where the ANN models were trained and tested, for special attention.

Chapter 5 is an application of the pattern recognition technique. This technique is used to upgrade an ANN model previously built for river color forecasting into a hierarchical structure of multiple ANN models to improve the modeling accuracy. This project demonstrates that dynamic changes in a study domain will introduce new features into the domain. Thus, to maintain the integrity of ANN models, it is necessary to upgrade the ANN models to learn the new features. The modular model structure with the pattern recognition approach makes the upgrade task much easier. Chapter 5 also discusses the issues related to the implementation of ANN models on-line in the actual operation environment. To materialize the full power of ANN applications, not only the models should be accurate, but also the users of ANN models should be properly educated. It is important to note that Chapter 5 is an early application of the advanced ANN theories discussed in this thesis, which is still evolving at the time.

Chapter 6 demonstrates the full capability of the improved ANN modeling protocol by applying it to model the oil sand treatment process. This was a collaborative project between the University of Alberta and the Alberta Research Council to understand the oil sand treatment process behaviors. Previously several modeling approaches had been attempted on the same database and yielded few significant results. With the new ANN modeling protocol, the project was completed in three months with very satisfactory results.

Finally, Chapter 7 links various aspects of the ANN modeling technique together and summarizes the significant findings.

References

Baxter, C. W., Stanley, S. J. and Zhang, Q. (1999), Development of a Full-Scale Artificial Neural Network Model for the Removal of Natural Organic Matter by Enhanced Coagulation, *AQUA*, 48(4): 129-136.

Baxter, C. W., Zhang, Q., Stanley, S. J., Shariff, R., Tupas, R-R., and Stark, H. L., (2001), Drinking Water Quality and Treatment: The Use of Artificial Networks, *Canadian Journal of Civil Engineering,* 28 (suppl. 1): 26-35.

Conlin, J., Peel, C. and Montague, G. A. (1997), Modeling of Pressure Drop in Water Treatment, *Artificial Intelligence in Engineering,* 11(4): 393-400.

Gagnon, C., Grandjean, B. P. A. and Thibault, J. (1997), Modeling of Coagulant Dosage in a Water Treatment Plant, *Artificial Intelligence in Engineering,* 11(4): 401-404.

Han, T. -H., Nahm, E. -S., Woo, K. -B., Kim, C. J. and Ryu, J. -W. (1997), Optimization of Coagulant Dosing Process in Water Purification System, in *Proc. of the 36th Annual Society of Instrument and Control Engineers Conference,* Tokushima, Japan: SICE.

Maier, H. R. and Dandy, G. C. (1996), The Use of Artificial Neural Networks for the Prediction of Water Quality Parameters, *Water Resources Research,* 32(4): 1013-1022.

Mirsepassi, A., Cathers, B. and Dharmappa, H. B. (1995), Application of Artificial Neural Networks to the Real Time Operation of Water Treatment Plants, in *Proc. of the 1995 IEEE International Conference on Neural Networks*. Perth, Australia: IEEE.

Rodriguez, M. J. and Serodes, J. B. (1996), Neural Network-Based Modeling of the Adequate Chlorine Dosage for Drinking Water Disinfection, *Canadian Journal of Civil Engineering,* 23(3): 621-631.

Rodriguez, M. J., Serodes, J. B. and Cote, P. A. (1997), Advance Chlorination Control in Drinking Water Systems Using Artificial Neural Networks, *Water Supply*, 15(2): 159-168.

Sacluti, F., Stanley, S. J. and Zhang Q. (2000), Applying Artificial Neural Networks to Predict Cast Iron Water Main Breaks, in *Proc. of the Canadian Society for Civil Engineering 2000 Conference*. London, Ont.: CSCE.

Stark, H. L., Stanley, S. J., and Buchanan, I. D. (2000), The Application of Artificial Neural Networks to Water Demand Modeling, in *Proc. of the Canadian Society for Civil Engineering 2000 Conference*. London, Ont.: CSCE.

Stanley, S. J., Baxter, C. W., Zhang, Q. and Shariff, R. (2000), *Process Modeling and Control of Enhanced Coagulation,* American Water Works Association Research Foundation, Denver, Colorado, Pp. 146.

Stanley, S. J. and Zhang, Q. (1997), The Use of Artificial Neural Networks for Process Modeling of Enhanced Coagulation, in *Proc. of the Water Quality Technology Conference,* American Water Works Association, Denver, Colorado.

Tupas, R-R. T., Stanley, S. J., Zhang, Q. (2000), Artificial Neural Network Modeling of Filtration Performance, in *Proc of Canadian Society for Civil Engineering 2000 Conference*, London, Ontario, Canada.

Zhang, Q. and Stanley, S. J. (1999), Automating Water Treatment Plants by the Artificial Neural Network Approach, *Proceedings of the American Water Works Association 1999 Engineering and Construction Conference*, Orlando, Florida.

Zhang, Q. and Stanley, S. J. (1997), Forecasting Raw-water Quality Parameters for the North Saskatchewan River by Neural Network Modeling. *Water Research* 31(9): 2340-2350.

Zhang, Q., Stanley, S. J., Greenaway, G., Zenari, L., and Ancel, S. (2001), Water Consumption Revenue Forecasting With Artificial Neural Network Technology, Proceedings of 2001 Information Management and Technology Conference, published by AWWA, Denver, Colorado.

Chapter II Overview of the Architecture and Modeling Protocol of Artificial Neural Network

Introduction

The objective of this chapter is to provide an overview of the fundamental components of artificial neural networks. The overview will focus on the ANN architecture and the ANN modeling protocol. Clear understanding of these fundamental concepts of ANN is required for further discussion and analysis of the inner workings of ANN models presented in the next chapter.

The architecture of an artificial neural network is usually referred to a group of artificial neurons being interconnected into a networking structure as shown in Figure 2-1. The basic components of this networking structure are the artificial neurons and the interconnection links. Each artificial neuron is a process unit that is capable of processing and storing information. Each link is capable of enhancing, suppressing, or blocking the information feeding through it. An artificial neuron consists of input links, summation function, activation function, and output link. This chapter will examine these basic compositions of an artificial neuron in details.

Since ANN modeling is a complex process that can be both dynamic and iterative at the same time, a modeling protocol establishes a common framework to ensure the quality of ANN models. The protocol usually covers the full spectrum of modeling from source data analysis to evaluation of final ANN models. In general, the ANN modeling protocol includes source data analysis, model development, and model evaluation and fine-tuning. The ANN protocol evolves overtime with the development in the ANN modeling techniques. Presented in this chapter is a general modeling protocol that has been successful in many applications. In latter chapters, this general protocol will be expanded to include new developments found in this research which includes pattern recognition technique, study domain subdivision and modularization, and hierarchical modeling structure.

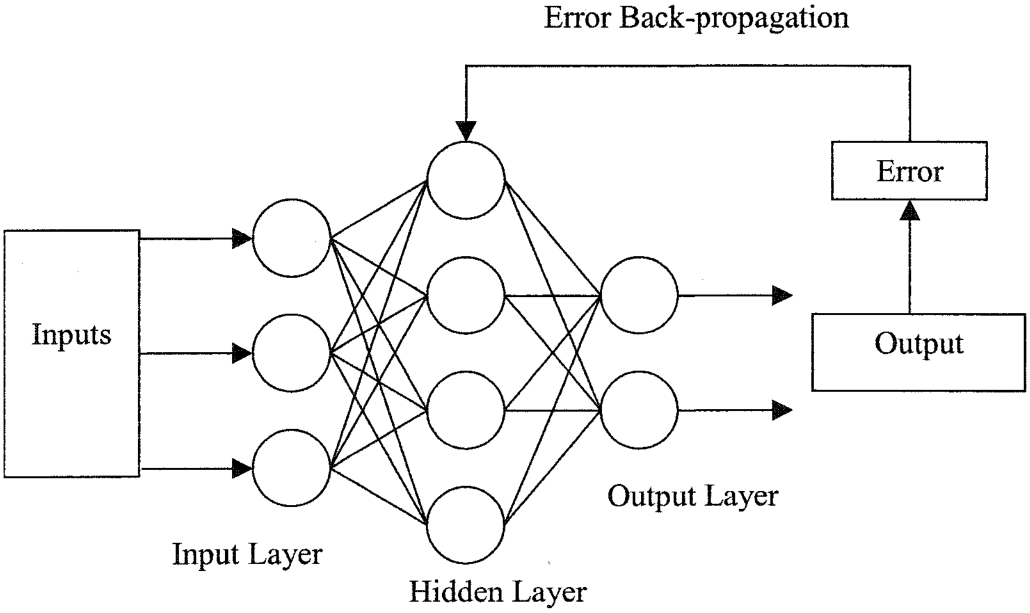

Figure 2-1 A Simple Artificial Neural Network

ANN Architecture

ANN modeling is an artificial intelligence method that mimics human brain's problem solving ability. ANN models are unique in that they can learn from the historical data presented to them, and subsequently to produce correct or nearly correct responses when presented with partially incorrect or incomplete stimuli (Rajasekaran, 1996; Maren, 1990).

These capabilities of neural networks are accomplished through their architecture and learning algorithms.

An Artificial Neuron

The basic building block of an artificial neural network is an artificial neuron. As shown in Figure 2-2, a typical artificial neuron consists of three components: input links (W_1, W_2, ..., W_n), a processing unit (the circle), and an output link (W_o). The processing unit is divided into two components. The side connected to the input links has a summation function and a threshold (W_t), which is a negative resident weight of the neuron. The summation function sums up the inputs and the threshold, and feeds the sum into the other side of neuron that has an activation function to produce an output stimulus. This output stimulus is then communicated to outside through the output link(s). These processes can be expressed in Equation 2-1 and 2-2.

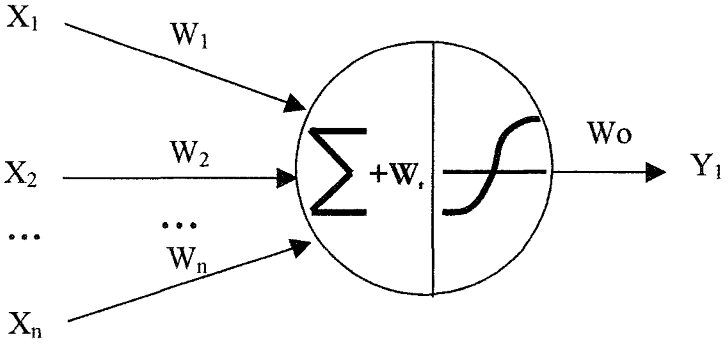

Figure 2-2 A Typical Artificial Neuron

Summation function: $S(X_1, X_2, \ldots, X_n) = X_1 \bullet W_1 + X_2 \bullet W_2 + \ldots + X_n \bullet W_n + W_t$ (2-1) where $X_1, X_2, \ldots, X_n$ are n outside stimuli;

$W_1, W_2, \ldots, W_n$ are weights on the input connections;

W_t is the threshold.

Activation function: output $Y_1 = W_o * f(S(X_1, X_2, \ldots, X_n))$ (2-2)

where W_o is the weight on the output connection.

A Simple Artificial Neural Network

The interconnections of these artificial neurons form an artificial neural network as shown in Figure 2-1. Within the network, artificial neurons are organized into an input layer, a hidden layer (or layers) and an output layer. The functions of neurons in each layer are different and specialized.

The neurons in the input layer are called input neurons. The input neurons receive inputs from external sources, scale the inputs to a proper range such as –1 to 1 or 0 to 1 for consistency in internal calculation and comparison, and then pass the scaled inputs to the hidden neurons in the next layer. Notice that the input neuron does not have activation function. Hidden neurons have activation functions, thus, they are the main process units of neural network. They allow the network to detect and capture features and patterns in the learning data and to perform the non-linear mapping between input and output variables (Zhang and Stanley, 1997). Output neurons receive input from the last hidden layer. Output neurons have both an activation function and a scale function. The activation function combines the hidden layer information together, and sends the activation results to the scale function, which scales the results back to normal output range and sends the results to external sources.

Activation Functions

The activation function is a key component of an artificial neuron and is required both in the hidden layer and the output layer. The basic business of activation function is to determine whether there should be any response from this neuron upon outside stimulus, and if there is a response, what magnitude of the response should be. The response curves of different activation functions are discussed in detail in Chapter 3.

There are many choices for activation function. The commonly used activation functions are listed in Table 2-1. Among them, the logistic function tends to be the most widely used. However, for each individual problem there might be a specific activation function that will work best. The corresponding graphical representation of each of these functions can be found in Figure 2-3 to Figure 2-10.

Table 2-1 Activation Functions

Activation Function	Equation	Range of Mapping
Logistic	$f(x) = 1/(1+e^{-x})$	0,1
Linear	$f(x)=x$	0,1 or -1,1
Tanh	$f(x)=\tanh(x)$	-1,1
Tanh1.5	$f(x)=\tanh(1.5x)$	-1,1
Sine	$f(x)=\sin(x)$	-1,1
Symmetric logistic	$f(x)=(2/(1+e^{-x}))-1$	-1,1
Gaussian	$f(x)=e^{-x^\wedge 2}$	0,1
Gaussian complement	$f(x)=1- e^{-x^\wedge 2}$	0,1

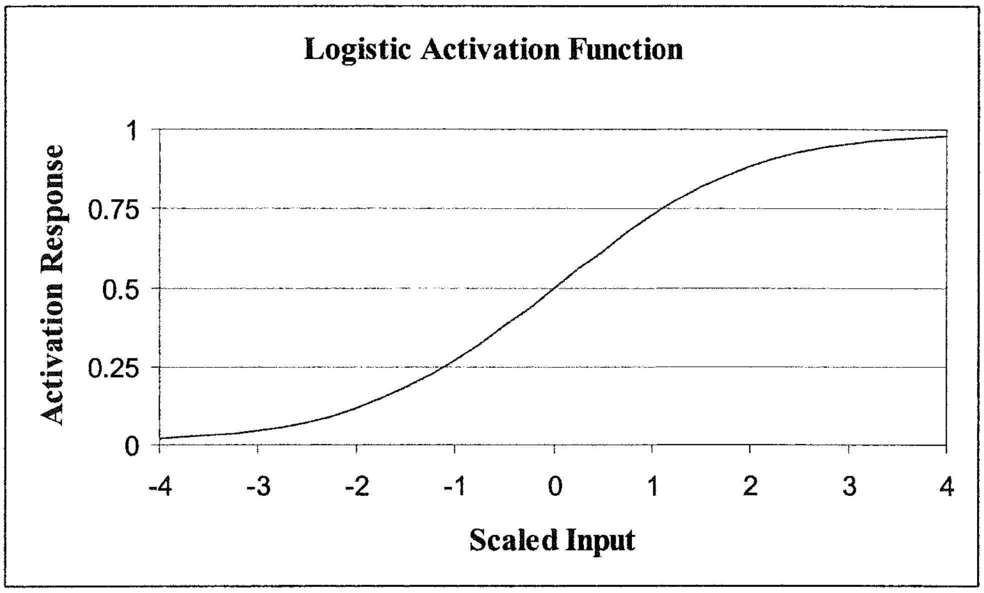

Figure 2-3 Logistic Activation Function

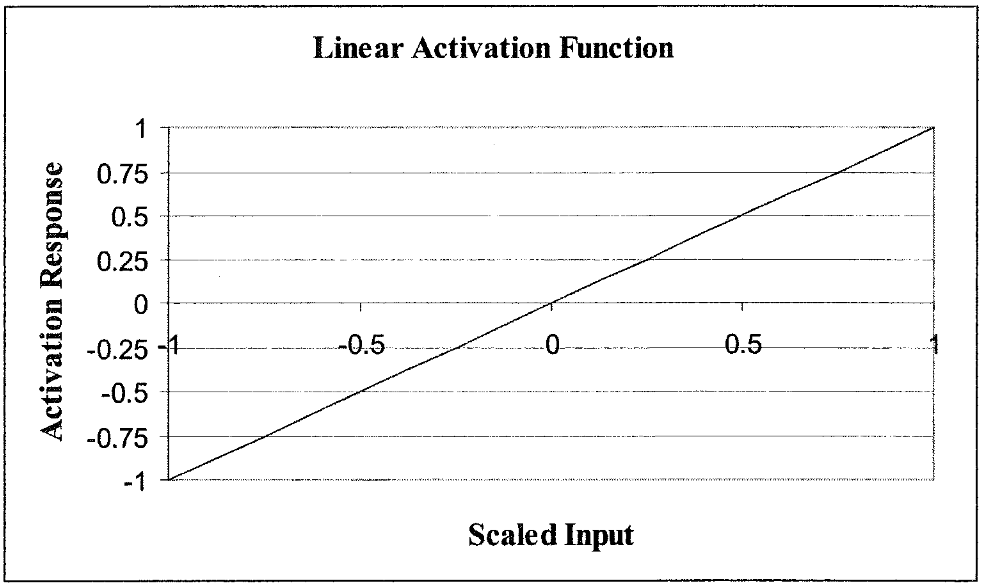

Figure 2-4 Linear Activation Function

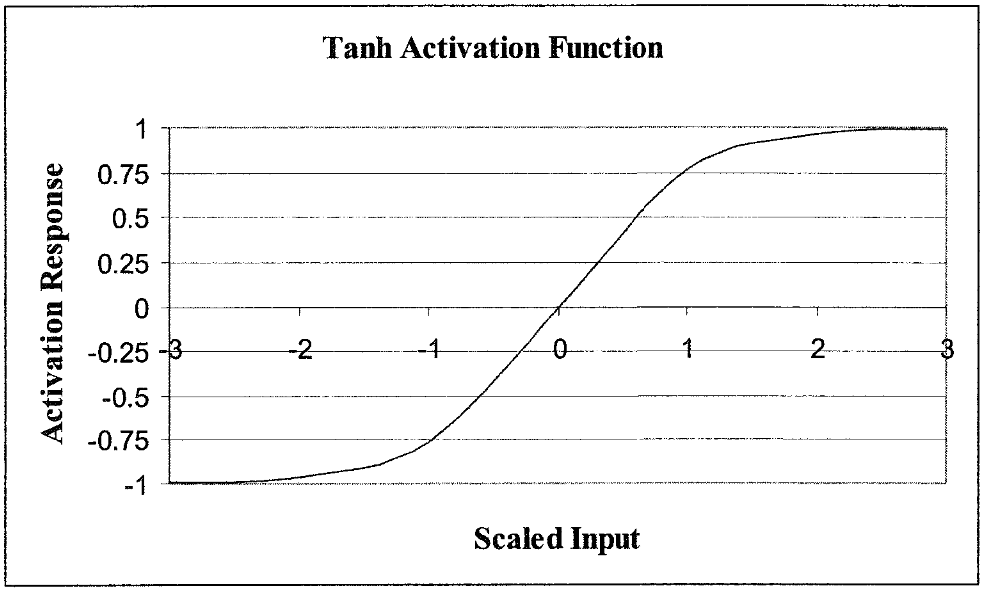

Figure 2-5 Tanh Activation Function

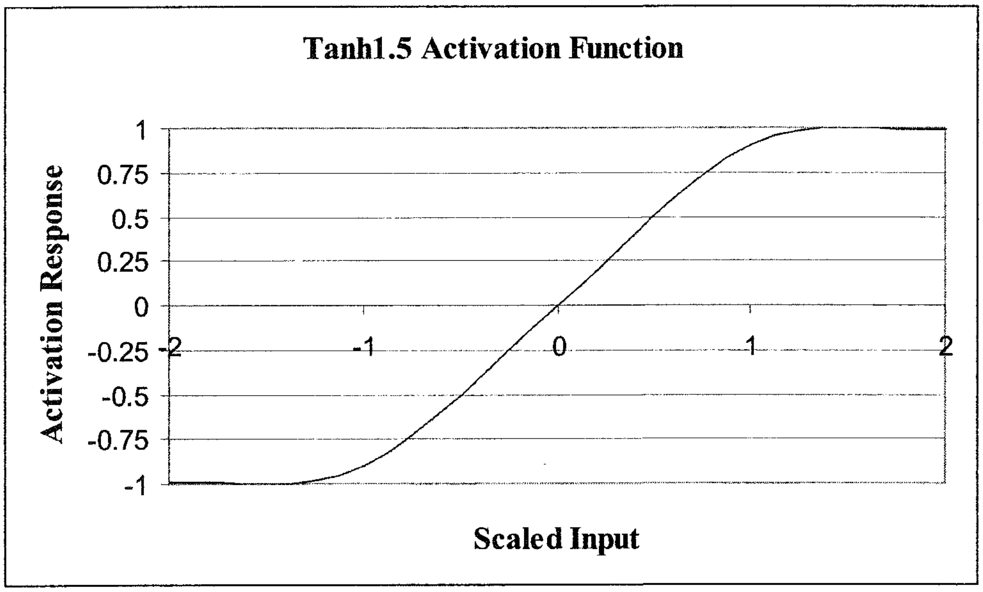

Figure 2-6 Tanh1.5 Activation Function

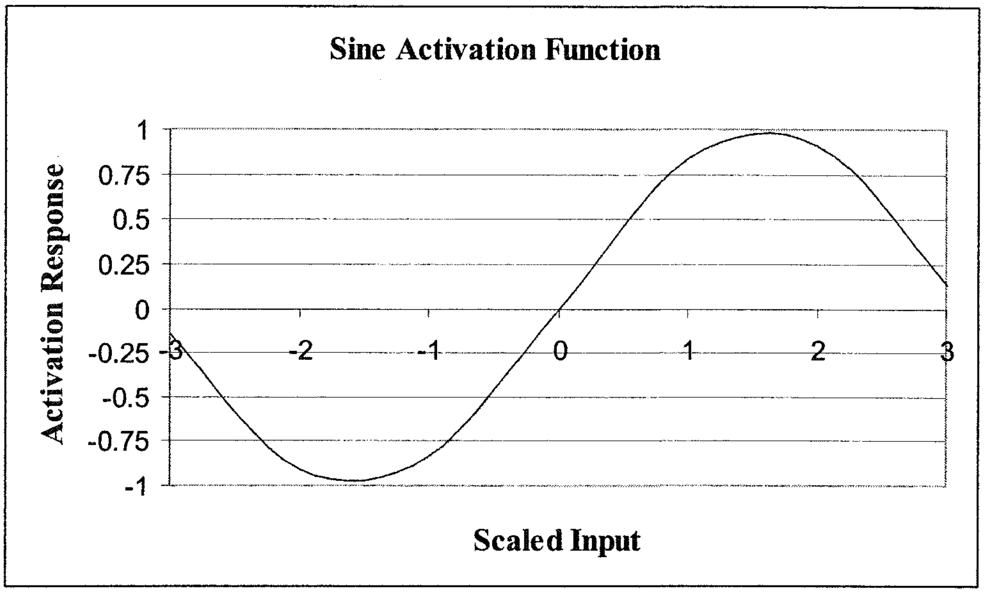

Figure 2-7 Sine Activation Function

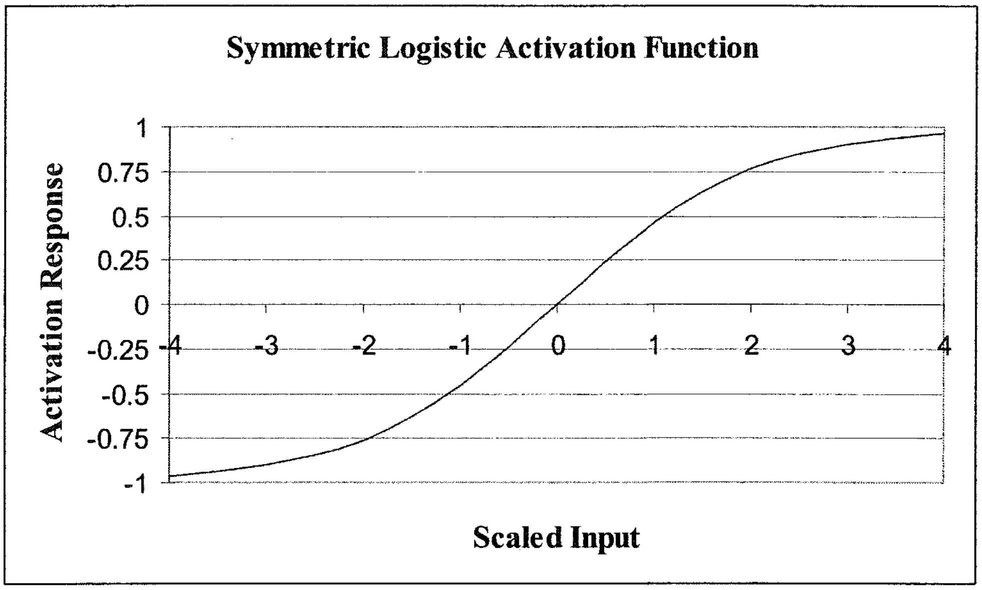

Figure 2-8 Symmetric Logistic Activation Function

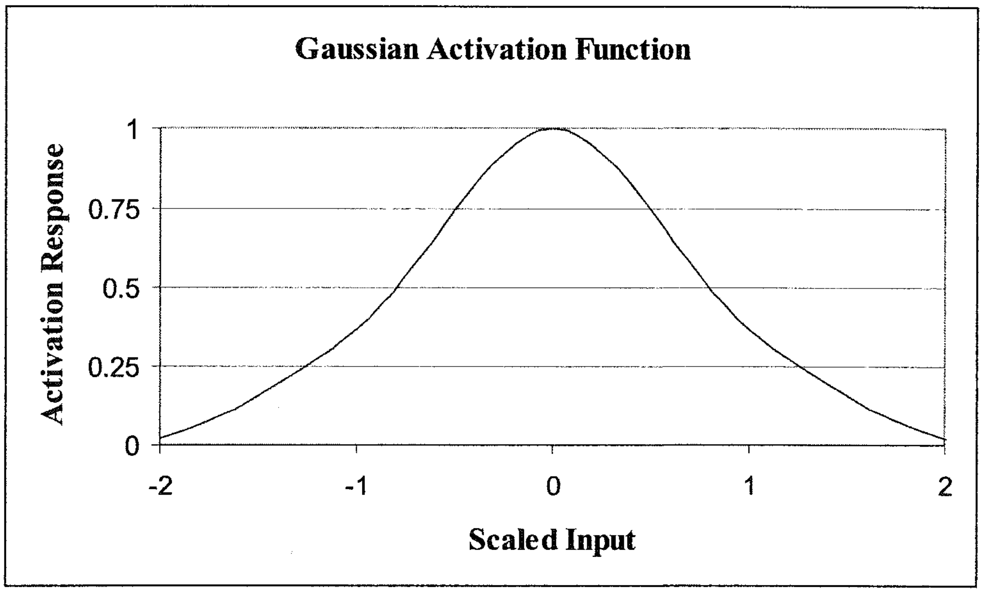

Figure 2-9 Gaussian Activation Function

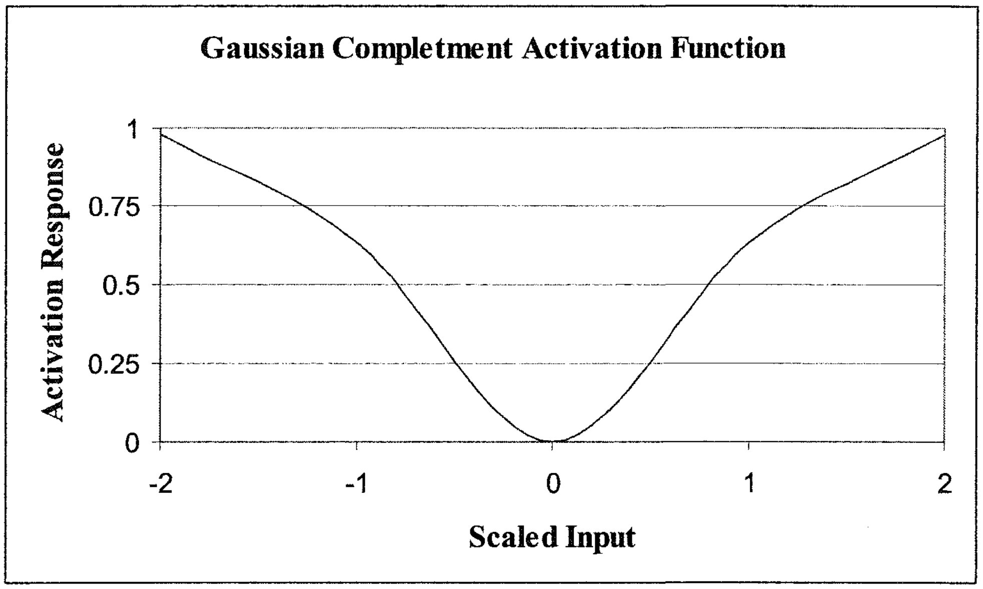

Figure 2-10 Gaussian Complement Activation Function

Scaling Function

Typically there are two scaling functions within ANN architecture. One scaling function is used to scale the input values to an optimal range for which neurons can operate on. Another scaling function is used to scale the activation from an output neuron back into a normal output value range. Typically the inputs are scaled between 0 and 1 or –1 and 1. Two types of scaling function are used: linear or non-linear.

Linear scaling proportionally scales the values into a specific range. Linear scaling can not condense an open-ended input data range into a closed-ended data range used by most neurons. If there is a new data point outside of the known ranges of an input parameter, there are two approaches to scale this data point. One is the straightforward linear scaling. The other one is the data truncation approach, in which this new data point is forced to take on a value equal to either minimum or maximum of the known data range depending on where it is located. Truncation of data means the modeling response is flat in the truncated region (same response as those of the minimum or maximum range value).

Most non-linear scaling covers the full data range, and there is no truncation of the data range. With non-linear scaling, the data in the middle range benefit most because they fall into the most active zone of the activation function. The data at the tail ends of original data range might fall into the saturation zone of the activation function, and become inactive. Non-linear scaling functions are useful to minimize the effect of outliers, but in doing so also minimize the importance of extreme events, which may be of interest. The tanh and logistic scaling functions are the commonly used non-linear scaling functions. The tanh function scales the data to (-1,1) using the formula:

$$f(\text{value})=\tanh((\text{value}-\text{mean})/\text{standard deviation}) \qquad (2\text{-}3)$$

The logistic scaling function scales the data to (0,1) using the formula:

$$f(\text{value})=1/(1+e^{-(\text{value}-\text{mean})/\text{standard deviation}}) \qquad (2\text{-}4)$$

It is important to note that in order to use both Equation 2-3 and Equation 2-4 properly, the input parameter should have a normal distribution. Thus, if the base distribution of an input parameter is not a normal distribution, proper transformation is required such as applying log to data that are log-normally distributed.

The most commonly used scaling function is the linear scaling without truncation, because it is simple and easy to understand. Scaling functions have limited impact on the whole modeling process.

ANN Learning Process

The learning algorithm is the method which neural networks use to learn from historical records. There are a variety of learning algorithms available. Among them, the back-propagation algorithm is the most commonly used. Thus, the back-propagation algorithm is discussed in details in this chapter. The learning algorithm for the Kohonen neural network is discussed in Chapter 4.

The birth of the back-propagation algorithm has an interesting history cited by Anderson and Rosenfeld (1988). In 1962, S. E. Dreyfus first proposed the back-propagation algorithm. In 1974, Werbos proposed a similar solution to learn multi-layer networks in his Ph.D. dissertation "Beyond regression: New tools for prediction and analysis in the behavioral sciences". Unfortunately, these solutions by Dreyfus and Werbos remained unknown to the neural network community. In 1986, it was Rumelhart, Hinton and Williams that independently described similar solutions to this problem, reached the largest audience, and popularized the solution of the back-propagation algorithm. Since then the back-propagation algorithm has been the most widely used algorithm for multi-layer networks.

The general concept of the back-propagation algorithm is described below. It is usually implemented by software in an iterative fashion. Consider a neural network with an input layer, an output layer and one hidden layer as shown in Figure 2-1. Once the network receives a data record, the input neurons process this record and pass the processed information to the hidden layer. The neurons in the hidden layer adjust themselves by the method of least square error to learn the record. The result of this learning process is passed to the output neuron to present it to the external world. This learning result is then compared to the desired result. If the error is not small enough, the learning process in the hidden layer is continued until satisfactory results are obtained. The detailed mathematical equations and background of the back-propagation learning algorithm are available in many neuron network textbooks. It is also available in the Appendix I of the author's Master thesis (Zhang, 1996).

ANN Modeling Protocol

ANN modeling is a complex process. The main reason for this complexity in modeling is actually due to the fact that there are many components in an ANN model that can be modified. Each adjustment in a component results in a different ANN model. In other words, too much flexibility in the modeling actually increases the difficulty to find the right path to the correct solution. The ANN modeling process is dynamic because there are many steps in the modeling process and there are decisions to make in each step. Each decision may affect the subsequent direction of the modeling process, and could result in a completely different ANN model. The modeling process is also iterative, mostly due to its repetitive learning algorithm such as the error back-propagation algorithm.

To properly guide the modeling process, there is a need for a standard approach to establish a common framework for modeling. The author first proposed a general ANN modeling protocol in his master thesis (Zhang, 1996). This modeling protocol evolved over time and in general, it can be divided into three major phases: source data analysis, model development, and model evaluation and fine-tuning. In the source data analysis phase, designers collect and analyze any necessary data to assess the feasibility of the modeling project. After the feasibility study, the designers select several architecture designs of potential models, test the design ideas, and build a pool of candidate models. Finally in the model evaluation and fine-tuning phase, designers select the best candidate model, fine-tune it, and eventually implement it.

Source Data Analysis

In order to select the best data for modeling, a thorough analysis of the data must be performed. A typical step by step protocol for the source data analysis is presented in Figure 2-11. It is very important to assess whether a process is suitable for ANN modeling. The assessment includes evaluating whether the objectives of the project can be achievable by ANN modeling, and if so, how the project should proceed. Another important aspect of the feasibility assessment is the domain study. A typical domain study includes input and output parameter selection. For each potential input or output parameter, the data accuracy and frequency must be validated. A descriptive statistical analysis should also be used to identify the mean, standard deviation, distribution, and range of the data, etc. The statistical analysis is also useful for identifying data outliers, erroneous entries, and non-entries within the data set. A recent development in the domain study is to apply the pattern recognition techniques to delimitate and subdivide the study domain into modular and easy to model sub-domains. This technique will be further described in Chapter 4.

Another important issue in the source data analysis phase is active user involvement (Zhang, *et al*, 2001). Since artificial neural network appears as a "black box" type model to users, there is no direct mechanism to show its reasoning, and thus it is difficult to gain users' trust at the beginning. It is also easy for the users to lose confidence in the model if its performance deviates from the intended goal.

In order to promote user ownership and a better understanding of ANN models, the users of ANN models should be invited to become involved in the ANN model building process as early as possible, and as much as possible. Direct involvement at the early stage of the modeling allows users to express their concerns, encourages them to provide input into the overall design, and provides a clear understanding of the capacity of ANN models.

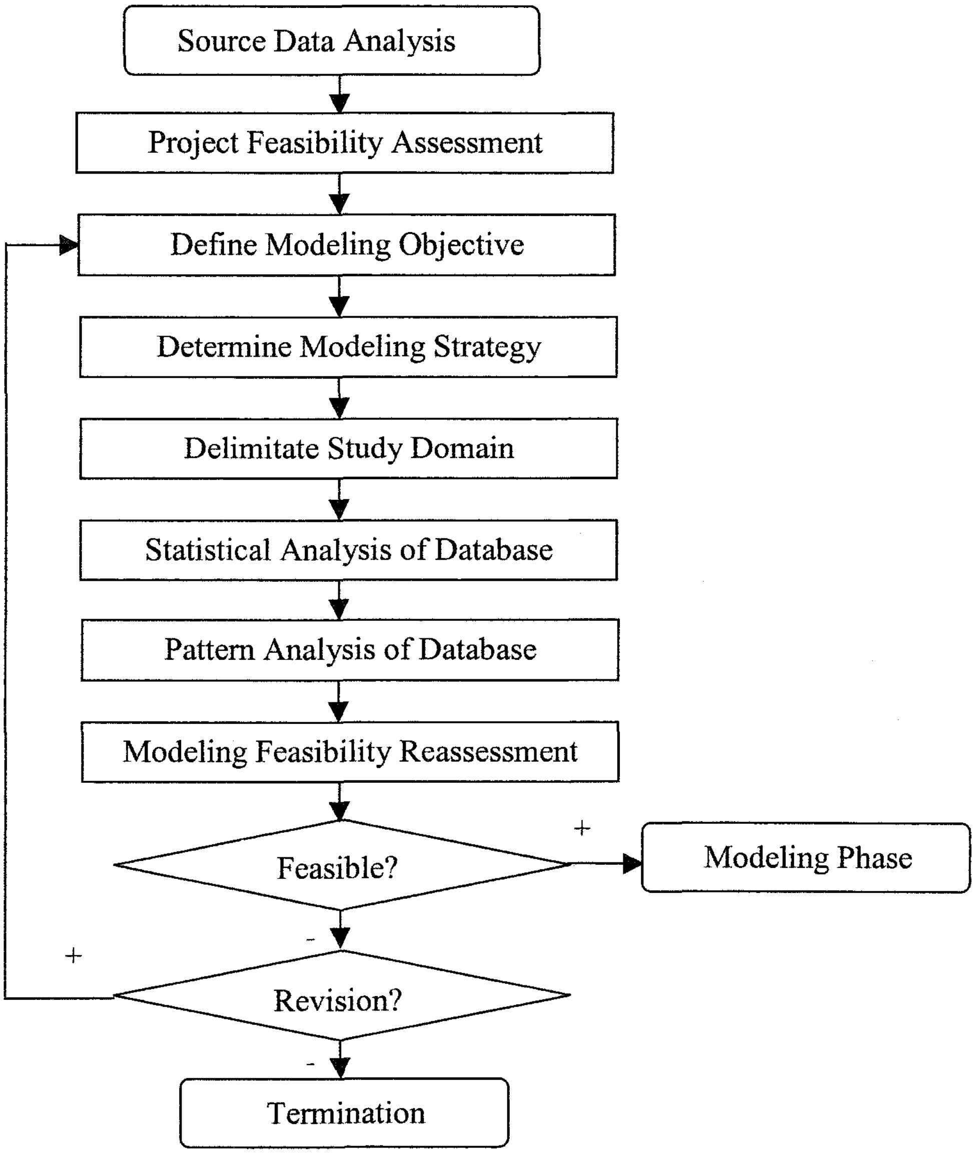
Source Data Analysis
Project Feasibility Assessment
Define Modeling Objective
Determine Modeling Strategy
Delimitate Study Domain
Statistical Analysis of Database
Pattern Analysis of Database
Modeling Feasibility Reassessment
Feasible?
+
Modeling Phase
-
+
Revision?
-
Termination

Figure 2-11 The ANN Model Design Protocol for the Source Data Analysis Phase

Model Development

Successful model development involves optimization of a multitude of modeling parameters. While the number of parameters and optimization routines vary according to the model developer, the following are generally considered to be the most important: selection of model inputs and outputs, selection of appropriate data patterns, organization of data patterns into multiple data sets, determination of the internal network architecture, and evaluation of candidate models. A complete description of these modeling components is presented by Baxter et al. (2001a).

The key issue in the model development stage is to ensure that the ANN models properly represent their study domains. In general, if an ANN model is over-trained, the model tends to memorize data points and its ability to predict outside of the memorized points is very limited. If an ANN model is under-trained, the model does not have enough knowledge of the study domain. Its prediction will be generally lack of accuracy over the complete domain.

Usually in the modeling process, the data are split into a training data set, a test data set, and a production data set. The test data set is used in the modeling process to cross-check the quality of the ANN model after each iteration of training. The production data set is a stand-alone data set that includes data never used in the training and testing process. The performance of the trained model on the production data set is a good indication of its capacity to predict the unseen events of the study domain. Thus, to build a good model, it is important to equally split the representative features of the study domain into the training set, the test set, and the production set. Only those ANN models that perform well on all three data sets will be selected into the candidate pool for further evaluation.

Model Evaluation, Fine-tuning and Implementation

Once a pool of candidate models is produced, statistical analyses as well as graphical analyses are performed to compare the models. Graphical analysis at this stage includes plots of actual versus model predicted output values. Such graphical plots can help highlight inconsistencies and problems with model predictions. Once the best candidate historical model has been selected, the next step is to implement the model in the actual application.

Since neural network models learn from the historical records, their knowledge depends highly on the accuracy and completeness of the historical data. However, it is very likely that the historical data might not cover the complete study domain. Thus, the best way to prove the successfulness of an ANN model is to actually run the model in its intended application environment.

In the water industry, once the ANN models for the water treatment process are built, the performance of ANN models can be evaluated online in real-time through integration with the plant SCADA system. The process models interface with the SCADA system to receive input data in real-time. These input data are then processed through the ANN model's run-time module, which subsequently returns a predicted output to the interface. By comparing the model-predicted value to the actual process output online in real-time over a set period of time, the success of model can be fully determined by its action in the actual application environment. The details of this type of implementation and testing can be found in a recent project report on the ANN on-line model by Baxter *et al* (2001b) to the American Water Work Association Research Foundation.

Conclusions

This chapter overviews the typical architectural components of an ANN model and a general ANN modeling protocol. The major architectural components of ANN model include artificial neurons, the links into and out of artificial neurons, and a learning algorithm. The discussion in this chapter focused on the scaling function and the activation function inside an artificial neuron. These two functions act together to determine the response of a neuron upon outside stimulus. Depending on which layer of network the artificial neuron is in, these two functions can be chosen differently to suit the primary purpose of the artificial neuron in that network layer. This chapter also highlighted the mechanism of the back-propagation learning algorithm, which is commonly used in many actual applications.

ANN modeling is a robust technique that can be used to model many non-linear scenarios. However, this flexibility in modeling also causes concerns on how to ensure the ANN models actually learn the right features of a study domain. To address this problem, the author first proposed an ANN modeling protocol as part of his master thesis in 1996. This modeling protocol established a general framework to build ANN models in a logical sequence and maximizes the chance to create reliable and accurate ANN models. Since then, this modeling protocol is accepted by many neural network researchers and evolved further with our improved knowledge of ANN architecture and learning algorithms. This chapter reviewed the major steps of that general ANN modeling protocol.

Some details of ANN architecture such as the threshold inside an artificial neuron, the interconnection weight and their adjustment, and the Kohonen learning algorithm, are not discussed in this chapter. These are the complex subjects intended for further investigation in Chapter 3 and 4. Chapter 3 will actually open the ANN black box and illustrate how the internal components of ANN model change with the learning process. Chapter 4 will discuss how pattern recognition technique can be used to improve the model building process.

References

Anderson, J. A., Rosenfeld, E. (1988), *Neurocomputing: foundations of research,* MIT Press in Cambridge, Mass., pp. 729.

Baxter, C. W., Zhang, Q., Stanley, S. J., Shariff, R., Tupas, R-R., and Stark, H. L., (2001a), Drinking Water Quality and Treatment: The Use of Artificial Networks, *Canadian Journal of Civil Engineering,* 28 (suppl. 1): 26-35.

Baxter, C. W., Tupas, R-R., Zhang, Q., Shariff, R., Stanley, S. J., Coffey, B. M., and Graff, K. (2001b), *Feasibility Study to Develop an Artificial Intelligence System for Optimization of Water Treatment Plant Operation*, published by American Water Works Association Research Foundation, Denver, Colorado.

Maren, A.J. (1990), Introduction to Neural Networks, in *Handbook of Neural Computing Applications,* edited by Maren, A.J., Harston, C.T., and Pap, R.M., published by Academic Press, Inc., San Diego, CA, pp. 1-11.

Rajasekaran, S., Febin, M. F., and Ramasamy, J. V. (1996), Artificial Fuzzy Neural Networks in Civil Engineering, *Journal of Computers and Structures*, 61(2): 291-302.

Zhang, Q. (1996), The Applications of Artificial Neural Networks in Water Industries, thesis for the Master of Science, in Department of Civil and Environmental Engineering, University of Alberta, Edmonton, Alberta, Canada.

Zhang, Q. and Stanley, S. J. (1997), Forecasting Raw-Water Quality Parameters for the North Saskatchewan River by Neural Network Modeling, *Water Research*, 31(9): 2340-2350.

Zhang, Q., Stanley, S. J., Greenaway, G., Zenari, L., and Ancel, S. (2001), Water Consumption Revenue Forecasting With Artificial Neural Network Technology, *Proceedings of 2001 Information Management and Technology Conference*, published by AWWA, Denver, Colorado.

Chapter III Internal Workings of ANN Models

Introduction

Although there are many successful real world applications of artificial neural network, it has a significant drawback. Even experts in the field of ANN research consider ANN model as a black box. This perception develops mainly due to the fact that it is very difficult to present most ANN models in a precise mathematical format. Thus, it is difficult to comprehend how the ANN models work.

Our knowledge of the inner workings of ANN models progresses with the evolution of ANN modeling technique. The neural networks of early 1960's are typically in a form of single layer perceptrons. A perceptron is an artificial neuron with a linear activation function. Those neural networks built with perceptrons can be easily converted into precise mathematical formulas, and their capabilities are clearly understood. In general, a neural network built with perceptrons is limited to solve linear separable problems, and has no capacity to model non-linear problems. A good example to illustrate the capacity of perceptron neural network is the famous XOR logic problem presented by Minsky and Papert (1988) in their study of perceptrons. The XOR logic as shown in Table 1 is a simple non-linear problem, which requires a non-linear boundary to solve the problem.

Since the 1980s, several complex learning algorithms such as back propagation were discovered to allow neural networks evolve into a multi-layer structure with various non-linear activation functions. These multi-layer neural networks can map multiple inputs to multiple outputs, and have the capacity to model non-linear problems. However, our understanding of the behaviors of these multi-layer neural networks is reduced to qualitative concepts because of inability to decipher the non-linear activities within the network of neurons.

Table 1 The XOR Logic

X	Y	**XOR Logic**
1	1	**0**
1	0	**1**
0	1	**1**
0	0	**0**

The lack of understanding of multi-layer neural networks results in many modeling difficulties in actual applications. For example, there is no clear definition on how much training is over-training, and there is no clear method to determine the confidence level of those interpolations and extrapolations made by neural network models. There are also issues such as how to define the noise in the source data. Many novice modelers simply drop the individual cases that can not be reasonably well modeled without specifying any criteria for doing this. By simply declaring the difficult cases as noise and dropping them from the modeling database, they could produce a neural network model that appears to fit the data well. In fact, there is a strong chance that they have misrepresented the source data and developed a model that does not truly fit the data set. These misunderstandings of ANN have a negative impact on the reputation of ANN modeling technique.

To improve our knowledge of ANN models, in this chapter, a systematical approach is presented to interpret the inner workings of ANN models. First, a graphical interpretation technique will be illustrated to open the ANN model and show the detailed function of each inner component. Second, with this graphical interpretation technique, many behaviors of the ANN models will be analyzed through case studies. The data presented in these studies are the original material from author's research. The studies include how the training process of ANN model changes the internal components of an ANN model, how noise impacts the training process, and the optimization issues such as robustness, fault tolerant, sensitivity, and optimal training of an ANN model.

Understanding Back Propagation Neural Network as a Function

To establish the foundation of the graphical interpretation technique, it is necessary to prove that a back propagation neural network, which is the most commonly used ANN in actual applications, is a function. If we can successfully prove that, many properties of function can be applied to explain the behaviors of the back propagation neural network. As the focus of this discussion is the back propagation neural network, all references to neural networks in this chapter will be to that type of neural networks.

To prove a neural network is a function, it is necessary to examine the very fundamental element of neural network architecture, the artificial neuron. If a typical artificial neuron is proven to be a function, then a neural network can be deducted as a function. This is because that the interconnection of the artificial neurons to form a neural network is equivalent to performing algebra and composition on basic functions, and eventually producing a complex function.

As discussed in Chapter 2, a typical artificial neuron consists of three components: input links, a processing unit, and an output link. The processing unit usually consists of two functions and a threshold. The threshold is a negative resident weight of the neuron such as the W_t as shown in Figure 3-1. The two functions are a summation function and an activation function. The summation function sums up the inputs and the threshold, and then the sum is fed into the activation function to produce an output stimulus. These processes can be expressed in Equation 3-1 and 3-2.

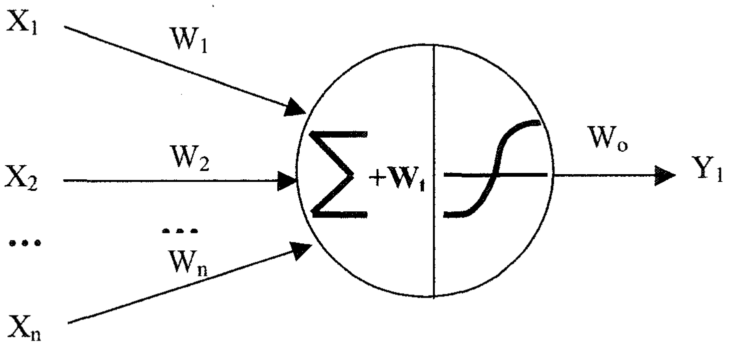

Figure 3-1 A Typical Artificial Neuron

Summation function: $S(X_1, X_2, \ldots, X_n) = X_1 \bullet W_1 + X_2 \bullet W_2 + \ldots + X_n \bullet W_n + W_t$ (3-1)

where $X_1, X_2, \ldots, X_n$ are n outside stimuli;

$W_1, W_2, \ldots, W_n$ are weights on the input connections;

W_t is the threshold.

Activation function: output $Y_1 = W_o * f(S(X_1, X_2, \ldots, X_n))$ (3-2) where W_o is the weight on the output connection.

Obviously, by definition, the summation function and the other common activation functions such as the logistic function and the tanh1.5 function are functions. Because the artificial neuron is a composition of the summation function and the activation functions, it can be concluded that the artificial neuron is also a function as a result of composition of known functions.

Theoretically, one can choose a non-function such as arcsin(x) as the activation "function". Arcsin(x) maps a value (x = sin(y)) to all possible angles (ys) that could produce the same value x. For example, if x = 0.5, arcsin(0.5) = $\{2k\pi + \pi/3\} \cup \{2k\pi + 5\pi/3\}$ where k is an integer. Thus, for x=0.5, there are many solutions. Since the solution is not unique, arcsin(x) is not a function. If this is the case, the artificial neuron will not be a function. Nevertheless, under normal circumstances, the activation function inside the neuron is indeed a function. Thus, our interpretation that the artificial neurons are functions remains true.

Typically, the artificial neurons are organized into a network in a feed-forward fashion. As shown in Figure 3-2, the outside stimuli X_n are scaled by the input layer neurons. The scaled stimuli X_n' are then fed into hidden layer neurons to produce intermediate signals Y_3 and Y_4. These intermediate signals are used by the output neuron as its inputs to produce a final signal Y_5. Similarly, any output of a neural network is generated in this feed-forward fashion. If each neuron in this feed-forward network is a function, then the overall network can be concluded to be a function. This is because the output function Y is simply a composition of all the previous neurons, and thus is a composition of functions. Therefore Y is a function and the whole network is a function. It can be therefore generally concluded that a back-propagation network is a function no matter how complex its precise mathematical formula may be.

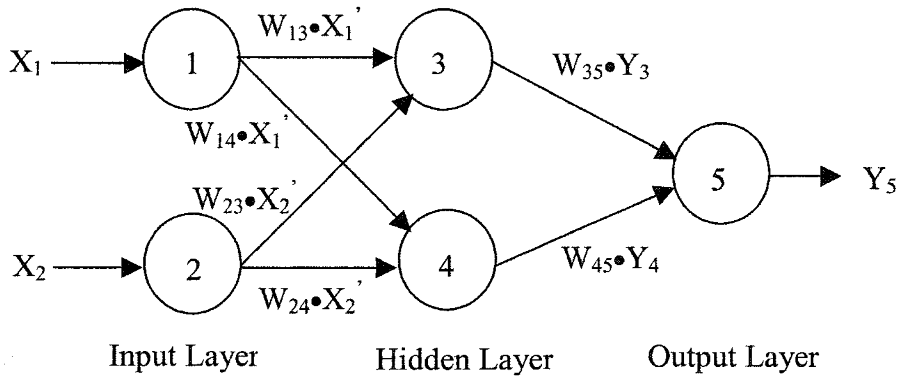

Figure 3-2 A Simply Feed-forward Neural Network

The Graphical Interpretation Technique: Mapping Inputs to Outputs

Since back-propagation neural network is a function, theoretically it is possible to express a neural network into a mathematical formula. However, the formula would be extremely long and non-intuitive to interpret. The alternative is to use graphical analysis to interpret how the network converts inputs into a unique output. This graphical technique is called mapping inputs to output(s). Since an artificial neuron is the basic element of a neural network, this section will closely examine an artificial neuron and analyze how the connection weights, the activation function, and the thresholds impact on its input-output mapping process.

By the definition of a function, the mapping between the elements of the input domain and those of the output domain can only be a many-to-one or a one-to-one relationship. In other words, it is allowed to map multiple independent data points in the input domain to a single data point in the output domain. This is the many-to-one relationship. It is also possible that there is one and only one data point in the input domain mapping to one and only data point in the output domain, and vice versa. This type of relationship is more stringent and it is called the one-to-one relationship.

Typically, the activation functions are designed to have its activation zone acting in a one to one relationship. One such example is the symmetric logistic function (Equation 3-3) as shown in Figure 3-3.

Symmetric logistic function: $f(\mathrm{x}) = 2/(1+\mathrm{e}^{-\mathrm{x}}) - 1$ (3-3)

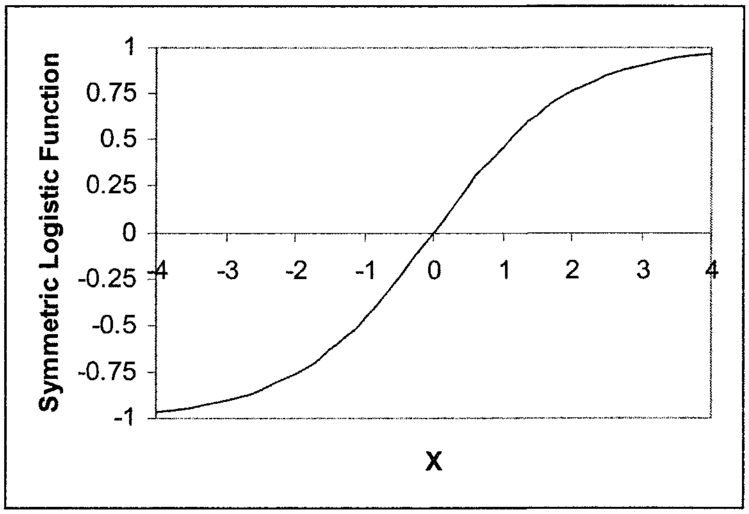

Figure 3-3 The Symmetric Logistic Function

The most active zone is between $x = -3$ and 3, where change of slope is more noticeable. The two end zones ($|x| \geq 3$) has relatively flat response because the function rapidly saturates at $y = \pm 1$.

Previously an artificial neuron was proven to be a function, thus the rules of transformation of a function can be applied to an artificial neuron. The common transformations of function in an artificial neuron are horizontal shifts, vertical and horizontal stretching, and vertical and horizontal reflecting. These transformations generate a family of function curves with various shapes from a single activation function. This variety in shape allows more flexibility in combining functions into a targeted shape. The following discussion illustrates that the basic tasks of the connection weights and thresholds are to transform the signal functions connecting into and transmitting out of an artificial neuron.

As shown in Figure 3-1, a typical neuron has input links, a threshold, and an output link. Each input or output link is presented as a connection weight. It is these connection weights and the threshold that transform the shape of the function. For simplicity, it is assumed that the neuron starts with only one input. Thus there are three adjustable components in this neuron: the input connection weight W_1, the threshold W_t, and the output connection weight W_o. By combining and adjusting Equation 3-1, 3-2 and 3-3, the output function for this neuron becomes:

$$f(x) = W_o * 2 / (1+e^{(-W_1 \bullet X + W_t)})-1 \qquad (3\text{-}4)$$

According to the rules of transformation, in Equation 3-4, W_l serves to compress or stretch the graph of activation function *f*(x) horizontally by a factor of W_l depending on whether W_l is greater or less than 1. W_t serves to shift the graph of *f*(x) to the left or right depending on whether W_t is positive or negative. W_o serves to stretch or compress the graph vertically by a factor of W_o depending on whether W_o is greater or less than 1. If W_o is negative, the graph is reflected about the x-axis.

It is very important to understand the properties of connection weights and threshold to fully decipher the behaviors of a neural network. The impacts of connection weights and threshold can be illustrated in the following example. When a neural network is in the training phase, W_l, W_t and W_o are adjusted. The adjustments of W_l and W_t affect the inputs to the current neuron. The adjustments of W_o affect the output of the current neuron, and subsequently the input to the next neuron.

Adjusting W_l and W_t shifts the most sensitive region of the activation function to a special input region. For example, if after a neural network training, W_l is set to 10 and W_t is set to -5, the activation function shifts to the new curve 1 as shown in Figure 3-4. The original graph is compressed and shifted to the left. The slope of the curve becomes much steeper. This effectively reduces the responsive region of input range from the original –3 to 3 down to –0.8 to –0.2. The steeper slope at the input range of –0.8 to –0.2 also makes the output response more sensitive. The output saturation zone, where change in inputs does not incur change in output, are now at $x > -0.2$ or $x < -0.8$. On the other hand, if W_l is set to 0.25 and W_t is set to -0.25, the graph of the activation function takes the shape of new curve 2 in Figure 3-4. The curve is now stretched and flattened, almost like a linear function. The non-saturated input range is increased 4 times in size and shifted to the right.

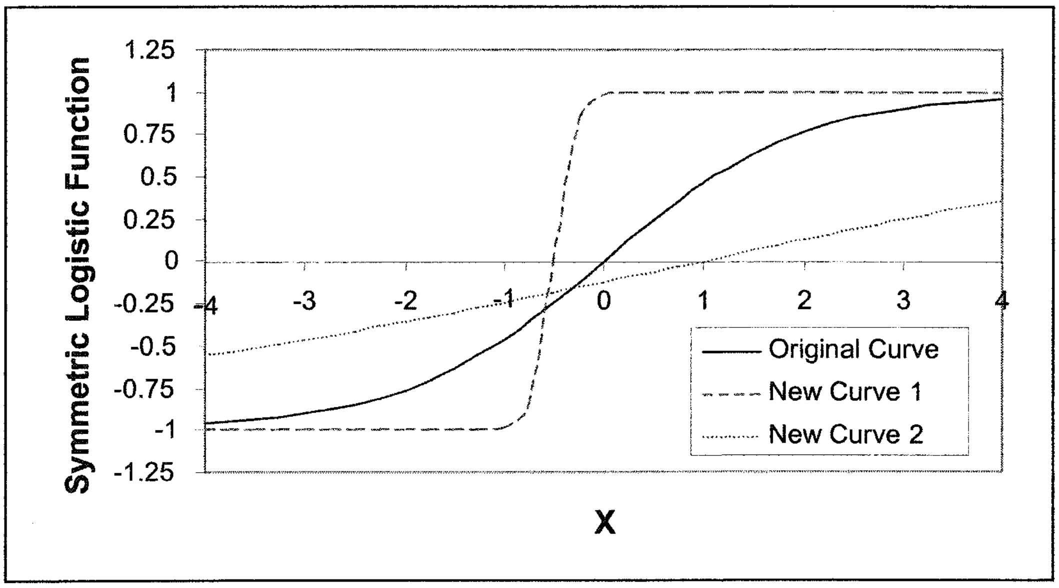

Figure 3-4 Impacts of Input Connection Weight and Threshold on the Activation Function

Adjusting the output connection weight, W_o, directly impacts the slope of activation function curve on the vertical direction. The effect of W_o is similar to that of W_l. They both change the slope of a function curve in a similar style, but they affect the function curve in a different direction. W_l impacts on the x-axis, and W_o impacts on the y-axis. The other connection between W_l and W_o is that if a neural network has a multi-layer connection, the output connection weight of the neuron in the previous layer is simply the input connection weight to the neuron in the next layer. One example of the impacts of

W_o is shown in Figure 3-5. The New Curves 3 is generated by setting W_o of the original curve to 3. Thus, the maximum vertical range is now changed from [-1, 1] to [-3, 3]. New Curve 4 is generated by setting W_o of the original curve to –0.5. Similarly, the maximum vertical range is now changed from [-1, 1] to [-0.5, 0.5] and the curve also flips around the x-axis.

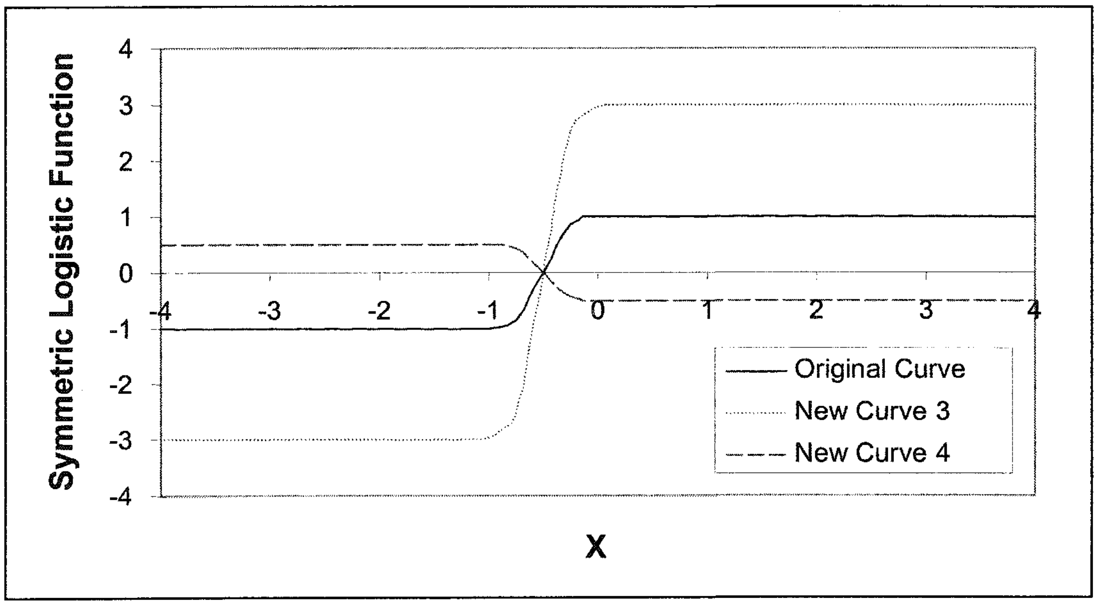

Figure 3-5 Impacts of the Output Connection Weight on the Activation Function Graph

The last key component of the input and output mapping is the activation function itself. Among the commonly used activation function, there are two major types: a monotonic function or a non-monotonic function. The symmetric logistic function is a monotonic function (Figure 3-3), which maps as a one-to-one relationship. The Gaussian function on the other hand is a non-monotonic function (Figure 3-6), which maps as a many-to-one relationship. Although Gaussian function is not commonly used, its non-monotonic shape actually makes it easier to combine the curves out of a function family to form a complex target function than the monotonic function.

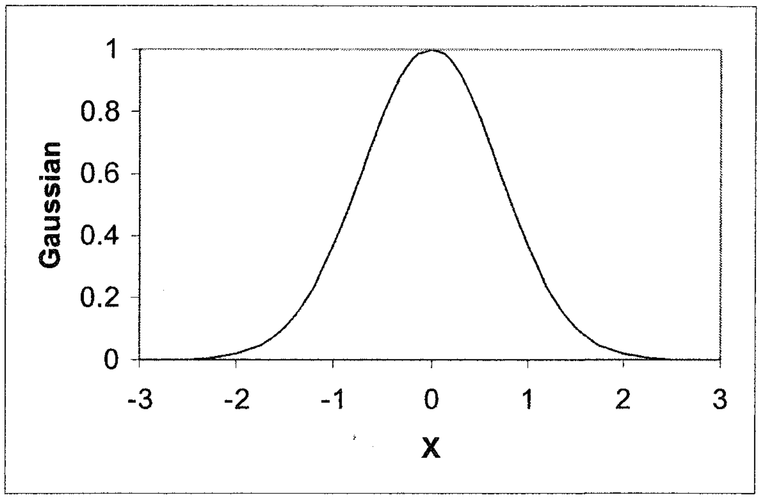

Figure 3-6 The Gaussian Activation Function

It is also important to note that within the monotonic function family, there is no evidence to support the claim that any activation function with steeper slope maps better than the other ones. There is a common misconception that steeper sloped activation functions such as Tanh1.5, map input to output better than the smoother sloped functions such as the logistic function, when designing a neural network for a very sensitive input to output mapping. In terms of mapping input to output, theoretically, this is not true simply because we just demonstrated that it is possible to change the shape of an activation function by adjusting either input connection weight, output connection weight, or threshold.

There is also a misconception that selecting a different activation function from the same monotonic function group will result in different performance in interpolation or extrapolation of the neural network model. For the same reason stated above, it is not true.

So far all the discussion has been based on the case that there is only one input to the neuron. However, all the principles and interpretations of connection weights and thresholds illustrated in the previous scenarios can be extended to the multi-input neurons. This is because all additional inputs must go through the summation function before they go on to the activation function. The summation function linearly combines all inputs (functions) into a final input (function), thus reduces the multi-input scenario into one input scenario again.

However, the linear combination of inputs does add complexity to artificial neurons. Additional inputs represent expansion of the domain to draw the input data from. For example, if there is one input, the source data are drawn from a linear source (one input → one output). If there are two inputs, the source data are drawn from a plane source. If there are n inputs, the source data are drawn from the n-dimensional space.

The major concern for the multi-input neuron is the quality of inputs, specifically, the correlation among the inputs. For example, assume there are two inputs for the neuron: X_1 and X_2. If X_1 is independent to X_2, the summation function becomes $S(X_1, X_2) =$

$W_1 \bullet X_1 + W_2 \bullet X_2 + W_t$. Effectively, this states that the source data are drawn from a two-dimensional space.

If X_1 is somehow related to X_2, the summation function could be dramatically changed and result in completely different dynamics in neuron output. Assume X2 can be expressed as a function of X_1, such as $g(X_1)$, then the summation function becomes $S(X_1) = W_1 \bullet X_1 + W_2 \bullet g(X_1) + W_t$. There are two points to be considered here. First, obviously the source of data goes from two dimensions to one dimension. Losing a dimension means losing input variety and thus likely affects the model accuracy. However, $g(X_1)$ could introduce a non-linear component to $S(X_1)$, and thus increase the information capacity of the X_1 dimension. For example, assume X_2: $g(X_1) = X_1^2$. As the result, the output function has changed from a monotonic function into a non-monotonic function shown as New Curve 5 in Figure 3-7. Thus, the non-linear component g(X1) could help the model in prediction.

Overall, the strategy in selecting the inputs is that it is preferable for all inputs to be independent. The selection of the non-independent parameters should be limited and should only serve as a necessary or required data redundancy component.

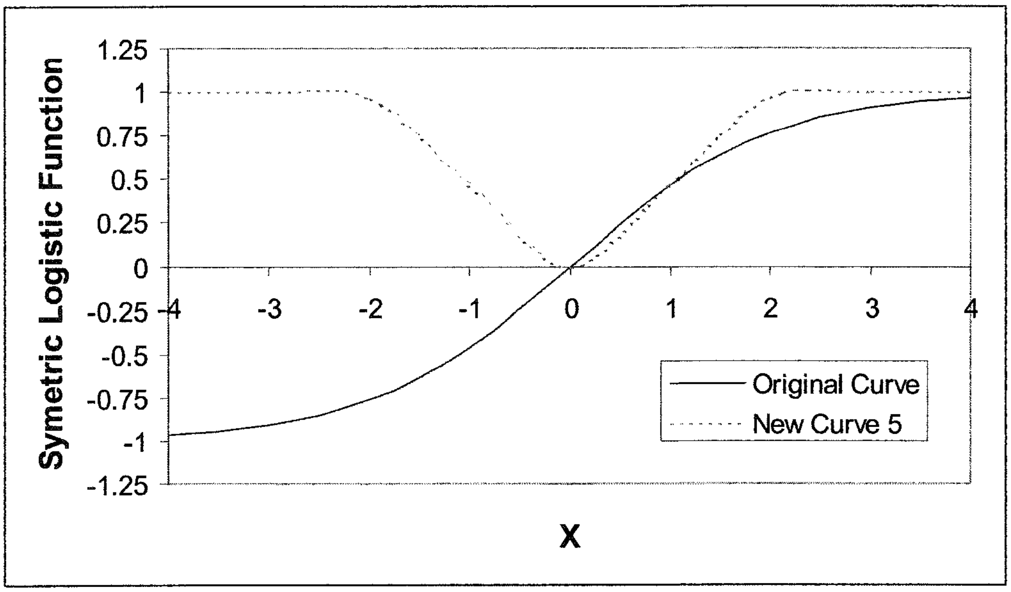

Figure 3-7 The Impact of Nonlinear Input Component

Inside a Complex Neural Network: Case Studies of Mapping Noise

Once we understand the internal activities of an artificial neuron, the next logical step is to decipher the inner activities of a neural network. It is important to point out that the basic functionality of each neuron of a neural network remains almost the same, the only difference is that information passed from the neurons in the previous layer to the neurons in the next layer are much more complex now. This processed information is refined and constricted, but still can be presented as function. The combination and composition of these functions (processed information) directly impact how neural networks recognize features, interpolate, and extrapolate the domain. Again, the best approach to demonstrate how the inner components of a neural network adapt to learn the features is graphical interpretation. In this section, graphical analysis is used in several case studies to illustrate how noise records impact the inner components of neural network.

What is Noise

Noise is always a concern in real world application of ANN modeling. Since ANN learns from historical records, if a faulty record is not screened out and remains in the training process, ANN would learn that faulty record to some degree. If that faulty record is dominant in a region, then this noise would be fully learnt as a false feature of the domain. If the faulty record has a lot of correct neighboring data points, the impact of the faulty record would be smoothed to a less degree on ANN learning. This section will further discuss what is noise, the sources of noise, and how noise affects the behaviors of neural network models.

What is noise? In terms of modeling, there are two major types of noise. One is the pure error generated in the process to collect that piece of information. The other type of noise is associated with the difficulty in modeling incurred by missing information. Pure error could be instrument error or human error. It should be pointed out that pure error is different from random error because pure error is the data point that is statistically different from the distribution patterns of a data population. One such example is the outlier. It is usually not difficult to detect the pure error because it can be determined by the traditional statistical analysis.

However, the noise due to missing information is much more difficult to detect and determine, because most of time, this type of noise is not statistically significant and requires expert knowledge to identify. For example, assume there is a true function T = *f*(X, Y, Z). Thus each piece of complete data will be (X, Y, Z, T). If the data collection process is incomplete, then likely each piece of data only has (X, Y, T). If these incomplete data sets are provided for the neural network to learn and map (X, Y) to T, chances are there would be many data pieces, which have similar Xs and Ys but very different output Ts. This creates a problem in the input to output mapping. In this scenario, there may be more than one data point in the output domain that will match one data point in the input domain. A function simply does not have the capacity to complete this type of mapping by its definition.

It is possible to obtain a neural network model even with this type of scenario. But the neural network model simply can't predict with satisfactory results. This can be illustrated through a simple case. Assume there are two noise patterns P_1 and P_2. P_1 is (X_1, Y_1, T_1) and P_2 is (X_2, Y_2, T_2). (X_1, Y_1) is identical to (X_2, Y_2) but T_1 is very different from T_2 in value. If back-propagation is the learning algorithm, the neural network model would be trained to minimize the overall error in predicting these two cases. After the training, the neural network model will predict T_1 as T_1' and T_2 as T_2', in which T_1' is equal to T_2' so that the square root of $(T_1\text{-}T_1')^2 + (T_2\text{-}T_2')^2$ is minimized. Obviously the value of T_1' is the average of T_1 and T_2, and deviates away from the correct predictions for both T_1 and T_2. This type of prediction error can not be reduced or eliminated by any further training. Too much of this type of noise in the training data set will substantially reduce the accuracy of model performance. Thus it is a major concern of ANN modeling.

However, if (X_1, Y_1) is not completely identical to (X_2, Y_2), then the neural network would tend to memorize them. Although neural network has the capacity to memorize them, the interpolation dynamic inside the neural network will be dramatically changed due to memorization. This can be illustrated in the following example.

Case Study 1: A Simple Noisy Case

In this example, the true function of some instrument readings is y = X. Assume this true function is not known at the moment, the current task is to model the instrument readings with neural network. The initial sampling takes reading over the input range from 0 to 3. To complicate this scenario study, it is also assumed that at X = 1.6, an error happened with the instrument which results in a reading of 4 as shown in Figure 3-8. Since a reading of 4 is not statistically different from the other readings, the reading is accepted. The initial readings are listed in Table 3-2. The objective of this case study is to open the neural network black box and to understand how the connection weight, threshold, and activation function affect the inner dynamic of a neural network model.

To analyze the properties of activation function, two types of activation function are used to compare their capacities to model the nonlinear data in Table 3-2. First, the Gaussian activation function, a non-monotonic function, is examined. Then, the symmetric logistic function, a monotonic activation function, is used to study the same data set. During the training phase, the whole data set is used as the training set.

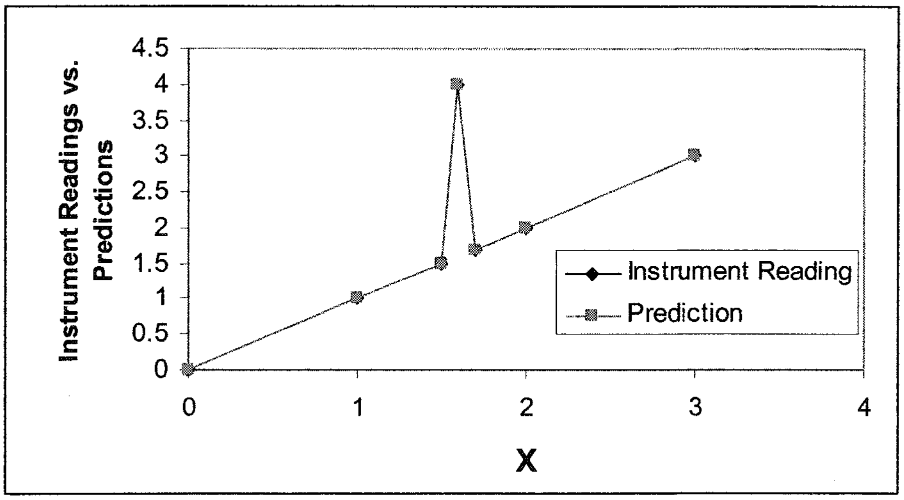

Figure 3-8 Case 1 Prediction Results with Gaussian Activation Function

Table 3-2 The Initial Readings of an Instrument

x	**Reading**
0.0	0.0
1.0	1.0
1.5	1.5
1.6	4.0
1.7	1.7
2.0	2.0
3.0	3.0

The Gaussian activation function is first used to train a 1-3-1 feed-forward neural network, which has one input neuron, 3 hidden neuron, and one output neuron. The neuron network trained is able to predict the training data with 100% accuracy. As shown in Figure 3-8, the original data and the network prediction completely overlap.

This example highlights why it is so important to consider noise data with neural network modeling. As shown in Figure 3-8, because of neural networks' capabilities in modeling data, it is actually able to model the error adequately. Most other modeling techniques such as regression analysis would not be able to match the noise data and it is likely that they would indicate that data point is an outlier. Modeling with neural network in this case gives no indication based on modeling results that there may be a problem. As a result, it is very important to assess noise data when modeling with neural networks. The issue with outliers will be discussed in greater detail in the next chapter when a pattern classifier is presented to highlight outliers in complex data sets.

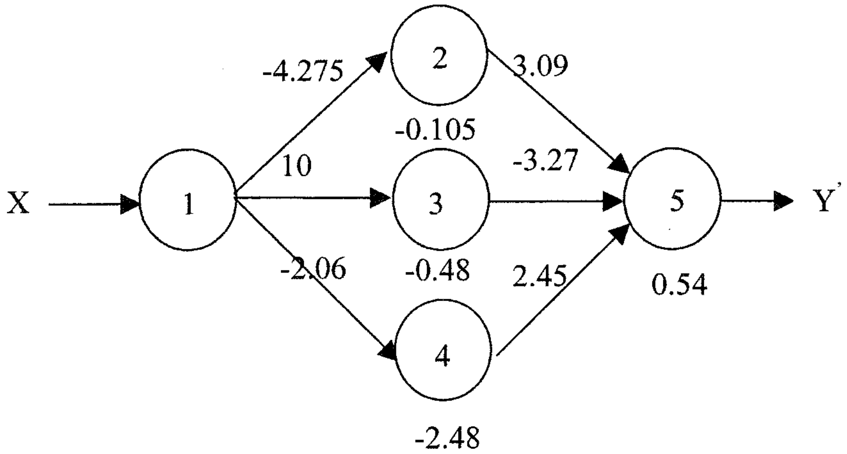

Figure 3-9 Case 1Neural Network Structure with Gaussian Activation Function

It is the interest of this research to study why neural networks have such a powerful modeling capability. The answer lies inside the ANN architecture. The training result inside the neural network is shown in Figure 3-9. The connection weights are shown on the link and the thresholds are shown under each neuron.

The most effective approach to analyze the neural network internal dynamics should start at the output neuron. In this case, there are three input links into the output neuron (Figure 3-9). Named by the related neurons, these links are Link 2-5 (neuron 2 to neuron 5), Link 3-5, and Link 4-5. These links convey the processed information from the hidden neuron to the output neuron. To demonstrate how the information flows through these links, the change of information in each link over input X is shown in Figure 3-10. This type of chart is called the input chart. The value of the link is obtained by multiplying the output of the previous neuron by the connection weight. The summation of the links over input X is also shown, which takes the threshold into account too. The x axis of Figure 3-10 is the summed inputs to the activation function of Neuron 5.

As shown in Figure 3-8, the sample data can be interpreted as consisting of two components: a basic function y = X to represent the normal readings from the instrument, and a noise pulse function at the neighborhood around X = 1.6. Thus, there are two major questions to be answered from Figure 3-10. First, how the noisy peak in Figure 3-8 is generated inside the neural network. Second, how the function y = X is generated inside the neural network.

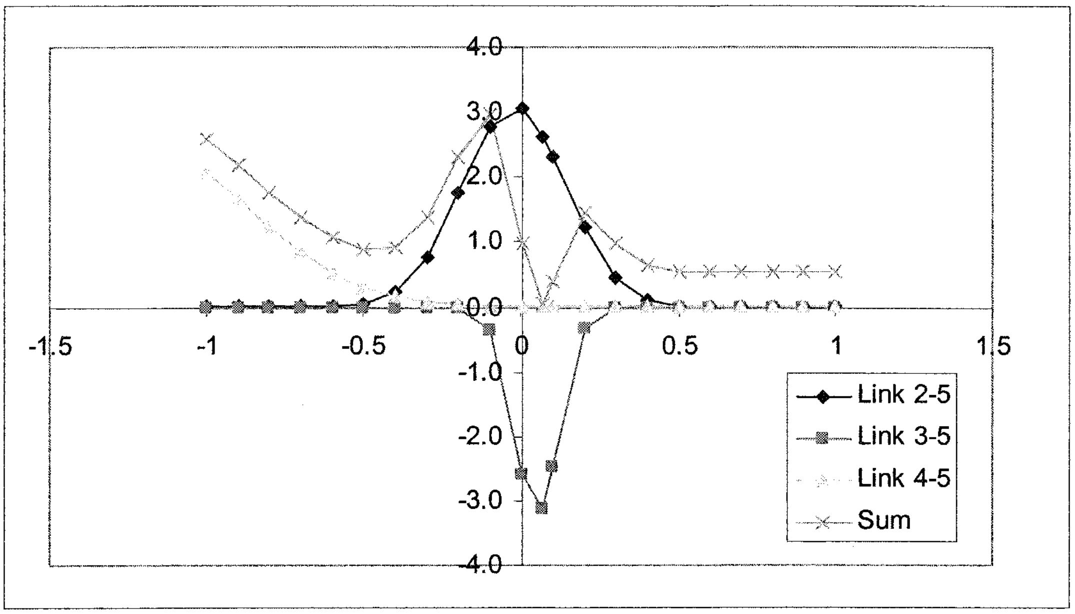

Figure 3-10 Case 1 Input Chart for the Output Neuron

Since the summation function is the final input to the Gaussian activation function inside the output neuron to produce the final output, let's first examine its curve in Figure 3-10. The basic shape of the summation curve (the Sum curve) in Figure 3-10 is a Gaussian curve. However, there are three important modifications to the basic curve. First, at the neighborhood of x = 0.07 (the scaled value of X = 1.6), the trend of the Gaussian curve is disrupted and dipped to a very small positive value. Second, in the region of x < -0.5, the Gaussian curve is changed to a positive raising trend. Third, the whole Gaussian curve is shifted vertically up about 0.5 from 0.

All these features of the summation curve are the results of summing the linear components of the summation function. As shown in Figure 3-10, Link 2-5 generates the basic Gaussian shape. Thus, the overall shape of the summation curve is a Gaussian curve. Link 3-5 is a modified inverted Gaussian curve. The inversion is caused by the negative connection weight (-3.27) on Link 3-5. The central peak of the inverted Gaussian curve is also compressed to a limited region around the neighborhood of x = 0.07 (at X = 1.6). The rest of the region is flat around zero. This is caused by the connection weight on the Link 1-3, which causes the output of neuron 3 to be compressed by a factor of 10. The combination of Link 2-5 and Link 3-5 generates the inverted peak at the neighborhood at X = 1.6. Apparently this is the region where the pulse noise is in the source data. It is an inverted peak because the Gaussian function produces high values when its input is close to 0 (Figure 3-6).

Link 4-5 is responsible for generating the raised left tail of the summation curve and the threshold of neuron 5, which is 0.54 is responsible for raising the overall curve upward in the y-axis for about 0.5. The combination of these two components in effect generates the y = X function. Link 4-5 maintains the lower part of y = X at X < 1.5. The threshold sets up the upper part of y = X at X > 1.7.

From the above discussion, it can be concluded that each component of the artificial neuron actually controls a separate feature of the source data. Neuron 2 produces the basic Gaussian function to control the overall shape of response. Neuron 3 produces the noise peak of the source data. In other words, the location of the noise peak is memorized in neuron 3. Neuron 4 and the threshold of neuron 5 produces the y = X function. The connection weights and the other thresholds properly change the transition information to its proper shapes. Overall, this is the minimum requirement to reproduce the source data. It is impossible to reproduce the source data with any fewer neurons.

Case Study 2: A More Complex Noisy Case

To prove that our interpretation of neural network inner workings is not a random success but a systematic approach, a second scenario is studied. Furthermore, in this case study, the research will focus on the detailed behaviors of different activation functions, their capacity to reproduce the non-linear model, and the behaviors of additional hidden layers.

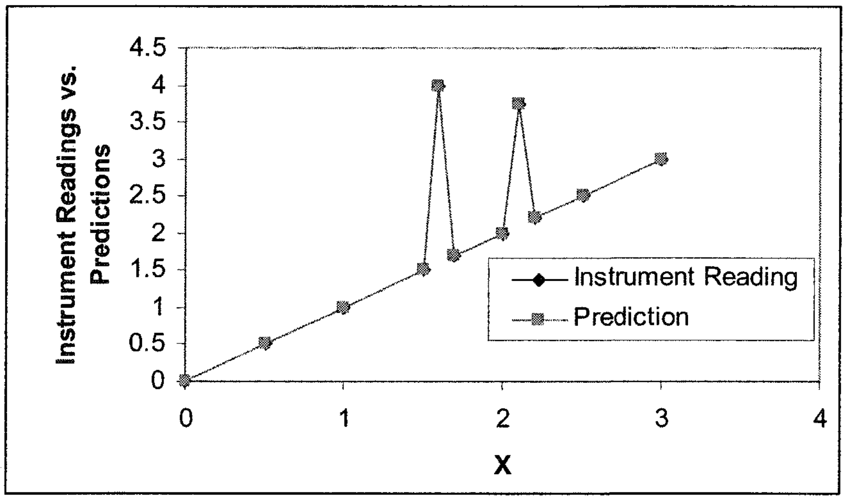

Figure 3-11 Case 2 Prediction Results with Gaussian Function

A second noise peak is added at X = 2.1. Several extra sampling data are added, and the revised source data is presented in Table 3-2. The Gaussian activation function is again used to reproduce the source data with 100% accuracy (Figure 3-11). The network used has a 1-4-1 structure, which contains one input neuron, four hidden neurons, and one output neuron. The connection weights and thresholds are presented in Figure 3-12. The plot of the links to the output neuron is presented in Figure 3-13.

Table 3-2 The Modified Readings of an Instrument

X	Reading
0.0	0.0
0.5	0.5
1.0	1.0
1.5	1.5
1.6	4.0
1.7	1.7
2.0	2.0
2.1	3.75
2.2	2.2
2.5	2.5
3.0	3.0

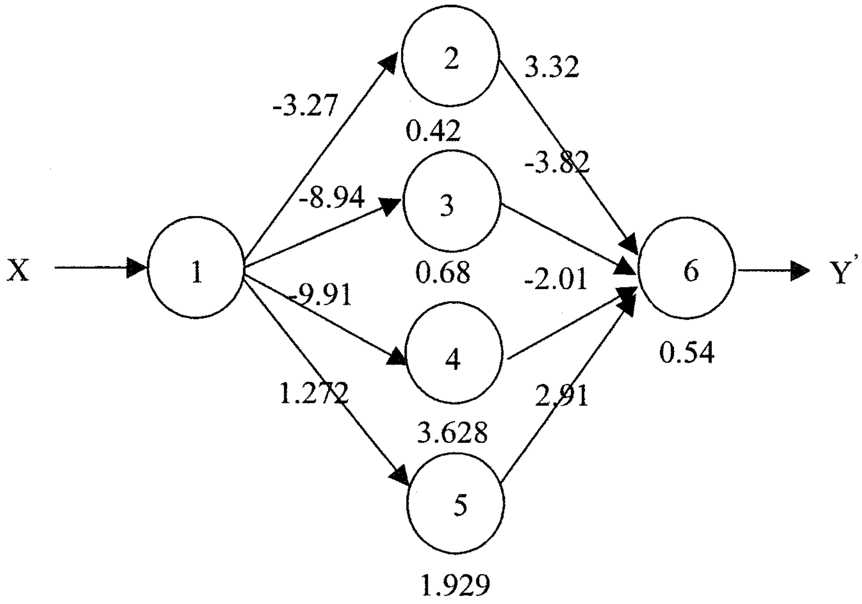

Figure 3-12 Case 2 Neural Network Structure with Gaussian Activation Function

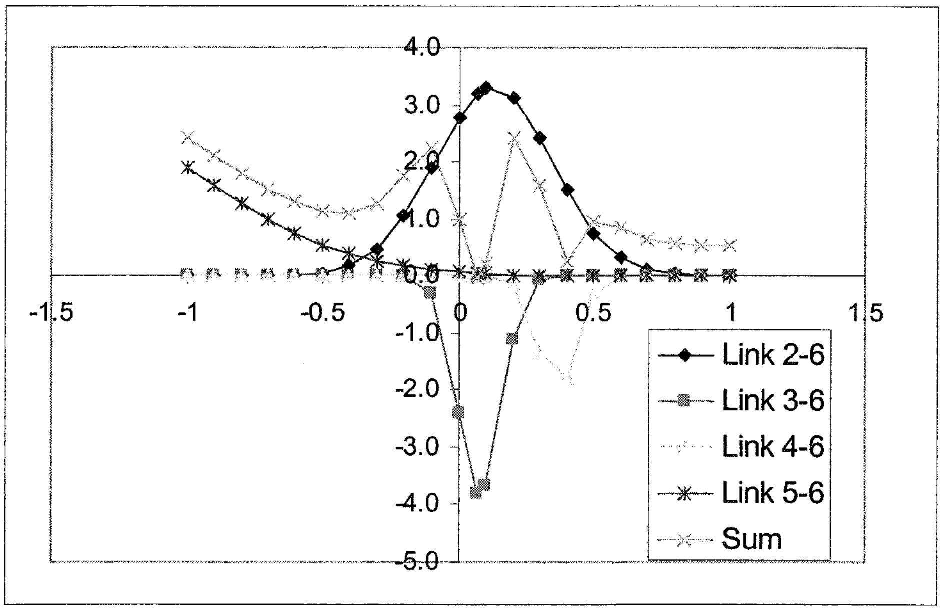

Figure 3-13 Case 2 Input Chart for the Output Neuron

The principles used to interpret Figure 3-10 can also be applied to interpret Figure 3-13. The features controlled by the neurons are the same as in the previous case study, except that one additional neuron, Neuron 4, is required to memorized the location of the second noise peak. Thus, it is possible to conclude that for every random error, there is a need for an individual neuron to memorize the location of the noise. As the number of the noisy data points increases, the number of neurons needed to memorize the noise also increases. This is a huge drain of resources for neural networks. This statement is true for the random noise. However, further research is needed to study systematic noise, which could be expressed as a function. How neural networks represent such a systematic noise function should be an interesting subject for further study.

It is also very important to note that the behavior of each neuron and the connection weight are predictable. Thus, it is possible to manually adjust the connection weights and thresholds to modify the network behavior to fit some specific requirement. This would provide greater flexibility for the advanced modeling using neural networks.

To compare the Gaussian function with the symmetric logistic function in terms of modeling capacity and flexibility, a second group of ANN models was trained with the same source data using the symmetric logistic function. The modeling results are exactly the same as those shown in Figure 3-8. The model details are presented in Figure 3-14.

As shown in Figure 3-14, the neural network has a 1-3-2-1 architecture. It was found that it was impossible to completely reproduce the initial data set with one hidden layer. The minimum solution has to be a two-hidden-layer architecture. This can be explained by examining the output neuron. As shown in Figure 3-14, there are two links connected to the output neuron. The variance of these two links over input X is shown in Figure 3-15. Notice that there are three main features with the sum curve in Figure 3-15. First, there is a peak at x = 0.07. This is the exact location where the noise peak is located. Second, there is a steep rising trend at the left end of the sum curve. Third, there is a moderate rising trend at the right end of the sum curve. The second and third feature of the sum curve is actually responsible for producing the y = x function.

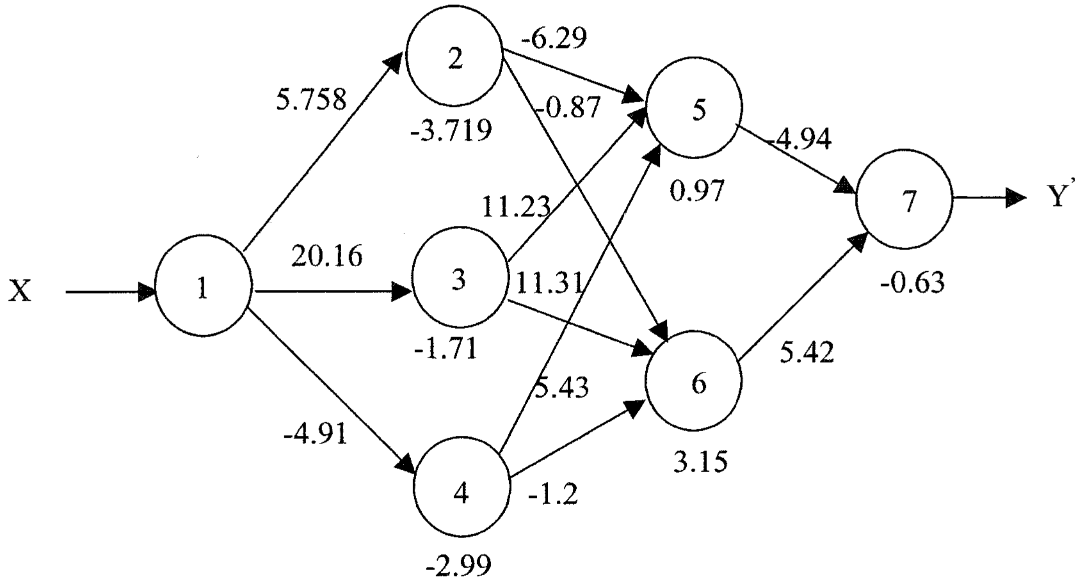

Figure 3-14 The Neural Network Structure with Symmetric Logistic Function for the Case 1 Data Set

Notice that any peak or valley on the graph presents a sudden change of the trend. Since the symmetric logistic function is a monotonic function, any modification of a single symmetric logistic function will not change it from monotonic to non-monotonic. Thus, at least two curves are needed to produce a peak as shown in Figure 3-15. In order for the symmetric logistic function to produce a peak, there are three conditions that must be met. First, the symmetric logistic function has to have a steep grade change in the area of the peak. In this case, this requirement is achieved by increasing the connection weight into Neuron 5 and 6. As shown in Figure 3-15, the connection weights from Neuron 3 to Neuron 5 and 6 are both high in value (11.23 and 11.31). As the result, the output curves from Neurons 5 and 6 are compressed and produce the steep grade as required.

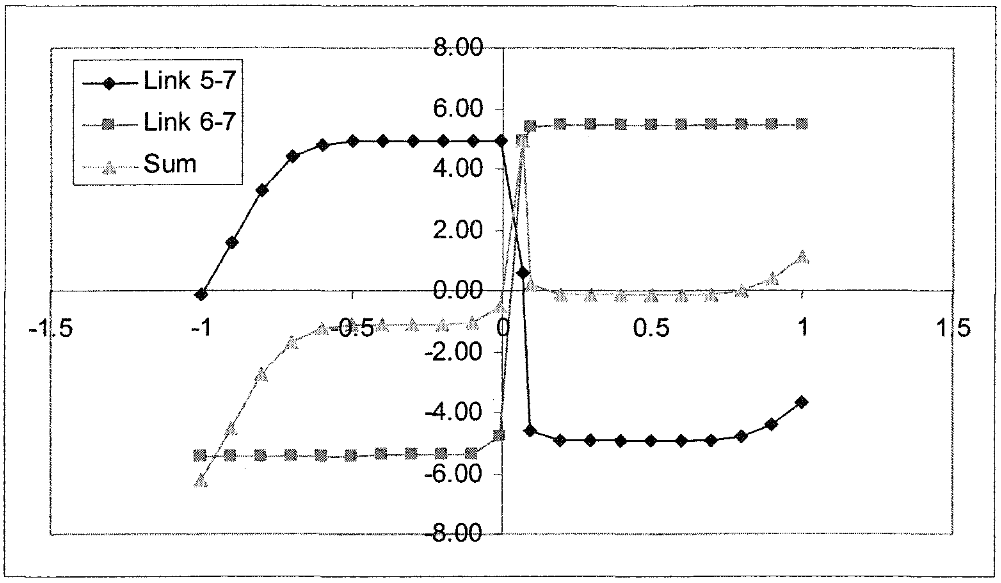

Figure 3-15 Input Charts to the Output Neuron

Second, one of the two curves must be reflected horizontally to allow an intersection of two curves in the peak area. This is achieved by adjusting the connection weight on Link 5-7 into a negative value (-4.94).

Third, the amplitude of the steep grade must be amplified to produce a peak. In this case, that is done by adjusting the connection weights on Link 5-7 and Link 6-7 to an absolute value greater than 1. Here they are –4.94 and 5.42. As discussed above, the process to produce a peak or valley is much more complicate for the symmetric logistic function.

It should also be pointed out that the Link 5-7 curve in Figure 3-15 can only be produced in a two hidden layer structure with the symmetric logistic function. This is because the Link 5-7 curve is a non-monotonic curve. There must be another hidden layer before Neuron 5 to produce the Link 5-7 curve. In this case, three neurons in the first hidden layer (Figure 3-14) are needed to produce three link curves, which are combined into the non-monotonic sum curve (Figure 3-16). Link 2-5 is responsible for the left end of the sum curve. Link 3-5 is responsiblé for the middle of the sum curve. And Link 4-5 is responsible for the left end of the sum curve.

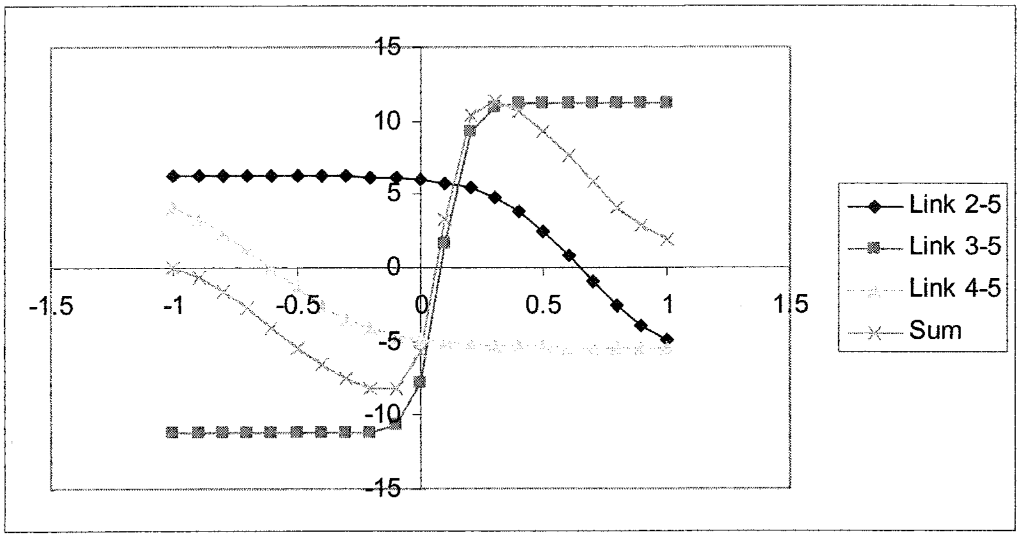

Figure 3-16 Input Chart for Neuron 5

To reproduce the source data in case 2 with the symmetric logistic function, a neural network structure of 1-2-4-1 is needed. The partial connection weights and thresholds are shown in Figure 3-17. The analysis of the output neuron is shown in Figure 3-18, in which the features of the input curves are very similar to those of Figure 3-15. The general principles for analyzing the curves are the same. It is interesting to see that two extra neurons are needed in the second hidden layer to cleanly produce the second peak.

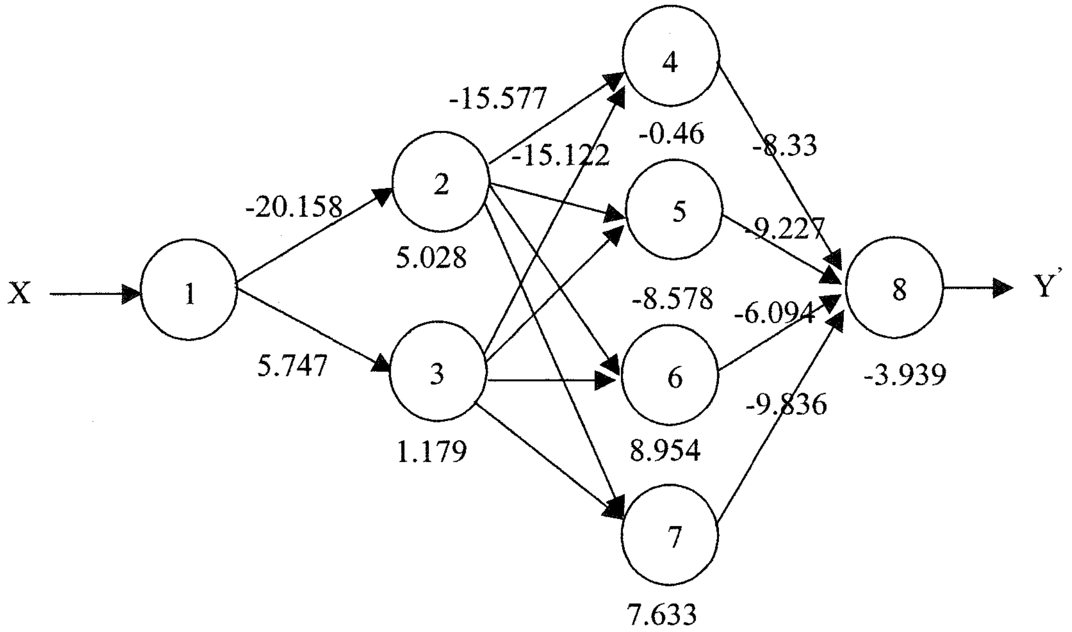

Figure 3-17 The Neural Network Structure with Symmetric Logistic Function for the Case 2 Data Set

There is another interesting note on how the non-monotonic Link 4-8 Curve is generated by two neurons (Figure 3-19) instead of three neurons shown in Figure 3-16. In Figure 3-16, the non-monotonic sum curve, which changes the trend of the function twice, is generated by combining three link curves (Link 2-5, Link 3-5, and Link 4-5) before entering the activation function. The activation function only serves to smooth the sum curve to produce Link 5-7 in Figure 3-15, which also has a similar feature of twice changing its trend. On the other hand, the sum curve in Figure 3-18 only changes the trend once before entering the activation function. The feature of changing the trend twice in the Link 4-8 Curve in Figure 3-18 is actually produced by the activation function. With fewer neurons (resource) available to complete the same task, the learning process is very delicate and it is harder for learning algorithms to converge on.

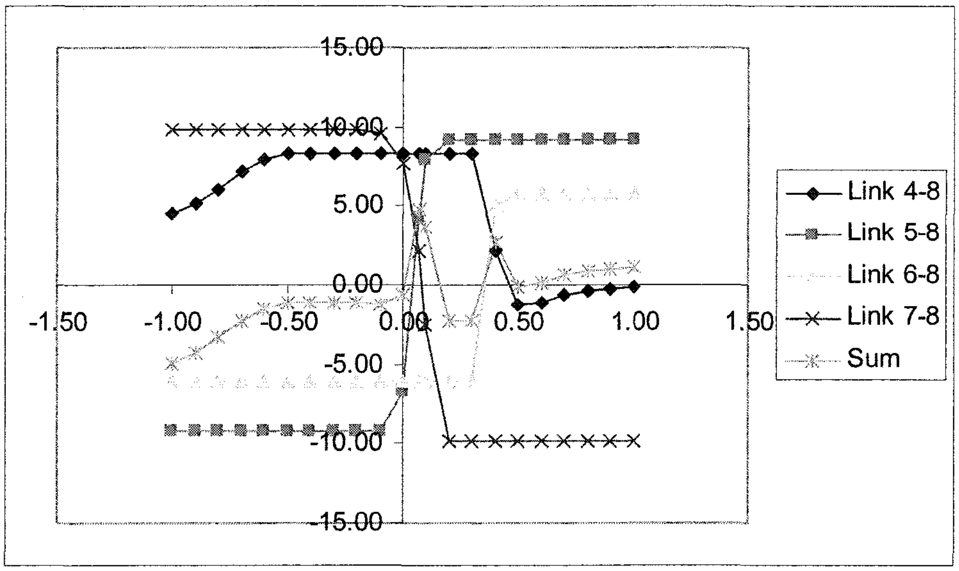

Figure 3-18 Input Charts to the Output Neuron

As shown before, for the Gaussian function, only one neuron is required to memorize the location of a peak, and only one hidden layer is required to reproduce the peak. For the symmetric logistic function, two neurons are required to memorize the location of the same peak, and two hidden layers are required to reproduce the peak. Thus, it is not difficult to conclude that non-monotonic activation functions such as the Gaussian function has more capacity to represent the non-monotonic features of the source data than the monotonic activation function such as the symmetric logistic function. Therefore, it is recommended that whenever there is non-monotonic feature presented in the output parameter such as peaks and valleys, the non-monotonic activation should be used to allow fast convergence and a simplified design in architecture.

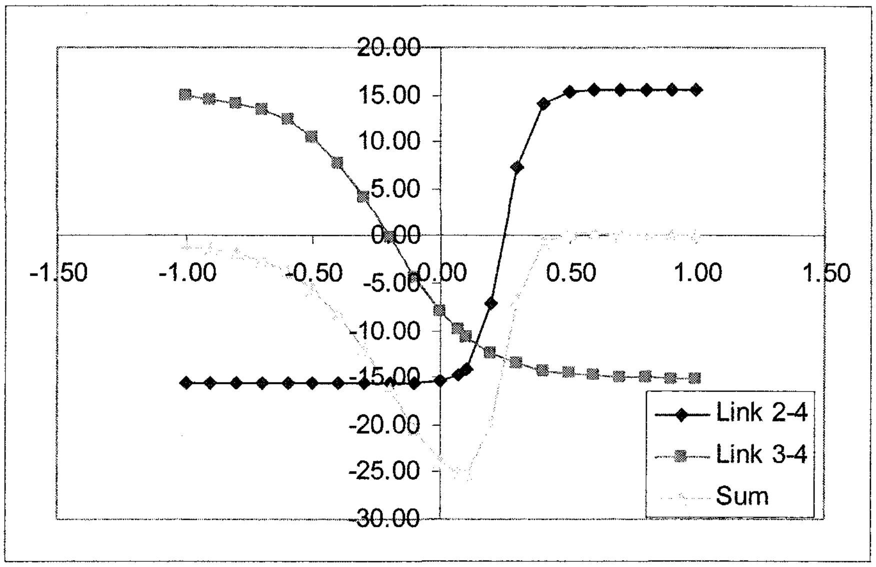

Figure 3-19 Input Charts to Neuron 4

Case Study 3: Robustness of Neural Network

Robustness is a key feature of neural network modeling. Neural networks are robust in both their learning capacity and their prediction capacity. If sufficient data are provided, a neural network can learn the data with minimum supervision. As shown in the previous case studies, neural networks can even learn the noise data adequately. Neural networks are also robust in its prediction on the unseen data after the model is built. Neural networks deduce their predictions on the unseen data by comparing the unseen data to the existing features stored in the models. This prediction methodology based on the strategy of deduction by similarity is logical and fault-tolerant. The objective of this case study is to analyze the fundamental source of robustness of the neural network from its architecture.

The robustness of a neural network in learning is mainly due to its non-linear composition structure, a network of individual processors, the neurons. This networking structure provides the freedom to tone either a neuron or several neurons in a network layer to handle a specific task. As shown in Case Study 1, a neural network model learns both the noise location (the peak) and the base function (y = x), and stores them in separate neurons. Neuron 3 stores the noise location. Neurons 4 and 5 are responsible for simulating the function of y = X. Neurons 4 and 5 have to work together while Neuron 3 can act well alone.

The networking structure of neural networks also allows linking multiple neurons across the network layers to handle more complicated tasks. If the neurons in one network are insufficient to store a feature, several neurons cross the network layers will be grouped together to complete the task. As shown in Case Study 2, the neural network with the symmetric logistic activation function requires neurons in two hidden layers to reproduce the noise peak. Thus, networking structure is the key to explain why neural networks are so robust in learning.

Neural networks are also flexible in adjusting themselves to learn new features. In Case Study 2, to learn a second noise peak, the neural network simply uses one additional neuron to store the new feature. The rest of the features are still preserved in other neurons. The following Case Study 3 suggests how a neural network handles the excess capacity in its architecture.

As discussed in Case Study 1, with the Gaussian activation function, the minimum requirement to reproduce the noise data in Table 3-1 is a networking structure of 1-3-1. It is interesting to examine how a neural network adjusts to learn the same problem with a 1-4-1 structure with one extra neuron. The resulting network is presented in Figure 3-20. The inner workings of the network are analyzed through the input to output chart (Figure 3-21).

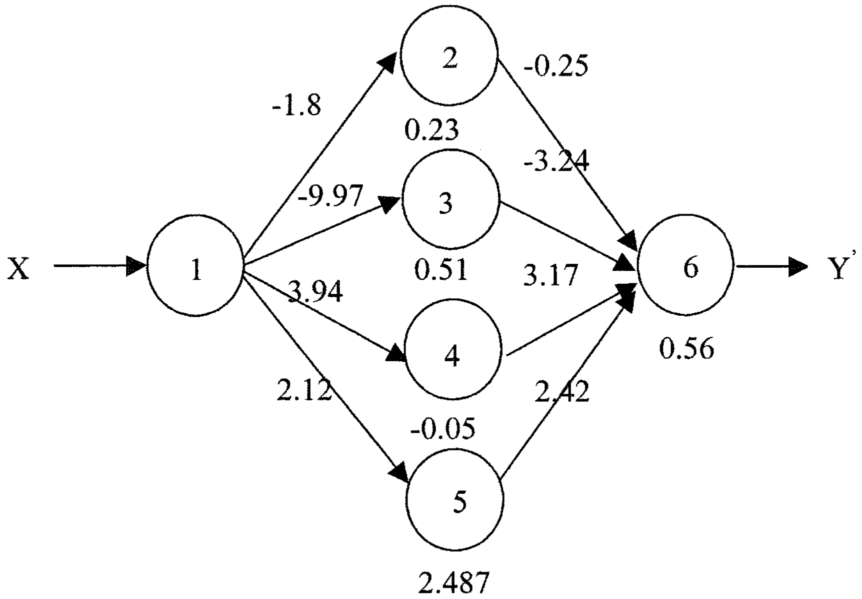

Figure 3-20 The Excess Capacity Study

By comparing Figure 3-21 to Figure 3-10, it can be concluded that both figures are almost identical, except in Figure 3-21 there is an extra flat line (Link 2-6) close to the x-axis. This line contributes virtually nothing to the sum curve. Thus the neuron responsible to produce that line becomes an inactive node. This demonstrates that excess capacity in the neural network architecture does not necessarily translate into better accuracy in prediction. In this case, there is no extra feature to memorize or to improve on. Thus, the network sets the extra neuron (Neuron 2) to be inactive. This extra neuron does not negatively impact the neural network prediction capacities either.

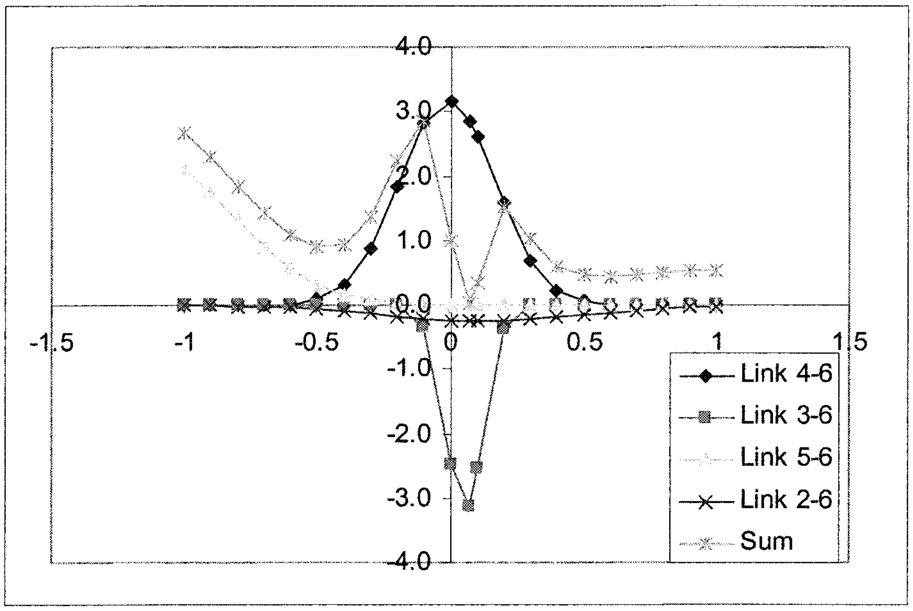

Figure 3-21 Input Charts to the Output Neuron in the Excess Capacity Study

Once a model is built, neural network is also robust in prediction mainly due to its fault-tolerant feature. The fault tolerance is accomplished within the network architecture in three ways. First, the summation function inside the neuron makes sure all inputs are in linear terms. Thus, the impact of an input spike would not be amplified by the other independent inputs. There is also a good chance that a spike in any particular input could be smoothed out after the summation because the spiked input carries less weight than the other inputs to the final sum. Second, if a relatively flat activation function such as symmetric logistic function is used, any sudden increase in any particular input can be scaled down by the activation function in the hidden layer before the spike reaches the output neuron. Third, if a spike were to locate in the saturation zone of an activation function, the response would be flat. For example, a scaled input of 5 or a scaled input of 50 to the symmetric logistic function produces no difference in output. In both cases, the outputs are exactly 1. However, for outlier, it usually has multiple inputs that are dramatically different from the normal values. The spikes on multiple inputs are very likely to result in a spike in the output because its chance of getting larger sum is much higher with extreme values on several inputs.

Case Study 4: Sensitivity of Neural Network Predictions

Sensitivity analysis examines the interpolation and extrapolation capacity of a neural network. Previously without detailed knowledge of the internal workings of neural networks, the only way to analyze the sensitivity of a neural network model is to plot response surfaces of the interested parameter(s) under various scenario settings. Although this method is satisfactory, it does not provide any explanation on the model behavior. A better approach is to analyze the neural network interpolation and extrapolation capacity with the graphical interpretation technique. This case study demonstrates that sensitivity of neural network is affected by the locations of learning samples. In other words, proper selection of the learning examples has significant impacts on how neural network predicts.

To simplify the analysis, this case study reuses the data and the model from Case Study 1. First, the neural network model shown in Figure 3-9 is used to interpolate the region where $0 \leq X \leq 1.5$ and the region where $2 \leq X \leq 3$. The model is also used to extrapolate the region where $X > 3$. The result (the Network Prediction 1 curve) is shown in Figure 3-22. The interpolations are poor and the extrapolations are quickly saturated at y =3. Obviously, more learning points are needed.

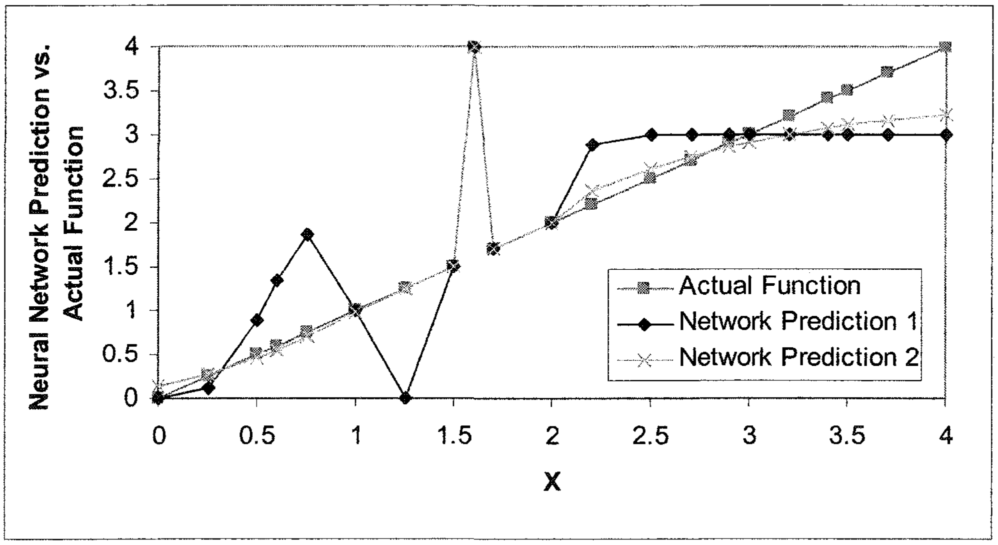

Figure 3-22 Sensitivity Study of the Noise Data

To improve the model, three more data points from the region where $0 \leq X \leq 1.5$ were added to the training data set, and the model was retrained with the same architecture. A much better sensitivity study (the Network Prediction 2 curve) was achieved. To analyze the retrained model in detail, the change of the input links over input X at the output neuron is shown in Figure 3-23.

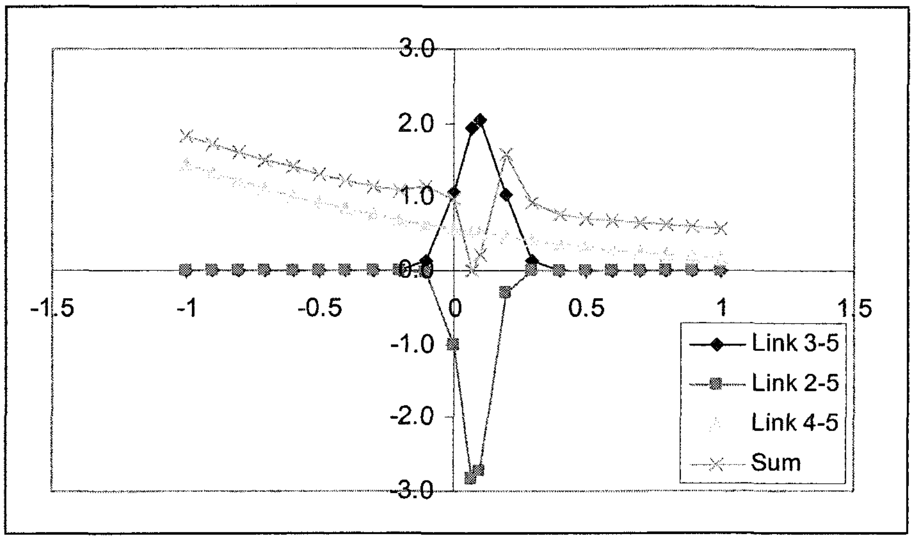

Figure 3-23 Input Charts to the Output Neuron

By comparing Figure 3-23 to Figure 3-10, it can be concluded that although the basic task of each individual neuron does not change, the shapes of their output curves (the input links to the output neuron) are adjusted to the new data points. Subsequently the sum curve is able to represent the linear function better. The interpolations at the region where $2 \leq X \leq 3$ are much improved and the extrapolations are more accurate and saturated at a slower rate. Nevertheless, the extrapolation is still not satisfactory because it does not reflect the true trend of the actual function. More data in the extrapolation zone are needed to improve the prediction in that area.

This example illustrates that the interpolation capacity of neural network model can be greatly improved by providing more learning samples at strategic locations. In this case, additional learning points are set at the locations where the curvature of the old prediction line changes (Network Prediction 1 line in Figure 3-22). This effectively tells the neural network model to fix the sensitivity in those regions where curvature changes. However, it is important to note that more learning points does not necessary improve the accuracy further. Any more training points on the non-curvature variance spot provide no additional new feature to the neural network and thus, no further improvement on the prediction can be result.

The extrapolation capacity of neural network model is usually limited. This is because extrapolation points are often located close to or within the saturation zone of the activation function, and the response at this region is always flat. Therefore, the confidence in ANN extrapolation is often lower than that of interpolation. This emphasizes the importance of a data screening system to flag extrapolation points before feeding the data to neural network models. Proper warning on the model prediction improves the reliability of the ANN model. This data screening system is the key subject discussed in Chapter 4.

Sensitivity of neural network around the noise data is also an interest for this study. As discussed before, neural network does memorize features such as a pulse noise in this case study. Thus, it is very important to contain the noise and limit its impact only in its immediate neighborhood.

To study the method to limit the impact of noise, two correct data (X = 1.5 and X = 1.7) are placed in the immediate neighbor of the noise (X = 1.6), and the model is retrained. The correct data effectively limit the noise impact zone in a narrow region where 1.5 ≤X ≤ 1.7. If X =1.7 were not placed in the training data, the noise impact zone would be expanded in the region of 1.5 ≤ X ≤ 2.0 as shown in Figure 3-24.

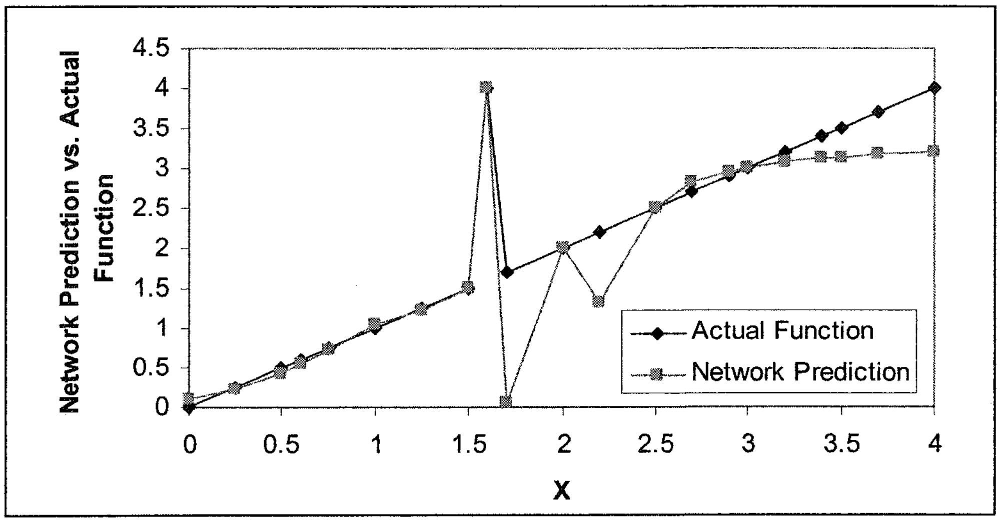

Figure 3-24 The Noise Impact Zone in the Sensitivity Study

The interpolations in the noise impact zone become poor with shape swings in both slope and magnitude. In other words, although the noise impact is contained within the impact zone, the impact magnitude changes much more dramatically. It is interesting to point out that the retrained models actually behaves like an over-trained model in the noise impact zone, and the previous model can be regarded as under-trained in the noise impact zone. Thus, it is very important to recognize and eliminate the noise from the training data in the source data analysis stage to prevent either under-trained or over-trained scenarios in the modeling stage.

Conclusions

The feed-forward neural networks are considered to be black-box models by many users. Although a single artificial neuron is simple and easy to understand, the networking structure of many neurons clouds our ability to understand them at the very fundamental level. That is at the level of connection weight, threshold, and activation function. The lack of understanding at the internal workings of neural network raises many critical questions, such as how to judge whether a neural network model has learnt the correct features of a study domain, or how to measure the confidence level of neural network predictions on unseen data? Without clear answers to these questions, it is difficult to justify putting neural network models into real world applications. Thus, it is very important to find an approach to decipher neural networks.

Although it seems that solving such a critical problem requires unusual means, in this chapter, the problem was addressed based on a very simple and fundamental mathematical understanding of neural networks. That is most feed-forward neural networks are functions. After establishing this fact, a graphical interpretation technique was presented to analyze a neural network model as a complex function. Aided with this graphical interpretation tool, several case studies were performed to analyze the general behaviors of the back-propagation neural networks. The studies include the basic functionality of each internal component of a neural network, the impact of noise on the model training process, the robustness of neural network in both learning and prediction, and the general sensitivity of a neural network model.

With this graphical interpretation tool, it was illustrated that the connection weights and the thresholds inside the neural network model serve to transform the shape of function curve. The threshold shifts the function curve vertically. The connection weights can stretch, compress, and reflect the function curve. The combination of these transformations makes it possible to reproduce almost any shape for an output function. This is why neural networks are so robust in many aspects of the numerical modeling.

It was also found that non-monotonic activation functions in general can store more features with fewer resources (neurons) than monotonic activation functions. This is because non-monotonic functions, as suggested by the name, have more curvature changes. Thus, their function curves are more flexible and easy to shape. However, non-monotonic activation functions are more likely to have poor interpolation. This is because more training data are needed to set the curvature at various parts of the function curve. If training data were missing at the spot where curvature changes, poor interpolation would happen as a result of under-training. Thus, it is very important to examine the quantity and quality of the source data before determining which type of activation function to use.

Noise records have a significant impact on both the internal structure and the prediction behavior of neural networks. Case studies illustrate that inside a neural network, noise is treated as a feature of the source data and its location is memorized by one or more neurons. As a result, the noise exerts an impact zone with the location of the noise as its center. Any interpolation within the noise impact zone is hypersensitive. The predictions tend to be dramatically higher or lower in value. In other words, the neural network model is over-trained in the noise impact zone.

This chapter also explained two key features, robustness and sensitivity, of neural network models. The robustness of neural network shows not only in the learning process but also in the model prediction. The robustness in learning is mainly due to its non-linear networking structure of the neurons. The interconnected networking structure provides various ways to capture the features of a study domain. A simple feature can be stored in a single neuron. A complex feature can be stored in a group of neurons, even cross the layers if necessary. The robustness in prediction is mainly due to neural network's ability to deduct the prediction from the features of a study domain stored inside the networking structure. The non-linear inner structure of neural networks also minimizes the potential impact of a faulty input.

The sensitivity study explains the interpolation and extrapolation capacity of neural networks. It was found that correct interpolation capacity of neural networks relies heavily on proper selection of the training data. Noisy training data waste significant resources of neural network and exert a faulty response zone in the neighborhood of noisy data. The most valuable training points are the curvature changing spots on the function curve. Excess training data in other regions yield no better results. Extrapolation capacity of neural networks in general is fairly limited. This is probably due to the saturation zone of activation function inside the neural network. If an extrapolation falls inside the saturation zone, the sensitivity of the prediction is very low and close to zero as the prediction always returns the same value. Thus, it is important to have a data screening system to warn the extrapolation scenarios. Proper warning establishes accurate confidence levels on the neural network predictions. This subject is further discussed in Chapter 4.

So far the case studies in discussion have one input, and thus are one-dimensional. Most of the real world applications require multiple inputs, and thus are n-dimensional. The n-dimensional inputs definitely increase the variety of information available and introduce more features into the neural network model. Nevertheless, the graphical analysis technique is still valid for the n-dimensional problems. The summation function inside the neurons keeps n inputs in linear terms. Thus, a neural network with multiple inputs is still a function, and the approach to analyze the hidden layer neuron and the output neuron remains valid.

However, multiple inputs do mean that the source data are drawn from an ever-enlarged study domain. As the number of n increases over 3, our visual presentation ability of the data becomes very limited at best. As a result, the ability to visually interpret the source data diminishes. At the same time, properly selecting training data and detecting the potential noise remains as a very critical and difficult task. To solve the problem associated with analyzing source data in a multi-dimensional space, the next chapter will introduce a pattern recognition approach to modularize the study domain. This approach allows an individual neural network model to be built for each module of a study domain.

Thus, greatly improves the prediction accuracy of the model in terms of interpolation. It also improves the confidence in the extrapolation by adding a confidence check to the prediction, which is measured by the statistical distribution of the source data in that module.

References

Minsky, M. L., and Papert, S. A. (1988), *Perceptrons*, Expanded Edition, MIT Press, Cambridge, MA.

Chapter IV Design of Artificial Neural Network Systems for Complex Domains

— A Pattern Recognition Approach

Introduction

Artificial neural network modeling deals with a lot of data. Proper data handling is the critical factor to the success of ANN modeling. At the modeling stage, the training data set extracted from the historical data must adequately represent the features of a study domain. The key factors in selecting proper training data include the quality of the data, how much is required, and what types of data inputs are needed. Only a discreetly and thoroughly selected training data set can produce a reliable and accurate ANN model. At the model application stage, the input data to the ANN models must also be carefully screened to ensure the credibility of the predictions. As shown in the previous chapter, an ANN model can only learn the features represented in the training data set, and has limited ability to extrapolate outside its trained domain. As a result, for the practical application of ANN models, it is of utmost importance to ensure the input data are within the domain on which the ANN model is trained.

The type of study domain has a great impact on the data handling for ANN models. For clearly defined domains with set boundaries for their parameters, which have been termed a closed domain, it is relatively easy to determine whether the input data are within the model domain. However, for many real world problems, this task is not so clear-cut. Most real world applications tend to have numerous governing factors without clearly defined boundaries. These more complex domains have been termed as open domains.

Presented in this chapter is a pattern recognition approach that can define the boundaries of a complex domain in a systematic manner. This approach can be used to assess the quality of historical data, detect the outliers, and determine the types of data required for modeling. It can also be used to subdivide a complex domain into sub-domains. By dividing a large domain into smaller sub-domains in which reasonable boundaries can be set, it effectively changes an open domain into a conditioned closed domain. One ANN model is then developed for each sub-domain. This approach can accurately model the whole complex domain in situations where it would be an impossible task for one single ANN model.

In general, pattern recognition is a technique used to divide a large quantity of data into cohesive groups by their characteristics. This division process is called classification. During classification, those individuals with similar characteristics are grouped together to form a class. As a result, each class possesses common group characteristics. These group characteristics are usually quite distinct across the classes, and thus fundamentally differentiate the classes from each other. The group characteristics are the equivalent of a gravity center of a class. Each member of a class not only possesses the common characteristics of the group, but also displays its own characteristics, which serve to distinguish one individual from another within a class. The closer a data point is to the group gravity center, the weaker is its individuality. On the other hand, the further a data point is away from its class center, the stronger is its individuality.

The pattern recognition approach has been widely used in many research fields and has many successful applications in civil and environmental engineering. Pattern recognition has been used in conjunction with fuzzy logic and artificial neural networks in areas of transportation engineering such as freeway incident detection (Adeli and Karim, 2000), traffic flow status classification (Yang and Qiao, 1998), and roadway safety classification (Sayed, et al 1995; Faghri and Hua, 1995). Pattern recognition is also used in construction engineering to identify structural health (Sohn, et al, 2000; Chao and Cheng, 1998) and structural model identification (Chen, et al, 1995), and in material engineering to analyze images from construction materials (Rajasekaran, et al, 2000; Cheng, et al 1996).

Recently, a new ANN modeling technique has emerged to model the noisy data problems (Tetko and Villa, 1997; Yang and Chang, 1998). This technique combines the pattern recognition technique into the artificial neural network modeling approach. This chapter will discuss this technique in detail. First, to understand the origin of this technique, the concept of human recognition is introduced. The discussion is then expanded into how to emulate human recognition mathematically, and eventually implement the concept with a type of artificial neural network called Kohonen network. Finally, it will be demonstrated how to use a hierarchical ANN modeling structure to divide the original open domain into relatively closed sub-domains for better modeling accuracy, and how to integrate the sub-domains together with an input data classifier. This technique has been used to model the performance of filters in a water treatment plant (Tupas, et al, 2000), and the treatment of oil sands which is presented in Chapter 6 of this thesis.

The Concept of Human Recognition

Pattern recognition is a process simulating how humans learn. In a human recognition process, there are actually three major steps. Human recognition starts with an observation. Then the individual will try to make sense of what is observed. Finally, based on the understanding of observation, the individual can make a judgement on future similar observations.

In other words, the human individual observes to collect information. Humans use biological senses and some additional abstract information to describe objects. For example, an object on the table is a cup because the object contains several unique characteristics. The human eyes sense that the object possesses certain size, color, texture, and shape (observable information). The combination of these characteristics also induces a conclusion that the object could contain certain amount of liquid (abstract information). In our observation, each object is described by its observable features and associated abstract information instead of being considered as a single entity without any description and relation to other entities.

In order to utilize this vast amount of collected information for future reference, the individual must rationalize the information. This rationalization process is usually called learning by classification. By classifying the observed information, a system is created to group objects with similar characteristics together. Each object in the group is somewhat distinct to others but also has common feature together. In the example of a cup on the table, the system will compare the features and characteristics of all kinds of cups it has observed and conclude on an object group called cup.

With this system, any new observation related to the system's previous observations can be immediately recognized and distinguished. For example, a blue cup on the table will be now recognized as a cup with a special flavor - color blue. The system also implies that the blue cup is a cup and not a bowl because bowl is a different object group. In this sense, the system in fact distinguishes a cup from a bowl. In fact, this classification system is also in a state of continuous learning. If any observation does not coincide with the existing groups, then a new group would be generated. Continuous learning preserves the integrity of the system. The ability to learn, rationalize, identify and distinguish completes a human recognition process.

The Concept of Pattern Recognition

Since a pattern recognition process mimics the human recognition process, the key concepts discussed in the previous section are translated accordingly. In the pattern recognition approach, an object is called a pattern. Subsequently, a pattern possesses observable attributes and abstract attributes. These attributes are called the pattern attributes. Just like a human recognition process, there are also three steps to create a pattern recognition system to describe a study domain. First step is to collect the patterns that describe the study domain. Next, these patterns must be separated into groups, in which each group presents a distinct concept. Based on these distinguishable concepts, a system will then be created to describe each concept and collectively the whole study domain. Third, any future patterns collected from the study domain can be recognized by the system either as a member of the existing components of the system or a part of a new component that must be learned to complement system's understanding of the study domain.

The key component of this pattern recognition modeling is the second step, which in turn can be divided into three processes: separate the patterns into groups representing distinct concepts, build models capable of describing each concept, and unite the models together to form a system. The process of separating patterns into groups is called clustering. Within each cluster, member patterns are closer to each other and their own centers rather than to the centers of other groups. Based on the clustering, models are built to describe each cluster. This process is called modularization. The modularization breaks down a complicate study domain into many much better defined sub-domains. The modeling within these well-defined sub-domains is thus expected to have better accuracy than modeling the whole domain at once. The modularization also allows the easy modification of the system to accommodate changes in the study domain. Modules of the system can be easily upgraded or added to describe the new system instead of scrapping the whole system and restarting. The caution with the modularization approach is that the number of the sub groups should be sufficient to span the whole domain.

Cluster and Its Measurements

There are two basic concepts related to clustering. That is how to measure the similarity of the patterns and what is typical representation of the cluster. Mathematically and most commonly, the similarity of the pattern is expressed in Euclidean distance, denoted by $\|X\text{-}Y\|_2$, where X and Y are two patterns. Formally, the Euclidean distance between two patterns X and Y is defined by their root sum-squared error

$$\|X\text{-}Y\|_2 = \{\Sigma_{(n=1,N)} (X_n - Y_n)^2\}^{1/2} \qquad (3\text{-}5)$$

where n is the number of the attribute of the pattern. A smaller Euclidean distance implies greater similarity. On the other hand, a larger Euclidean distance implies greater dissimilarity.

The typical representation of the cluster is the center of the cluster. This center is defined where sum of squared distances to all the patterns in the cluster is minimum. It can be easily proved that the center is actually the simple average of each attribute of the patterns as expressed in Equation 3-6 (Looney, 1997):

$$Z_n = (1/M)\ \Sigma_{(m=1,\,M)} X_n^{(m)} \qquad (3\text{-}6)$$

where Z_n is the center of the cluster;

n is the number of the attribute associated with each pattern;

M is the total number of the pattern in the cluster;

X^m is the mth pattern in the cluster.

Statistically, Z_n is the sample mean (center) of a true cluster. As the number of the m is increased to be sufficiently large, Z_n will gradually approach the true center of the cluster. Practically, when m is greater than 60, the size of the cluster is considered large enough to warrant Z_n as a good representation of the cluster.

With the definition of the center of a cluster, each pattern of the cluster now possesses an additional cluster attribute, which is its Euclidean distance to the cluster center. According to the central limit theorem, if the number of the pattern is large enough (or greater than 60 in general), the distribution of these Euclidean distances can be considered as a normal distribution. With this knowledge, it is now possible to measure the cohesiveness of the cluster by measuring the standard deviation of the distribution. A large standard deviation may indicate a loose cluster structure, while a small standard deviation suggests a cohesive and tight group of patterns. It is also possible to determine whether a pattern actually belongs to a cluster or not. For example, it can be defined that a true member of a cluster must have an Euclidean distance to the cluster center that is smaller than 2.5 standard deviations of the cluster. A weak member of the cluster can be defined as whose Euclidean distance to the center is between 2.5 to 3 standard deviations. Any pattern whose Euclidean distance to the center is greater than 3 standard deviation does not qualify as a member of the cluster at all.

The same concept can also be used to examine whether there is overlapping among the clusters. If two clusters were not separated cleanly, their boundaries would overlap. For those patterns in the overlapped space, their Euclidean distances to the centers of both clusters would be under 2.5 standard deviation and very close in magnitude. If there are many patterns in the source data have this problem, then there is an overlapped space existing among clusters. Therefore, the clustering process is insufficient and needs to be improved.

ANN Clustering with Kohonen Network

The clustering of the source data in this project is accomplished with a special type of neural network called Kohonen network. A Kohonen network is an unsupervised neural network, which means there is no need to provide the correct answer for the Kohonen network to learn explicitly. Thus Kohonen network is especially suitable for separating large groups of data into smaller clusters without supervision on how the separation should be done. Once the user decides the number of categories desired, the Kohonen network will automatically separate the original database into that number of categories based on the concept of measuring Euclidean distance to the related cluster centers.

A typical learning example is illustrated as below. Assume the user prefers to separate the original database into N categories. The Kohonen network randomly initializes K cluster centers, where K is usually less than N. When a pattern arrives to be classified, the Euclidean distances of this pattern to all K cluster centers are calculated. The center with minimum distance to this pattern wins the pattern as its member. As a result of winning this pattern, the winning cluster gets an update on the values of its center to reflect the fact that a new pattern is included in the cluster and more information is available to help the eventual discovery of the true center of the cluster. As the learning progress, the number of categories may increase as needed, but will not exceed N.

Although the Kohonen network is quite robust in category separation, it does have some drawbacks. The user needs to determine what is the optimum number of categories for the source data to be divided into. If the number of categories guessed is smaller than the true value, the Kohonen network will force the separation of the source data into that number of categories. As a result, it will create categories with mixed and inconsistent features. If the guessed number of categories is larger than needed, the Kohonen network will force the split of one or more true categories, and create potentially overlapping categories. To solve this problem, some heuristic rules and quantitative measurements such as the boundary conditions should be applied to search for the optimum solution.

The second requirement for the Kohonen network to function properly is that the size of the categories must be larger than 60 in practical sense to warrant the normal distribution assumption as discussed in the previous section. The larger the category size is in the initial separation, the better the cluster center truly represents the typical features of the cluster. However, if the user decides to further split the category into smaller groups call classes, the size requirement is not that stringent. This is because the true representation of the classes is not typically as significant as for the category. Models are built based on the category and not based on the smaller scale classes.

Model Structure for the Pattern Recognition System

For complex domains the classification approach presented previously can be used to develop a number of neural network models to model the overall complex domain. In general, it can be difficult to develop a single ANN model to describe the whole complex domain. By splitting the complex domain into smaller cohesive sub-domains, it is possible to develop a model for each of these sub-domains, and the modeling would be much more accurate. To accomplish this, a hierarchical modeling structure is required with a data filter to link the models together (Figure 4-1). This data filter determines which category and hence which neural network model is appropriate for the input data. An additional benefit of this system is that the data filter will also signal if the input data does not belong to any of the sub-domains and alert the user to a possible problem.

To accomplish these requirements, the design of the data filter is based on the Euclidean distance concept and under the assumption that the Euclidean distances from the cluster members to the cluster center is normally distributed. As new data is entered into the system, the data filter calculates the Euclidean distance of the new entry to all existing cluster centers. The cluster with the minimum distance to the new entry is identified. Then the data filter further examines whether the Euclidean distance between the new entry and the cluster center falls within the requirements to become a true member of the cluster. For example, as discussed before, a true member of the cluster can be defined as whose distance to center is smaller then 2.5 times of the standard deviation. If the condition is met, the data are channeled into the model responsible for that cluster for further processing. If the condition is not satisfied, then the new entry is considered as a member of a previously unidentified category. A proper flag should be set before admitting the new entry into the database. No proper analysis for the new entry can be done at this moment. If sufficient quantity of the entries of this sort is encountered, a new category and its corresponding model must be created to accommodate the change.

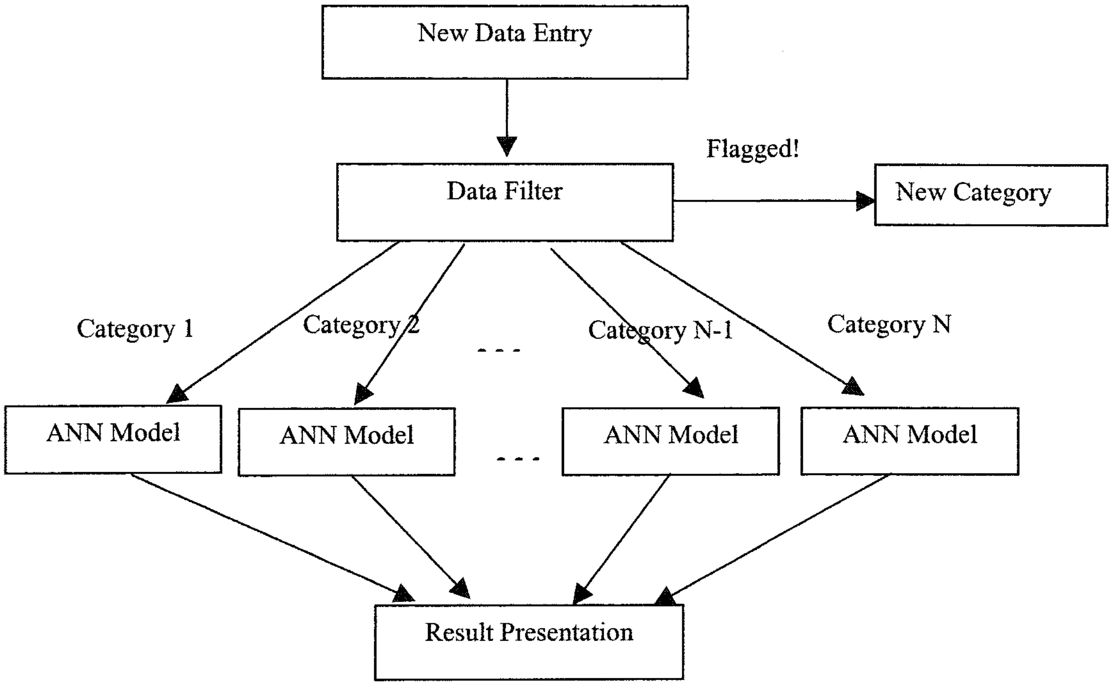

Figure 4-1 Hierarchical Model Structure of Pattern Recognition System

Pattern Recognition for Source Data Analysis

Another important application of pattern recognition in the ANN modeling approach is in the source data analysis. Classification of the source data in the initial screening process can easily identify the outliers in the source data because their characteristics are far different from the common records. In terms of Euclidean distance, the distances of the individual outlier to the existing class centers are usually greater than three standard deviations in distance to each class. It should be pointed out that these individual outliers do not aggregate together either, otherwise they would form a new category and become a legitimate group instead of being outliers.

Pattern recognition can also be used in the detailed analysis of the source data in order to select the most representative data to train the ANN model. The source data can be broken down into categories and then into classes under each category. Within each class its members can be further divided into subclasses to ensure higher degree of similarity within the subclass. This sub-division process can be continued until the members of the subgroup are very consistent in characteristics. The members of each subgroup can then be equally distributed into the training, testing, and production data sets. This data extraction approach will guarantee every important feature (distinct characteristics of the subclasses and subgroups) of a study domain was trained in the ANN model and at the same time does not overwhelm the ANN with too many details of the individual data points. Thus, this approach greatly improves the quality of the model and at the same time avoids the chance of over-training the ANN model.

Conclusions

Pattern recognition is an artificial intelligence technique that simulates human recognition process. Pattern recognition is an established technique and has been successfully applied to many fields of science and engineering. This chapter briefly introduces the origin and the mathematical background of the pattern recognition technique, explains the potentials of this technique, and eventually illustrates how to combine the concept of pattern recognition into artificial neural network modeling.

This chapter highlights a pattern recognition approach to build ANN models for complex domains. Modeling complex domains especially the open domains with artificial neural networks is always a difficult task. This is because many cause-effect factors of the domains might not be always available for modeling, and many parts of the domains could exhibit distinct characteristics. The unique feature of this approach is to divide a complex domain into several smaller and cohesive sub-domains, model each sub-domain individually, and then integrate the models together with a data classifier (filter) in a hierarchical modeling structure. This subdivision creates relatively closed sub-domains, and thus greatly improves the modeling accuracy.

In the model building stage, the pattern recognition technique can be used to detect outliers and to select most representative data to train an ANN model. Outlier usually has its own distinct characteristics and does not group well with the common records. Thus it is easy to detect through the classification process. Classification can also be used to further separate the source data into many small groups with very consistent characteristics. Each data point in a small group is a fair representation of the group. The data within these small groups can then be equally distributed to the training, testing, and production sets to fully represent the original study domain. This approach guarantees every feature of the source data is properly presented, and avoids putting too many data points from the same group into the training set to overwhelm the ANN models with too many details.

In the application stage, the pattern recognition system also has an important capacity to detect those input data that are out of the normal range. Any new data residing outside the boundaries of the existing modules of a study domain would be flagged immediately and set aside for further interpretation. This approach avoids blindly applying the model to any out-of-bound data for meaningless predictions, thus, greatly reducing the chance of making a significant error.

The pattern recognition modeling approach proposed in this chapter was applied to the actual applications in Chapter 5 and Chapter 6, where details of the applications of this technique can be found.

References

Adeli, H. and Karim A. (2000), Fuzzy-Wavelet RBFNN Model for Freeway Incident Detection, *Journal of Transportation Engineering*, Vol. 126, No.6, pp. 464-471.

Chao, C. and Cheng F. (1998), Fuzzy Pattern Recognition Model for Diagnosing Cracks in RC Structures, *Journal of Computing in Civil Engineering*, Vol. 12, No. 2, pp. 111-119.

Chen, H. M., Qi, G. Z., Yang, J. C. S., Amini, F. (1995), Neural network for Structural Dynamic Model Identification, *Journal of Engineering Mechanics*, Vol. 121, No.12, pp. 1377-1381.

Cheng, H. D. and Miyojim M. (1996), Synthesized Images for Pavement Management System Design, *Journal of Computing in Civil Engineering*, Vol. 10, No. 1, pp.60-66.

Faghri, A. and Hua, J. (1995), Roadway Seasonal Classification Using Neural Networks, *Journal of Computing In Civil Engineering*, Vol. 9, No. 3, pp.209-215.

Looney, C. G. (1997), Pattern Recognition Using Neural Networks: Theory and Algorithms for Engineers and Scientists. Oxford University Press, 198 Madison Avenue, New York, New York, pp. 458.

Rajasekaran, S. and Pai, G. A. V. (2000), Simplified Fuzzy ARTMAP as Pattern Recognizer, *Journal of Computing in Civil Engineering*, Vol. 14, No.2, pp. 92-99.

Sayed, T., Abdelwahab, W., and Navin, F. (1995), Identifying Accident-Probe Locations Using Fuzzy Pattern Recognition, *Journal of Transportation Engineering*, Vol. 121, No. 4, pp. 352-358.

Sohn, H., Czarnecki J. J., and Farrar, C. R. (2000), Structural Health Monitoring Using Statistical Process Control, *Journal of Structural Engineering*, Vol. 126, No. 11, pp. 1356-1363.

Tetko, I. V. and Villa, A. E. P. (1997), Efficient Partition of Learning Data Sets for Neural Network Training, Neural Networks, 10(8): 1361-1374.

Tupas, R-R. T., Stanley, S. J., Zhang, Q. (2000), Artificial Neural Network Modeling of Filtration Performance, Canadian Society for Civil Engineering 2000 Conference, London, Ontario, Canada.

Yang, H. and Qiao, F. (1998), Neural Network Approach to Classification of Traffic Flow States, *Journal of Transportation Engineering*, Vol. 124, No.6, pp. 521-525.

Yang, S. and Chang, K-C. (1998), Multimodal Pattern Recognition by Modular Neural Network, *Optical Engineering*, 37(2): 650-659.

Chapter V Upgrading ANN Model for Real-time Application

- A Case Study to Implement ANN Raw Water Colour Forecasting Model in the Rossdale Water Treatment Plant

Introduction

The artificial neural network (ANN) approach is a versatile modeling technique for complex non-linear situations. Since the introduction of this modeling technique into the field of environmental engineering in the late 1980's (Flood and Kartam, 1997), interest in ANN applications has increased at a steady rate, as indicated by an increase in the number of publications. Specifically, ANN applications have been developed in the areas of hydrology and water resources engineering (Daniell, 1991), groundwater remediation (Garrett et al, 1992), and biological waste treatment (Cote et al., 1994). More recently, ANNs have been used in transient drainage design (Shukla et al. 1996), river water color forecasting (Zhang and Stanley, 1997), water treatment process control applications (Zhang and Stanley, 1999), air quality monitoring (Hasham, Stanley and Kindzierski, 1998), and wastewater treatment plant operations (Boger, 1997).

With the rapid advance in neural network research, neural network modeling has evolved from an experimental tool into a technology ready for actual industrial applications. This article presents a case study of such evolutionary progress. This chapter will discuss the process of upgrading a neural network research model into an operation assistant tool with great potential for the industrial real time application.

Previously a neural network model was built to forecast the raw water colour in the North Saskatchewan River in Edmonton, Alberta, Canada (Zhang and Stanley, 1997). EPCOR Water Services recognized the potential of the model and decided to test the model at its Rossdale Water Treatment Plant (WTP) in Edmonton, Alberta, Canada. Thus, it was necessary to upgrade the ANN research model to meet the requirements of real time application. It is also important to emphasize that building a reputable ANN model is a continuous process. ANN model learns from the data it has seen. Thus, there is a boundary on its knowledge. From time to time, an upgrade of the ANN model may be required. This special requirement is intrinsic in the modeling process, and is also driven by the constant advances in neural network research. A neural network model can be improved by continuous learning in its study domain.

To integrate ANN model into a real-time operation process, there are many obstacles to overcome. These obstacles include obtaining the necessary data on a real time basis, designing the proper user interface, and educating the ANN system users on the neural network concept, benefits, and limitations. This chapter will address these concerns through a case study to implement an ANN raw water forecasting system in the Rossdale WTP. This chapter will assess the performance of the integrated system and recommend future actions to improve the system.

Key Issues in Designing an On-line ANN System

To adapt the neural network model for actual on-line application, there are many issues to consider. First, the designer must carefully assess the study domain, and be fully aware of the capacity and limitations of the neural network technique in the study domain. The closed non-linear study domain is the best candidate for the ANN technique. This is because in a closed domain the boundary of the knowledge is well defined, and the need to extrapolate outside the knowledge boundary is not frequent. On the other hand, the open study domain is difficult for artificial neural network to model because the knowledge boundary of an open domain is not clearly defined. As an ANN model is trained on the known events of the domain, its knowledge boundary is built on the known events. Thus, its knowledge boundary only covers a partial area of the overall domain. Since the system is an open domain, it is very likely that an unknown event relative to the ANN model knowledge boundary could happen. Thus, extrapolation out of the model knowledge boundary is necessary and results in lower prediction accuracy. To maintain reasonable prediction accuracy, frequent update (retraining of the ANN model) is required to cover more and more areas of the open domain.

Second, the designer should consider the data availability for the on-line real time application. Data availability on a real time basis is critical for the success of data intensive models such as neural networks. Lack of data could severely hamper the accuracy of the prediction. If any input data have to be assumed or guessed, the uncertainty is then introduced into the forecasting process and likely to reduce the forecast accuracy.

Finally, the designer must clearly communicate with the users on the capability and limitation of the ANN real time system. Since ANN systems tend to be black-box type models, it takes proper education to improve system users' confidence of the ANN system and eventually lead them to fully aware the potentials of the ANN system. Successful communication will also provide important feedback for design modification and implementation of the system, such as a user friendly interface.

Upgrading ANN Model for On-line Application

Once the principles of the overall design are determined, the next step is to implement them in the detailed model building process to obtain a suitable ANN system for on-line application. During the design phase, it is common that a test model is built to test the capability of the ANN system to handle the study domain. If the test model is successful, the test model will be upgraded to a full system.

The detailed model upgrade involves three major steps. The first step is to examine whether there is any significant change in the study domain after the test model is built. Any changes, which directly affect the modeled environment and its content, must be incorporated into the final ANN system. Thus, additional training of the test model maybe necessary to recognize the new feature(s) of the system.

The next step is to check whether there is any change in the data collection process or the availability of the new data. Any independent new data (uncorrelated to the existing input data) must be evaluated for possible improvement of the new model. Independent data usually bring in additional beneficial features to the system to make it easier for the system to distinguish hidden contents and learn the right concept. However, there is a risk that new data could also introduce noise into the ANN model and prevent the ANN models from learning the correct underlying logic of the domain knowledge.

Finally, the newest neural network modeling technique should be used to upgrade the model. Since neural network modeling is a fairly new research area, advances in research would render the old approach obsolete. The recommended detailed model upgrade process discussed below focus on two main areas: model retraining and model evaluation.

In terms of detailed ANN model training, the following aspects must be carried out to determine the proper retraining approach. Determine the noisiness of the source data, and adjust the training, test, and production data to the same level of noisiness. Assess any need for change in the input parameters. Simplify the architecture by using the old architecture and its slight variations as the initial test structures. Reduce the number of hidden layers and the number of neurons in the hidden layer. Review the possible implications of the systematic model training approach.

In terms of model evaluation, the key issue is to design and evaluate the models with the same criteria. The most recent trend of ANN modeling is to divide the study domain into sub-components, and then design neural networks specifically for each component. As a result, multiple models may be developed for different sensitivity in various areas. One model can be excellent for one area but may not be good for the other areas. Thus, when evaluating the model, it is more important to examine whether the model meets the designated criteria than the overall performance criteria. Finally, the evaluation should emphasize applicability and reliability for the final model.

Case Study: Upgrade and Implement the ANN Raw Water Colour Forecasting System

Previously, an ANN research model was designed to forecast the raw water colour in the North Saskatchewan River, Edmonton, Alberta. The results and findings were published in Water Research (Zhang and Stanley, 1997). The overall objective of the current project is to upgrade this research model for on-line application at the Rossdale WTP.

The EPCOR Rossdale WTP draws raw water from the North Saskatchewan River to produce drinking water. The major source of upstream contamination is runoff from forested and agricultural lands. During the year, the daily average of raw water colour can fluctuate from 2 TCU to 85 TCU. In the worst case, the instantaneous raw water colour can reach as high as 120 TCU. The annual colour pattern has peaks at spring runoff and during summer rains. Figure 5-1 illustrates the general pattern of raw water colour at springtime. It is typical to see two peaks: the first peak occurs when the snow melts in the lower elevations closer to Edmonton, while the second peak is due to melting in the upper elevations. Spring runoff is considered to have begun when the raw colour increases to greater than 10 TCU.

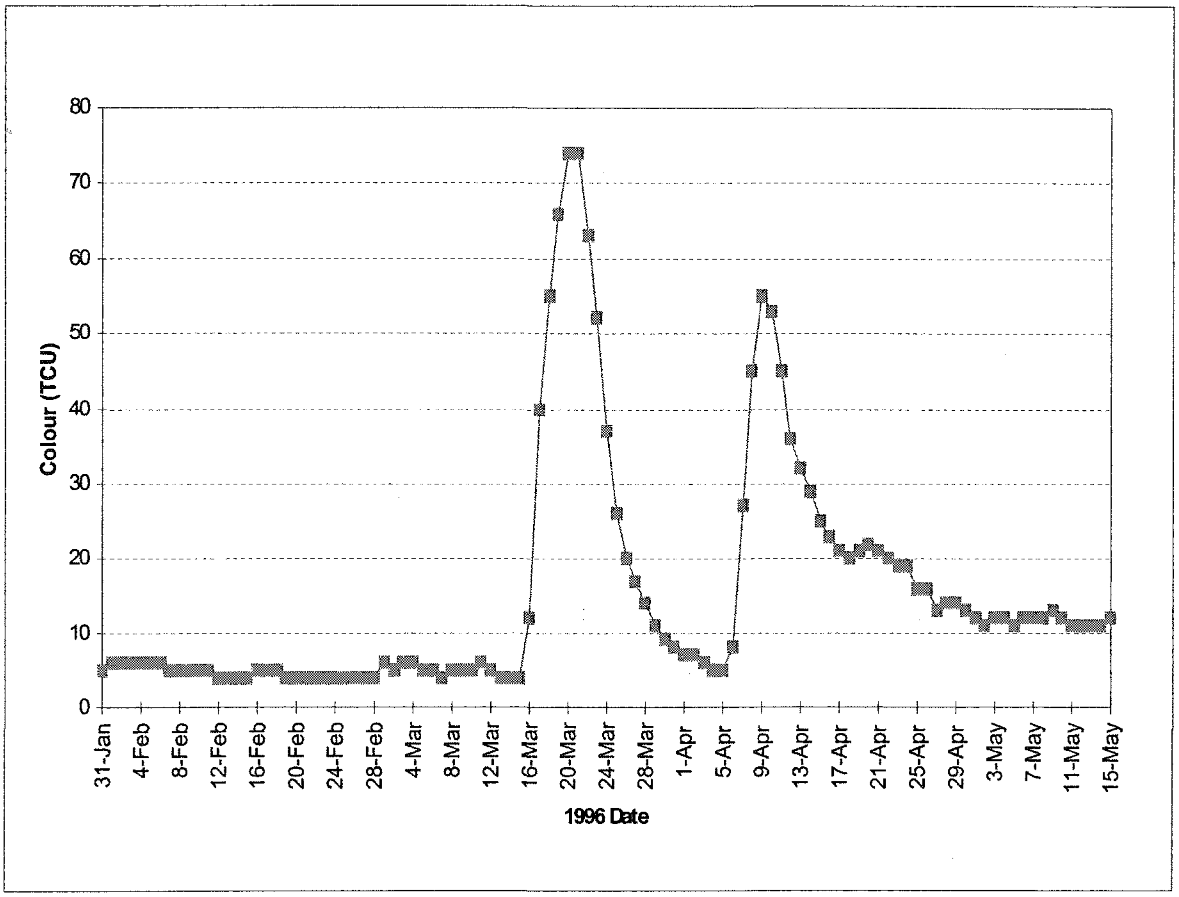

Figure 5-1 Typical Pattern of Raw Water Colour in Spring Runoff

Raw water colour plays an important role in the daily operation of EPCOR's water treatment plants as it provides a clear, immediate indication of changes in water quality. At all times of the year, alum dosing is based on the colour level. Colour becomes especially critical at spring runoff. At spring runoff, the presence of organic material as well as the by-products from microorganisms feeding on the organics has led to taste and odour problems with EPCOR's drinking water. For this reason, a great deal of effort has been placed on monitoring for spring runoff and treating the water to avoid taste and odour complaints. Powdered activated carbon (PAC) is used for taste and odour control. It is effective but expensive.

Due to the high cost of the carbon, the operators must closely observe when runoff begins to avoid feeding the carbon too early, thus wasting the carbon. On the other hand, missing the onset of runoff by even a few hours has led to customer complaints of taste and odour in the drinking water. It is therefore critical that the onset of runoff is accurately detected. The artificial neural network model was therefore developed to assist with predicting the start of spring runoff. The system is intended to forecast the raw water colour one day ahead to provide an early warning for the plant operators.

There are many intrinsic difficulties in forecasting the raw water colour of North Saskatchewan River one day ahead. First, the study domain is an open system. In a broader view, the whole watershed upstream has impacts on the raw water colour in the river. Upstream weather, surface runoff, and man-made events such as point-source pollution could all result in higher colour in the river. All of this information is difficult or impossible to convert into numerical values to train the ANN model. A more manageable approach is to limit the study in a smaller section of the whole watershed. For example, in this study, the immediate upstream watershed, from which water travels one day or less to reach the intake of the treatment plant, was chosen as the study domain to be focused on. However, due to the limited availability of watershed information, many impacts from upstream can only be represented by a surrogate parameter such as the river flowrate.

Second, the availability of the data for prediction on a daily basis is a significant limiting factor in design consideration. The availability of the data is represented in two aspects. First, is there an economically feasible sampling mechanism to extract the data? And secondly, how frequently is the data available? For example, although the designers have identified many colour-causing mechanisms such as the snowmelt in two creeks three-hours upstream, it is difficult to incorporate that information into the model because there is no weather station there to monitor the situation. The designers have to rely on surrogate weather information produced at the local airport. Even so, sometimes, weather information such as snow depth on the ground, or rainfall is still not available daily. The models have to be modified to accommodate the situation. Obviously, the prediction accuracy is affected and compromised.

Finally, it was the first time that an ANN system was implemented on-line and in a real time base in a water treatment plant, much experience was to be gained from the process. Since the ANN model is black-box type model, if the users don't know how to use it or do not understand the potential and limitations of the model, they will come to quick judgement and lose confidence in the model. Thus, designing an effective user interface and providing user education on the ANN system are crucial tasks of the project.

Model Upgrade: Comparison of the Approaches

The most efficient way to illustrate the improvements of the upgraded model is to compare the old approach with the new approach used to upgrade the model. The old approach utilized a database containing daily data from 1990 to 1994. It used the data of 1990-1993 as the training data set, and used the data of 1994 as the test data set. There was no data reserved as an independent data set (the production set) to test the reliability of the final product. There was no noise-level analysis in the source data. The modeling approach was trial and error with few heuristic rules guiding the search for the best model. The whole approach resulted in a black-box type model with little ability to explain the modeling process and the final model. There are also many time series parameters in the research model. Time series parameters are potential sources for a lagged prediction in real time forecasting. For example, if the raw water colour at Day t is used as part of the inputs to predict the raw water colour at Day t+1, in many situations, the predictions for the colour at Day t+1 tend to mimic the values of colour at Day t, and thus result in a lagged prediction. However, if the raw water colour at Day t is not used as part of the inputs, the problem of lagged prediction is significantly reduced.

To upgrade the model, the first step is to redefine the study domain and thus select the right data for the new domain. As mentioned before, the new domain is the watershed immediately above the water treatment plant where raw water travels 1 day to reach the plant. Since 1990 and 1991 data contains information outside this domain, they were not used for the upgrade process. With additional information from 1995-1997, the new database contains daily information from 1992 to 1997. Since the database exhibits an annual cycle pattern (Zhang and Stanley, 1997), the 1992-1995 data were used as the model building data set. The designers believe the data in these four years would be sufficient to cover the major features of the study domain. The 1996-1997 data were reserved as an additional production set to test the capacity of the model to predict the annual cyclical feature of raw water colour, the reliability, and adequacy of the model.

Several simplifications of the ANN model architecture were achieved to both accommodate the on-line application and to improve the model performance. Originally there were twelve input parameters (Zhang and Stanley, 1997). The upgrade eliminated one turbidity time series input to reduce the chance for potential lagged prediction. The upgrade also eliminated the colour magnitude index to improve the applicability of the model. Not needing to estimate this system index greatly reduced the subjective content in the model and the chance of mistake in estimation. Finally, the models were designed to accommodate the optional use of the upstream rainfall information. Since this information is not available all the time on a daily basis, it is desired that the neural network models have the capacity to predict with or without the rainfall upstream.

The most noticeable improvement in the upgrade process is a systematic approach to assess the data in the database, which is the pattern recognition approach. In this approach, a Kohonen network was used to automatically separate the database into cohesive groups. The basic concept employed by the Kohonen network is the Euclidean distance concept. Based on the closeness (Euclidean distance) of the data points in the n-dimensional input space (“n” is the number of inputs for the Kohonen network), data points are grouped together (Looney, 1997). Each group has its common feature(s) consistent within the group, and quite distinguishable among the groups. Further subdivision of each group by Kohonen network yields a series of sub groups, each subgroup contains more detailed features and the similarity among the data points within the subgroup is enhanced. With sufficient subdivision, the members of each subgroup would be very similar in features. Thus, equally spreading these members into the training, testing, and production data set can form data sets that well represent the study domain. This approach also minimizes the chance of missing coverage to some sections of the domain, and the problem of under-training or over-training of the neural network model.

Once the candidate models are built, the upgrade assesses them with a multiple evaluation technique. Error assessments such as R-squared and root mean square error are no longer the only criteria. Statistical measurements such as the test of the goodness of fit and residual analysis are necessary. Furthermore, sensitivity, feature-matching ability, and applicability have become key issues in determining the best model. For example, sensitivity is the response of the model reacting to the changes in the input values. The more sensibly and logically the model reacts to the inputs, the better the model. The model should not only be sensitive and dynamic to the key inputs, but it should also react logically to changes in the other parameters. Some out of bound conditions could be tested if needed. Feature-matching ability assesses whether the model matches the distinct graphical features of the system, such as the peaks and valleys.

The final design of the upgraded system employs a multi-model combination approach. Since the final product of the model is a prediction, if the prediction only provides a single value, it is very unlikely the actual value will be an exact match to the prediction most of the time. A more logical approach is to predict a range of possible values. In other words, the predictions will provide minimum, maximum and a most likely value in between. Thus, when designing the models, each model must be tuned to its specific purpose. For example, the model to predict the daily maximums will be very sensitive to the inputs. Any change in the major inputs will result in a noticeable change in the prediction. In terms of evaluation criteria, the R-squared and residual analysis will not be favourable for this kind of model, but the graphical examination of the peak generation and initial colour jump detection will be more accurate. On the other hand, the minimum value model is more conservative in prediction. It does not predict the change of the trend well but follows the trend well instead. The most likely value is predicted by the model whose performance is the best overall in all measuring criteria. The final system is presented in Figure 5-2.

The system is subdivided into two prediction categories: predicting with or without rainfall information upstream. Within each category, the prediction is also divided into two types. Type 1 models predict the whole colour value. This type of model predicts with less dependency on the present colour value and is designed to be more sensitive to a change in weather information. In an overall picture, the predictions from type 1 models tend to fluctuate on top of the trend. Although the performance of the type 1 model may not be excellent on statistical evaluation, it is satisfactory in terms of predicting the colour jumps and colour peak well. The overall prediction tendency for this type of model is aggressive (over-prediction). Thus, the prediction by this type of model is considered as the possible maximum colour value. Type 2 models predict the colour change from today. It uses today's colour value as the base and adds the predicted difference on top of it. This type of prediction relies more on the present value and thus, tends to be conservative and follow the trend well. Within each type of prediction, there are also two models, each specialized in predicting certain conditions. Their predictions complement each other, and the final prediction value is the average of the two.

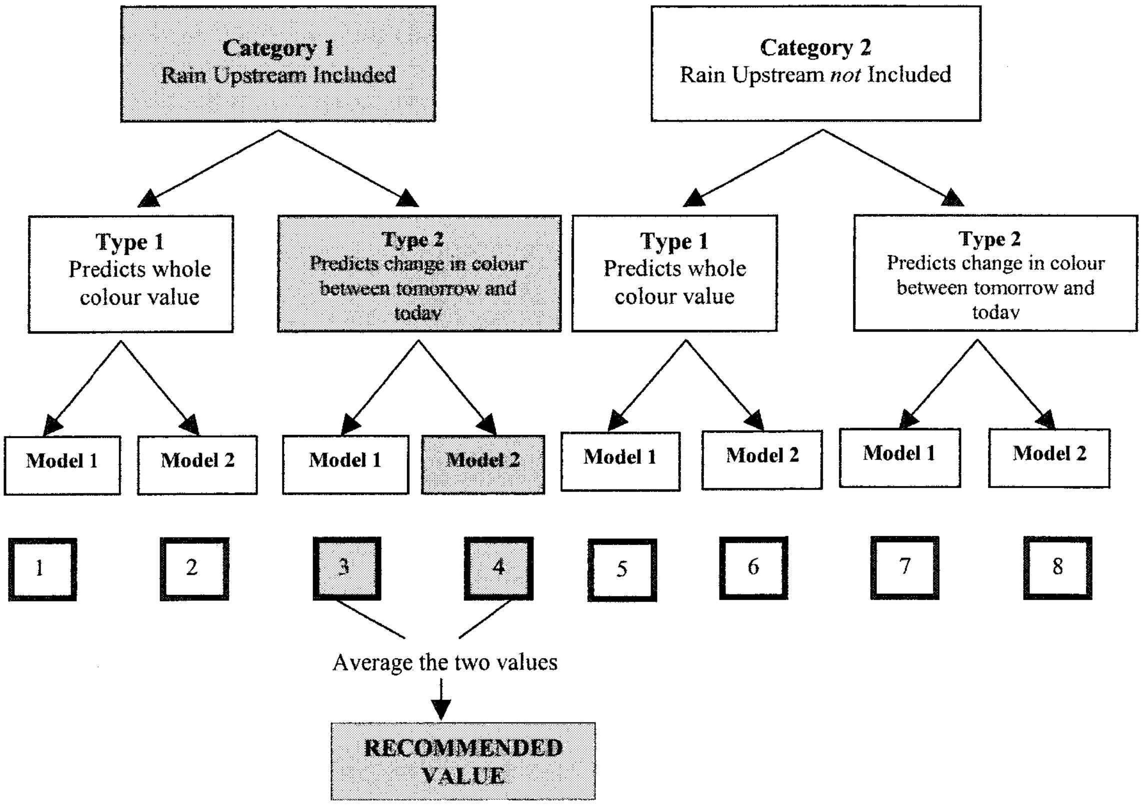
Category 1
Rain Upstream Included
Category 2
Rain Upstream *not* Included
Type 1
Predicts whole colour value
Type 2
Predicts change in colour between tomorrow and todav
Type 1
Predicts whole colour value
Type 2
Predicts change in colour between tomorrow and todav
Model 1
Model 2
Model 1
Model 2
Model 1
Model 2
Model 1
Model 2
1
2
3
4
5
6
7
8
Average the two values
RECOMMENDED VALUE

Figure 5-2 The Structure of the Final Prediction System

It was found that among the four types of predictions, the best overall model is the type 2 model in Category 1. Obviously, the additional rainfall information upstream helps the prediction. Its performance on the production set is presented in Figure 5-3. The R-squared for the prediction is 0.937 and the root mean square error is 0.18 TCU. The predictions catch spring colour jump, match the peaks, and also follow the trend well. However, the predictions have a tendency to over-predict some late summer medium size colour peaks.

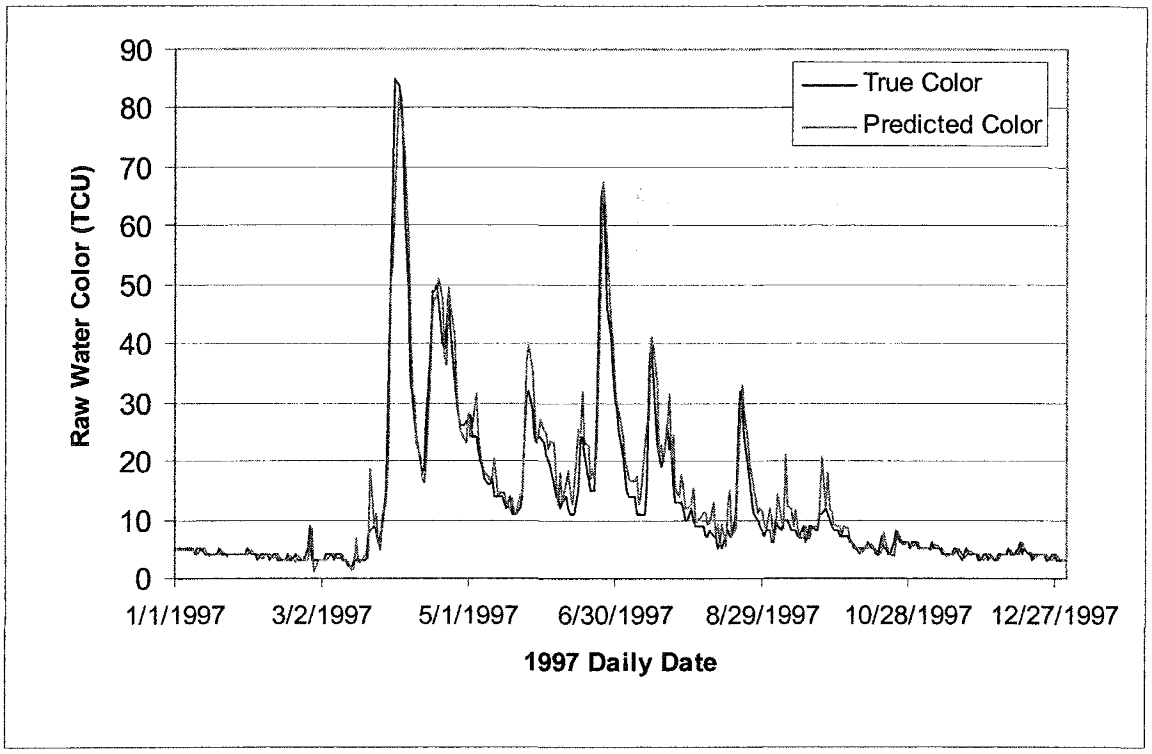

Figure 5-3 The Predictions on the Production Data Set (1997)

On-line Implementation

The final verification of the system is to implement and test it in the Rossdale WTP. On-line implementation of the raw water colour prediction model was accomplished by first producing runtime versions of the neural network models and then integrating them into the plant's supervisory control and data acquisition (SCADA) systems. The runtime versions of the ANN models can be expressed as either C program module, Visual Basic program module, or Excel spreadsheet file. All the necessary model inputs were made directly available from the SCADA system in real-time to the computer which executes the neural network system.

A number of programs and interfaces were built to retrieve input data and to execute the raw water colour prediction model on a real-time basis. Some input data originated from instruments located at the Rossdale Water plant (colour and turbidity). The rest of the data are from the Government of Alberta, Department of Environmental Protection databases (ambient temperature, precipitation, and river flowrate). To retrieve data from Alberta Environmental, a system was set up to automatically dial-up and attach to their database on an hourly basis and transfer the required values to the plant SCADA system.

Once all the required input values had been retrieved and stored on the SCADA system on a continuous basis, a two-way link was set up between the SCADA system and the computer which executes the neural network system. An interface using Visual Basic and Microsoft Excel was used to receive input data from and to pass the predicted raw water colour values back to the SCADA system. The predicted values of colour were then displayed graphically on operator stations for viewing. Figure 5-4 displays the architecture of information flow for the on-line raw water colour forecast model.

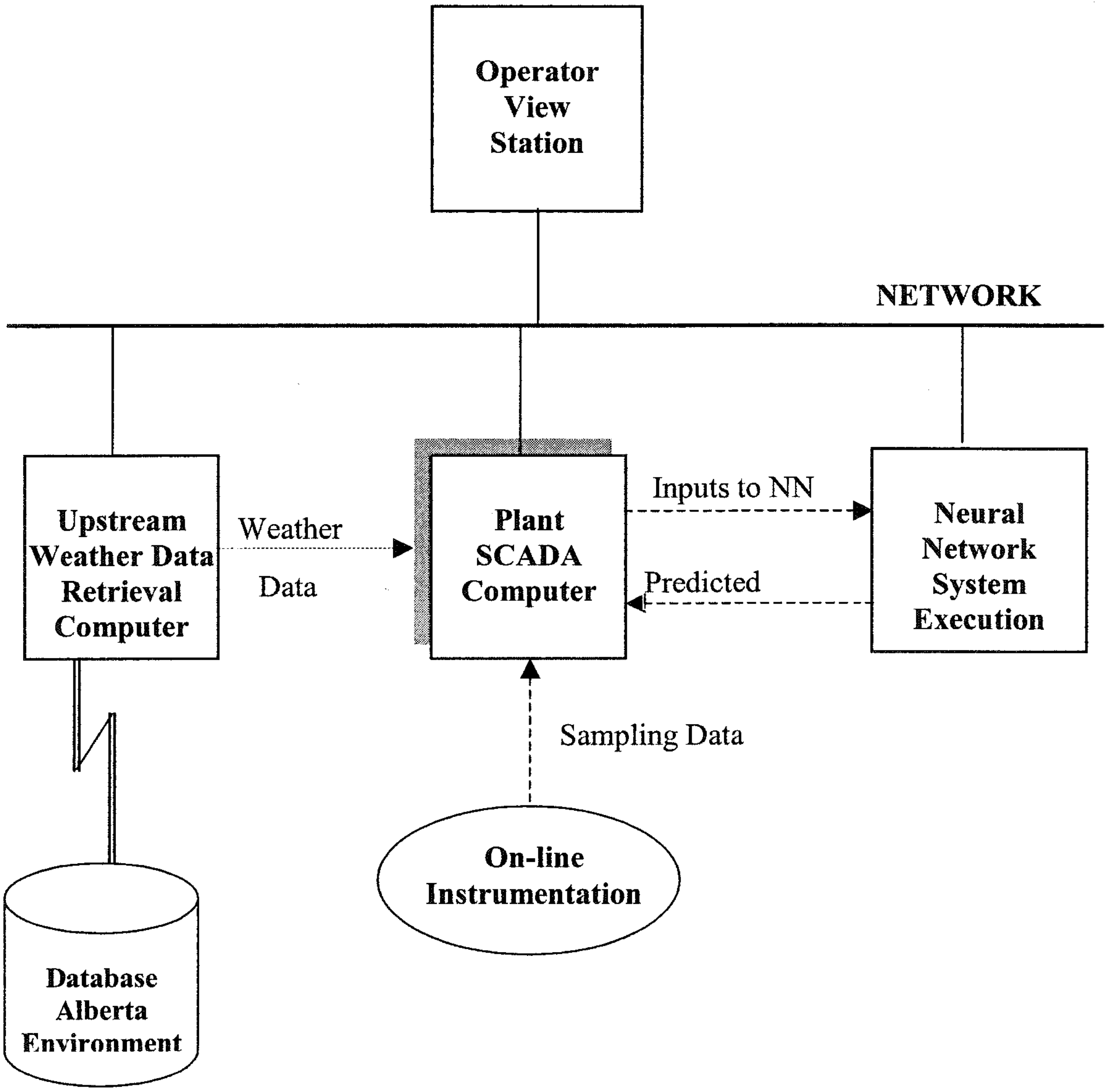

Figure 5-4 Information Flow for On-line Raw Water Colour Prediction

Although the on-line raw water colour prediction system is capable of executing the neural network prediction system continuously, the actual frequency of update depends on the availability of new information. This frequency varied throughout the testing period due to problems with receiving information in a timely fashion from the databases at Alberta Environmental Protection. During certain periods of time, the information was not available at all and some gaps in the results were produced. When all systems were working well, the average update frequency of data was between 2 and 12 hours.

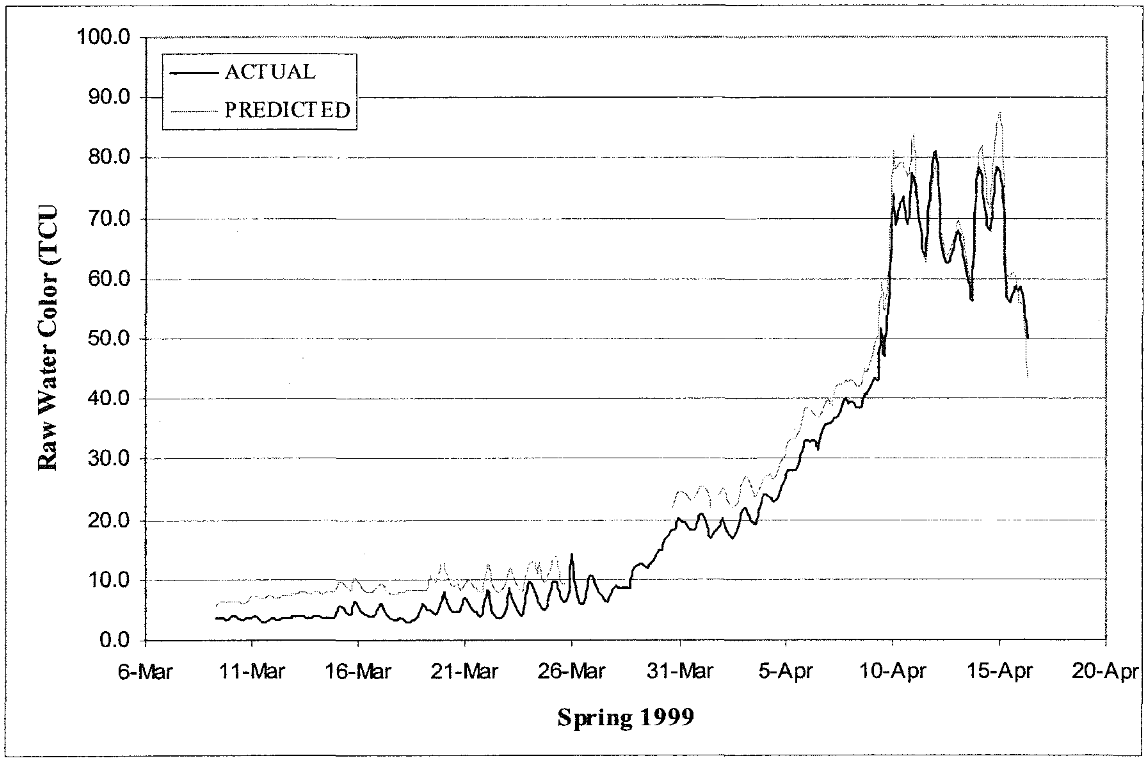

Figure 5-5 On-line Raw Water Colour prediction - Spring Runoff 1999

The results from the 1999 Spring Runoff period are displayed in Figure 5-5, where actual and the best predicted raw water colour trends are shown. The data in Figure 5-5 were generated every four hours whenever possible. There are two gaps in the predicted values in the early spring due to weather information being unavailable from the Alberta Environmental Protection databases at that time. The statistical results for the pre-spring raw water (actual colour less than 10 TCU), spring runoff (actual colour greater than 10 TCU), and overall early year are shown in Table 5-1.

Table 5-1 R-Squared Measurement for the On-line Prediction

Period	R^2	Average Error (TCU)
Pre-spring	-5.1	3.9
Spring Runoff	0.95	4.3
Overall Early Year	0.97	4.1

Discussion

By examining Table 5-1 and Figure 5-5, it is easy to see that the performance of the neural network system seems dramatically different between pre-spring and spring runoff. The R-squared for the pre-spring is negative while it is 0.95 for spring runoff. The reasons for this dramatic difference in performance are two fold. First, when designing the ANN models, there is an emphasis on predicting the high colour values (10 TCU or more) rather than the accuracy on the low colour values (less than 10 TCU). This is due to the PAC dosing strategy used in the Rossdale WTP as mentioned before. Higher prediction accuracy on the high value will help operators prepare the PAC usage for the next day. On the other hand, when actual colour is less than 10 TCU, PAC is not required. Lower prediction accuracy in this range is therefore not as critical for chemical dosing as when the colour value is high, but the operators still need the predictions to be accurate enough to know whether the next day colour will be higher or lower than 10 TCU.

Secondly, the performance in pre-spring should be evaluated by multiple criteria such as R-squared value and the graphical analysis. The performance is poor if one considers the R-squared value only. In Figure 5-5, it is obvious that the predictions tend to be twice the actual values. When calculating the R-squared, the differences between the actual and the predicted values are squared, and thus severely penalized. That is why the R-squared for the early spring is negative. However, by performing graphical analysis, the predicted values actually recognize all the small colour jumps and trend well. Thus, the overall performance is not as bad as indexed by the R-squared value only.

Nevertheless, the poor performance in the early spring did worsen the operators' opinion of the ANN prediction system. There were several false alarms of the spring color jump (when colour > 10 TCU) and the predictions tended to be twice the actual values. These factors resulted in operators losing confidence in the ANN prediction system. Some of them were willing to give the system a second try, but the missing weather data in late March made it impossible. Eventually, the operators gave up on the system until the plant control engineers showed them the ANN system performance in late spring.

Did the on-line system meet the utility's expectations? The best approach to this question is to compare the on-line system with the previous methods used by EPCOR Water Services. Previously, EPCOR had to rely on expert opinions and manned observations upstream. Some of the creeks in the upstream watershed have flow monitoring stations, but these are typically still frozen when runoff begins. The first phase of runoff involves melting of the snow on the ground, which flows overland into the still frozen creeks. This melt flows over the surface of the creek's ice and makes its way to the North Saskatchewan River, carrying organic material. This "first flush" is typically very high in colour. The creeks themselves begin to thaw later on, but it is the onset of snow melt that signifies the start of runoff and high colour.

Therefore, in the past, the only way to accurately identify if runoff was about to occur was to drive into the upstream watershed and visually check if the snow upstream had started melting. Three consecutive days with daily high temperatures of 8 degrees Celsius or greater provided fairly good confidence that runoff was about to begin, but this was only a guideline. Colder temperatures at night could delay the melting whereas one very warm day, even when not preceded by other warm days, could cause runoff to begin. Therefore, the best option was to make regular trips to the upstream creeks as soon as the weather began to warm up. As it is an approximately one-hour drive to the nearest significant creek upstream, this was a time-consuming task and because conditions could change in a few hours, it was still possible that the onset of melting would be missed. This problem was alleviated somewhat by asking local residents to watch the creek near their house, but if they were away from home the melt could again be missed. Identifying the onset of melting in the upstream creeks allowed EPCOR to pinpoint runoff to within a few hours but not the magnitude of the colour.

The neural network prediction system allows a fair warning of the initial colour jump without having to drive out to the watershed, and a good colour magnitude forecast during spring runoff. However, the operators did not have confidence in the system prediction, as the system hasn't accurately predicted the onset of runoff. Nevertheless, the ANN system does predict the magnitude of colour well once the raw water colour is in the high magnitude. Hence it can help with chemical dosing and planning.

Conclusions

This paper describes our first attempt to implement an ANN system on-line in the water treatment plant on a real time base. This transition from research to actual industrial application is a giant step but also a difficult one. Although the model upgrading process is fairly successful with many improved techniques applied in the process, the implementation stage did not yield the best result that the ANN system capable of.

Overall, a sound real world ANN application must begin with a practical design. Clear understanding and sufficient knowledge of the study domain is critical to define an achievable goal for the project. A clear defined goal in turn helps to identify and collect all available data that is representative of the study domain. Notice that the availability of the data impacts directly on the accuracy of the prediction. Eventually employ all the available techniques such as pattern recognition, architecture simplification, range prediction, and multi-model approach to build a reliable ANN system. Knowing when, what, and how to upgrade an ANN system enables comprehensive design to improve reliability and accuracy of the ANN model.

Although the implementation of the ANN system on-line was not as successful as we had hoped for, it has been a good experience. There were many details and lessons learnt from this project. First, a better system component is needed to predict the initial day of the spring colour jump. The false alarm of the current ANN system is a major issue with the operators.

Upon the final design evaluation, the designers believe that the current design to only forecast the colour value one day ahead may be too short-term for detecting the true starting point of spring run off. Thus, it is recommended to design the model to predict four days into the future. That is, the ANN models will have the same inputs but will now have four outputs, predicting the colour values one, two, three and four days ahead. It may also be possible to add some new information in the model inputs to assist the detection of the starting point of spring runoff. For example, incorporating the trend in daily air temperature as an input may improve the prediction, as it has been used in the past as an intuitive indicator by staff (three consecutive days over 8° C).

Second, better education and follow-up with the operators is required. Proper education helps the users obtain a precise understanding of system capability, and establishes a proper level of expectation. Full awareness makes for better use of the system.

Third, there are more details that can be added to the user interface to improve the user-friendly aspect. It will be beneficial to incorporate the system directly into an operation status display screen, so that the result is there with all the other operational data and will be looked at by the operators. Redesign of the interface to a form of mouse click and display could also encourage the use of the system by the operators.

Fourth, follow up of the implementation must be maintained. Resolving all the issues during the use of the system as quickly as possible will improve the ownership of the system. More use of the system would build familiarity and comfort.

Since the completion of this project, there are more recent advances of the ANN modeling technology in our research group. A combination of the pattern recognition approach with ANN will create a semi-black box type neural network model. The designers now have a lot of controls on how to train the network for specific purposes and are able to interpret many components of the finished ANN model. It is our intention to further upgrade the model to this newest ANN technology.

EPCOR Water Services is aware of the great potential of the ANN technology, and has committed to implement more ANN systems into their actual treatment processes. The lessons learnt from this project will benefit all the subsequent applications.

Acknowledgements

This chapter is a revised version of a paper, which I co-authored with Audrey Cudrak, Riyaz Sheriff, and Steve Stanley. Ms. Cudrak, Mr. Sheriff, and Dr. Stanley, all from EPCOR Water Services, provided technical assistance in the area of water treatment process and the SCADA system.

I would also like to thank Dave Rector and Steve Hrudey from EPCOR Water Services for their assistance with the on-line implementation and wonderful support of the project.

References

Boger, Z. (1997), Experience in Industrial Plant Model Development Using Large Scale Artificial Neural Networks. *Information Sciences* 101(3-4): 203-216.

Cote, M., Grandjean, B. P. A., Lessard, P., and Thibault, J. (1994), Dynamic Modeling of the Activated Sludge process: Improving Prediction Using Neural Networks. *Wat. Res.* 29(4): 995-1004.

Daniell, T. M. (1991), *Neural Networks - Applications in Hydrology and Water Resources Engineering.* International Hydrology and Water Resources Symposium, Perth.

Flood, I. and Kartam, N. (1997), Neural Networks in Civil Engineering I : Principles and Understanding. *Journal of Computing in Civil Engineering* 8(2): 131-163.

Garrett, J. H., Ranjithan, S., and Eheart, J. W. (1992), Application of Neural Network to Groundwater Remediation. *Expert Systems for Civil Engineers: Knowledge Representation.* R. H. Allen. New York: American Society of Civil Engineering: 259-269.

Hasham, F. A., Stanley, S. J., and Kindzierski, W. B. (1998), *Modelling of Urban Air Pollution in the Edmonton Strathcona Industrial Area Using Artificial Neural Networks.*

Transportation, Land-Use, and Air Quality: Making the Connection, Portland, OR: ASCE.

Looney, C. G. (1997), Pattern Recognition Using Neural Networks: Theory and Algorithms for Engineers and Scientists. Oxford University Press, 198 Madison Avenue, New York, New York, pp. 458.

Shukla, M. B., Kok, R., Prasher, S. O., Clark, G., and Lacroix, R. (1996), Use of Artificial Neural Networks in Transient Drainage Design. *Transactions of the ASCE* 39(1): 119-124.

Zhang, Q. and Stanley, S. J. (1999), Automating Water Treatment Plants by the Artificial Neural Network Approach, *Proceedings of the American Water Works Association 1999 Engineering and Construction Conference*, Orlando, Florida.

Zhang, Q. and Stanley, S.J. (1998), Use of neural networks for process modeling and control of coagulation. *Journal of Environmental Engineering, American Society of Civil Engineering, vol. 125(2): 153-160.*

Zhang, Q., and Stanley, S.J. (1997), Forecasting raw-water quality parameters for the North Saskatchewan River by neural network modeling. *Water Research* 31(9): 2340-50.

Chapter VI Artificial Neural Network Modeling of Oil Sands Extraction Processes

Introduction

The Athabasca oil sands of northern Alberta are contributing an increasing proportion of Canada's oil production. Within three years, with the expansion of the two existing commercial operations and the opening of a third, extraction of surface-mined oil sands will produce over 500,000 barrels of oil per day and will recover over 90% of the oil in place.

While these figures attest to the successful engineering of extraction processes, the physical and chemical mechanisms of oil sands extraction are not understood completely. In its simplest form, the extraction process is comprised of a stage where the oil sand is mixed with water to separate the bitumen from the sand. The resulting slurry is diluted further and pumped into a series of flotation vessels in which the bitumen is recovered. However, there is a complex interaction involving hydrodynamics, residence time, process temperature, water chemistry and other oil sands characteristics that determines the process efficiency.

Attempts to develop an understanding of the process by examining one factor at a time have yielded inconsistent and inconclusive results. The use of factorial designs, while providing one means of identifying interactions among variables, is ill-suited to accommodate the myriad of input variables that could affect the process. Furthermore, even with well-designed sets of experiments on lab-scale units using process waters, oil sands, chemical addition strategies and temperatures identical to those encountered in a continuous commercial process, the results could not be guaranteed to simulate commercial results on account of differences in hydrodynamics.

Over a twenty-year period, the Alberta Research Council (ARC) has generated a database containing input and output variables from thirty individual projects, comprising over 2200 tests on lab-scale units of various designs. Each test was characterized by as many as 45 input and output variables, including:

- oil sands characteristics, including oil content, particle size, soluble salts and geological environment;
- extraction unit design, through a qualitative descriptor;
- process temperature, water to ore ratio, water chemistry and chemical process aids;
- residence time;
- fraction of feed bitumen recovered in primary and secondary flotation stages; and,
- amounts of water and solids recovered with the bitumen.

For the reasons cited above, conclusions that could be drawn from any single subset of the database were not always consistent. Given that a high level of care and control was exerted over each test, it appeared that the inability to reach consistent conclusions about the process mechanisms should be attributed to the high level of interaction among the variables. All previous efforts to identify these interactions, and therefore be able to predict process outputs given known process inputs, have failed.

Under these circumstances, it seemed worthwhile to try using an artificial neural network (ANN) approach to examine ARC's database of oil sands extraction experiments. This approach is a versatile modeling technique for complex nonlinear phenomena. It does not require a physical model to represent a process. Instead, it focuses on building a cause-and-effect mapping between the parameters having an impact on the process (input parameters) and the process outputs.

Although the ANN approach has been used in various disciplines of engineering since the 1980's (Flood and Kartam, 1997), its use in the oil sands extraction industry is quite new. The use of neural networks first appeared in other branches of the petroleum industry in the early 1990's. Initial applications included the analysis of formation lithology from well logs, and the selection of a reservoir model for conventional well test interpretation studies (Mohaghegh, 1995). By the late 1990's, neural networks had been used for seismic pattern recognition (Balch, *et al*, 1999, and Reymond, *et al*, 1999), permeability predictions (Anxionnaz, *et al*, 1999, and Olson, 1998), drill bit diagnosis (Arehart, 1989 and Dashevskiy, *et al*, 1999), and analysis and improvement of oil and gas production (Murray *et al*, 1999, and Shelley, 1999). To the best of the author's knowledge, this chapter describes the first application of ANN modeling to oil sands extraction.

The chapter is organized in the following way. First, a brief introduction of the ANN concept and a simplified training protocol for ANN models is presented. Building on these basic concepts, an advanced ANN modeling technique called pattern recognition is described. Finally, the results of applying ANN modeling to the oil sands extraction experiments and the associated sensitivity studies based on the ANN models are discussed.

Background to Artificial Neural Network Models

A Simple Artificial Neural Network

As its name implies, an artificial neural network is a modeling technique that simulates the human brain's problem solving process. Just as humans apply knowledge gained from past experience to new problems or situations, a neural network takes previously-solved examples, looks for patterns in the examples, learns the patterns, and develops the ability to correctly classify new patterns. In this way, the network internally organizes itself to learn the presented concepts. It does not require programming, logical inference schemes, or statistical algorithms to develop a solution. However, it does require expert knowledge of the parameters which govern the process (Zhang and Stanley, 1997).

An ANN is a highly interconnected network of many simple processors (Figure 6-1). These simple processors, called artificial neurons, are organized into an input layer, a hidden layer (or layers) and an output layer. The neurons are connected by weights. The input layer takes a pattern of stimulation from the outside world and passes the pattern to the hidden layer(s) where the pattern is processed. Finally, an output pattern is generated and presented by the output layer.

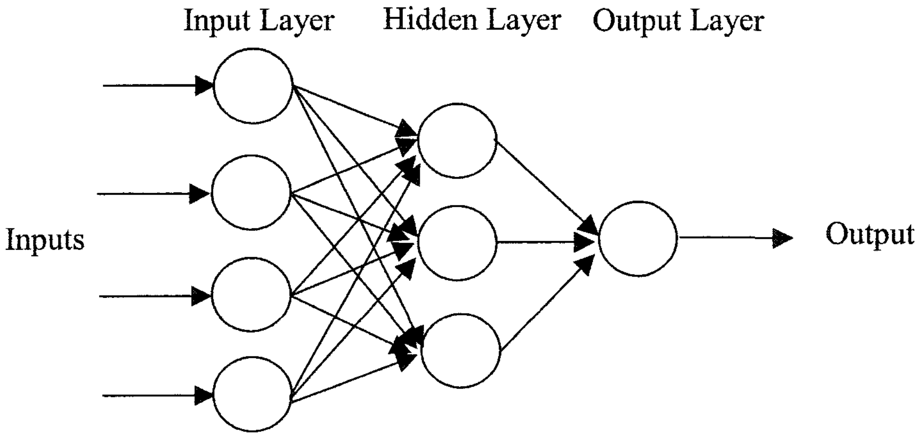

Figure 6-1: A Simple Neural Network

There are many algorithms that can be used to teach a neural network to learn real world data. The ANN models built for this study utilize the back-propagation learning algorithm. The detailed information for the back-propagation algorithm can be found in a paper by P. K. Simpson (1991) or in the on-line neural network textbook at Http://www.statsoft.com/textbook/stneunet.html. A back-propagation neural network learns by adjusting the interconnection weights between layers. The answers that the network produces are repeatedly compared with the correct answers, and each time the connecting weights are adjusted slightly in the direction of the correct answers. Eventually, if the problem can be learned, a stable set of weights adaptively evolves and will produce good answers for all of the sample decisions or predictions.

Successful ANN applications have some typical characteristics (Bowen, 1991). First, the algorithm to solve the problem is unknown or expensive to discover. Second, heuristics or rules to solve the problem are unknown or perhaps difficult to enunciate. Third, the application is data intensive and a variety of data describing the subjects is available. On the other hand, the ANN approach will not be suitable if adequate representative data for the processes is not available for modeling. ANN models are somewhat black box in nature, so they are not suitable for situations in which transparent computational procedures must be employed.

Advantages of ANN Modeling for Oil Sands Extraction Processes

There are a number of distinct advantages in using an ANN approach to modeling oil sands extraction processes, versus conventional modeling approaches. First, a mathematical algorithm is not necessary for building an ANN model. Constructing a conventional mechanistic model of oil sands extraction is extremely difficult, because of the uncertainty involved in scaling up a representation of the physical and chemical mechanisms underlying the process from a microscopic level to a macroscopic one. Conventional modeling requires mathematical algorithms to describe this uncertainty, whereas a neural network simply learns the process based on the information contained in the set of extraction data. However, ANN modeling does require a reasonable knowledge of the important factors governing the process.

Second, the ANN modeling approach is fast and flexible. With the pattern recognition modeling approach, a complicated scenario can be divided into several well-defined components and then modeled more easily and directly for each component. This modularization of the study domain allows quick removal or addition of new modules when necessary. For example, if there is a change in the oil sands extraction process, the related module simulating that part of the process can be adjusted without altering the other modules to fit the new process dynamic through model retraining.

Third, ANN models can handle nonlinear relationships well due to their inherent nonlinear data structure and computation process. This capability of modeling nonlinear relationships makes neural network modeling well suited for oil sands extraction modeling and evaluation.

Fourth, neural networks tend to be inherently fault-tolerant, since their data structure is loosely organized and there is no limit on the range of the input parameters. Depending on the significance of the parameters in the neural network architecture, the impact of any false input can be either blocked within the data structure or reduced in magnitude through the non-linear calculations inside the data structure. This fault-tolerant feature is a critical one for situations involving real-time process control.

General ANN Training and Evaluation Protocol

The actual process to develop a successful network is quite complex. Typically, the process is divided into four major phases: source data analysis, system priming, system fine-tuning, and model evaluation.

The major objective of source data analysis is to understand the source data. At this stage, several aspects of the source data are examined. Typically, a correlation table is produced to evaluate the correlation among the selected parameters. However, the correlations illustrated in the correlation table do not include any potential multi-parameter interactions. If time series parameters are included, there is also a need to examine the autocorrelation for the time series parameters to assess the time link among them. The other typical task of the source data analysis is to sort through the records to find missing data fields and mistaken records.

The main task of system priming is to select a set of input parameters and a proper architecture for the ANN model. Usually two kinds of input parameters are included in the design of the ANN model: the cause-effect parameters and the system indexes. The cause-effect parameters provide the clues to identify the potential results. The system indexes describe the distinctive features of the particular system to be modeled and provide additional constraints or conditions to improve the accuracy of the model further. The system index is unique to each domain and must be specially designed to fit the study domain. Sometimes a third type of input, the time series parameter, is also used to improve the prediction accuracy. For example, the output at the previous time interval can be used as an input to predict the output at the next time interval.

The goal of system fine-tuning is to split the representative data for the process into a training set, a testing set, and a production set to train, test and evaluate respectively the ANN models. The training set is the data set that the model will learn from. The learning is performed in a looping style. At the end of each learning cycle, the success of the ANN model at that stage is measured by its performance on the testing set. If the performance on the testing set meets certain criteria, the learning process will be terminated. This model will then be evaluated on the production set. In the evaluation process, the key issue is whether the model has just memorized the cases it has seen or has actually mastered the mechanisms of the process. Since the training set and the testing set are actively involved in building the ANN model, good performances on both data sets do not guarantee the model will perform reasonably on a data set it has not seen before. Thus, a stand-alone data set, the production set, is used to test the ANN model's ability to predict data new to it. Satisfactory performance on the production set provides reassurance as to the quality of the ANN model.

Model evaluation includes two aspects: the error assessment within the model, and the error assessment across the models. The purpose of the error assessment within the model is to evaluate how closely the neural network model can recognize the patterns presented in the training data set and its ability to reproduce them. The error assessment across the models is more complicated. Although sets of minimum training errors and testing errors are available for the candidate models, they do not represent every aspect of the models' behaviour because they are average values calculated over the whole data set. Thus, multiple evaluation criteria are needed. Sensitivity, feature-matching ability and applicability have become key issues in determining the best model.

Sensitivity is the response of the model reacting to the changes in the input values. The more sensible and logical the model's reaction to the inputs, the better the model is. Not only should the model be sensible and dynamic to the key inputs, but it should also react properly to changes in the other parameters. Sometimes out of bounds conditions can be tested, if necessary. Feature-matching ability assesses whether the model matches the distinct features of the system. In graphical terms, it examines the generation of peaks and lags in performance. For applications oriented to process control, the model is selected to fit the system characteristics, such as a dynamic system or a stable system.

ANN Modeling with a Pattern Recognition Approach

Pattern recognition is a process simulating how humans learn and identify objects in their world. In a human recognition process, there are three major steps. Human recognition starts with observation. Then an individual tries to make sense of what is observed. Finally, based on the understanding obtained from the observation, an individual makes judgements on further observations.

Since it mimics the human recognition process, a pattern recognition process consists of three steps to create a system to describe a study domain. The first step is to collect patterns that describe the study domain. Next, these patterns must be separated into groups, in which each group presents a distinct concept. Based on these distinguishable concepts, a system will be created to describe each concept and collectively the whole study domain. Third, further patterns collected from the study domain are recognized by the system either as a member of the existing components of the system or as part of a new component that must be learned to complement the system's understanding of the study domain.

The key component of this pattern recognition modeling is the second step, which in turn can be divided into three processes: separating the patterns into groups representing distinct concepts; building models capable of describing each concept; and, uniting the models together to form a system. The process to separate patterns into groups is called clustering. Within each cluster, member patterns are closer to each other and their own centers than to the centers of other groups. Based on the clustering, models are built to describe each cluster. This process is called modularization. The modularization breaks down a single, complicated study domain into a number of much more clearly-defined sub-domains. Modeling within these well-defined sub-domains is expected to be more accurate than modeling the whole domain at once. The modularization also allows the system be modified more easily to accommodate changes in the study domain. Modules within the system can be upgraded or added easily to describe the new system instead of scrapping the whole system and starting from the beginning again. A cautionary note regarding the modularization approach is that the number of modules should be sufficient to span the whole domain.

After the models for each module have been completed, they are organized into a hierarchical structure to form a pattern recognition system (Figure 6-2). It is a logical step, since the study domain has already been clustered and modularized. However, simple accumulation of the modules does not necessarily provide a better result than a solution without any modularization. A data filter must be designed to dynamically link the modules together, channel the data into the right module so that it can be analyzed properly, and finally present the results with a high degree of accuracy. The data filter should also have the capability of recognizing new entries that do not belong to any of the existing modules, and then alert the user with a warning signal. Corrective action would have to be taken to treat these new entries.

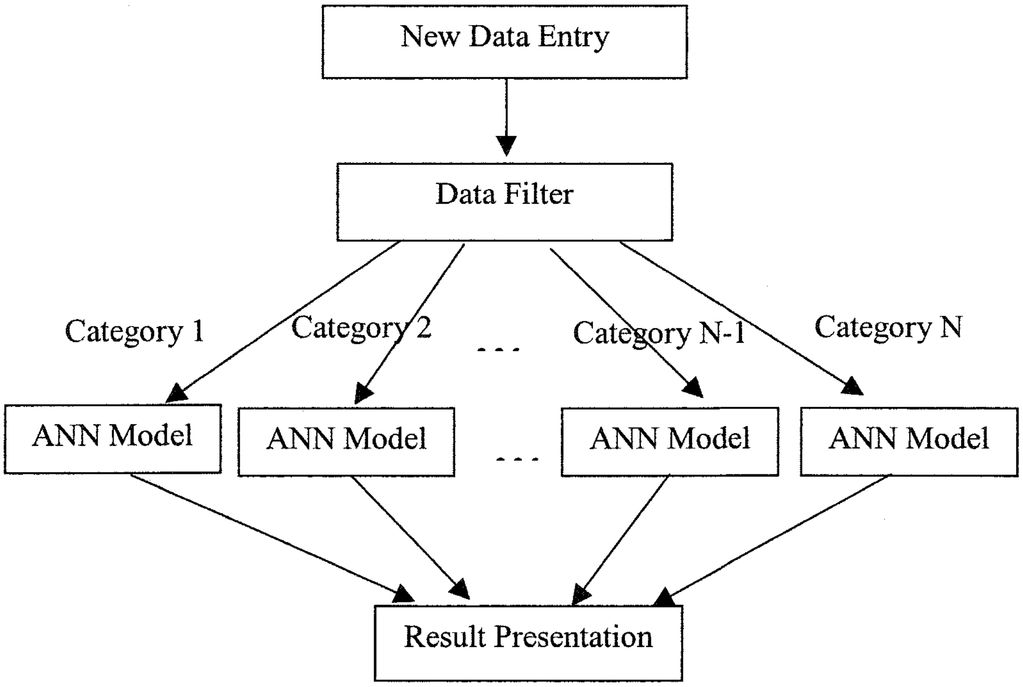

Figure 6-2: Hierarchical Model Structure for a Pattern Recognition System

The design of the data filter is based on the concept of Euclidean distance, and takes advantage of the fact that the Euclidean distances between members of the cluster and the cluster center are distributed normally (Zhang and Stanley, 2000). As a new data entry is entered into the system, the data filter calculates the Euclidean distance of the new entry to all existing cluster centers. The cluster with the minimum distance to the new entry is identified. Then the data filter examines whether the Euclidean distance between the new entry and the cluster center falls within certain requirements that would allow it to become a member of the cluster. For example, a condition for membership in a cluster could be that the distance from the prospective member to the cluster center must be smaller than 2.5 times the standard deviation for the cluster. If this condition were met, then the data entry would be channeled into the modeling module responsible for that cluster, and appropriate actions could be undertaken within the module. If the condition were not satisfied, then the new entry would be considered to be a member of a previously unidentified category. A flag should be set before admitting the new entry into the database. No further analysis of the new entry could be done at this time. If a sufficient number of new entries of this sort were encountered, then a new category with a corresponding modeling module would have to be created to accommodate them.

ANN Modeling of Oil Sands Extraction Experiments

Source Data Analysis

As discussed previously in the description of the ANN modeling protocol, the first step in the modeling is the source data analysis. There are several issues to consider at this stage. First of all, the records in the database must be examined to determine the integrity of the database. The items to tackle are the records with missing data fields and the records that are potentially in error.

Once the database has been cleaned up, the next step is to examine the correlation between input parameters. The purpose of this task is to identify the independent parameters and the dependent parameters, and to assess the degree of correlation between the dependent parameters. If several parameters are highly correlated, then only one or two of them should be used in the modeling, since too much redundant information in the input generates noisy scenarios for the neural network model to learn. However, in some cases, a limited amount of extra information helps the neural network. Determining how much redundancy is necessary is an art and varies from case to case.

Since the neural network can only handle numerical information, it is necessary to convert non-numerical data into an appropriate numerical representation. The range and magnitude of the numerical codes used to represent the data, as well as the relative location of the coded values within the range, do have an impact on the neural network modeling. For example, depending on the selection of the activation function inside the neurons, the neural network model can be very sensitive to the location of the coded data values within the overall range. If an S-shaped activation function is used, the network would be much more sensitive to coded values in the middle of the range than to those at either end of the range. The general rule in coding an input parameter is to incorporate as much knowledge as possible of the input parameter into the coding design.

The correlation analysis of the working extraction database for the study separated the parameters into two types: the parameters that affect or describe the oil sands extraction process (input parameters), and the parameters that report the process results (output parameters). It was found that, overall, the correlation among the input parameters was limited. There were a few exceptions that should be noted. As a result of operational procedures, the flotation time for the primary flotation stage was correlated highly to the scavenging time for the secondary flotation stage, with a correlation coefficient of 0.95. The bitumen content in the oil sand was correlated inversely to the water content in the oil sand, as should be expected, with a correlation coefficient of –0.85. For some reason, the type of mixing unit had a high correlation with the addition of air during the initial stage where the oil sand is mixed with water. It is likely that a consistent operational decision was made on whether to add air, depending on the type of mixing unit used in the experiment. Finally, the sodium content in the process water was correlated highly to the potassium content in the process water, with a coefficient of 0.82.

The output parameters of principal interest in the study involved bitumen recovery during the primary flotation stage. It was found that the fraction of feed bitumen recovered in the primary flotation stage (primary recovery) was correlated only slightly positively with the bitumen content in the oil sand and only slightly negatively with the water content in the oil sand. It was also interesting to find that the statistics for total bitumen recovery (recovery during the primary flotation stage plus recovery during the secondary flotation stage) were correlated highly with the statistics for primary recovery. Since the correlation between individual input parameters and individual output parameters was very limited, it can be concluded that most of the recorded input parameters could contribute to the extraction output.

It is important to note that the interactions of the input parameters can be more important to the output than the impact of any single input parameter. These interactions cannot be examined by the correlation study, but they can be examined by methods such as factorial design or a sensitivity study after the neural network system is built.

There were three input parameters in the original extraction database that were not represented by numerical values: the geological environment of the oil sand; the type of mixing unit; and, the addition of air during the initial stage when the oil sand is mixed with water. Air addition had only two values, on and off, so these two values could be replaced simply with 1 and 0. The coding design for the type of mixing unit was slightly more complex. There were three unit types in the working database, labeled as units A, B and C. Units B and C were similar in their hydrodynamic behaviour, but quite different from the hydrodynamic behaviour of unit A. Unit C was related a bit more closely than Unit B to Unit A. As a result, it was decided to assign a one-dimensional index to the type of mixing unit: Unit A was assigned a value of 1; unit C was assigned a value of 4; and, unit B was assigned a value of 5.

The coding design for the geological environment of the oil sand required another level of complexity. There were six types of environment recorded in the database. Three of them were the major environment types: continental, marine and estuarine. The other three were combinations of the three major types. As a result, the relationships among the six types of environment were non-linear. A conventional one-dimensional coding design would not be able to function as well as the two-dimensional coding design illustrated in Figure 6-3.

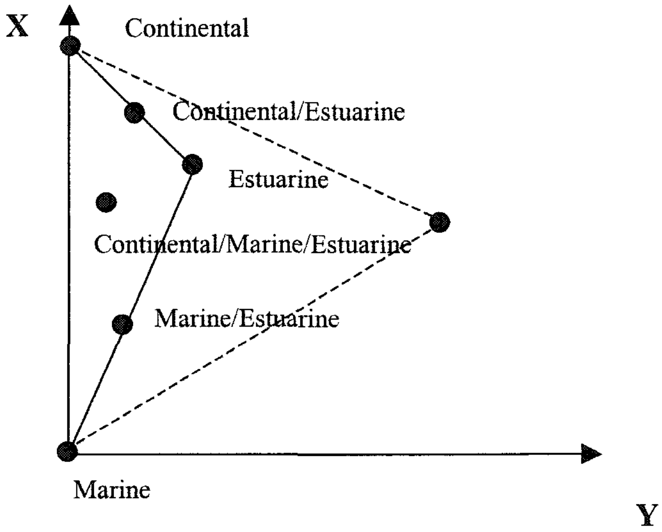

Figure 6-3: Triangular Relationship for the Geological Environment Parameter

In Figure 6-3, the coding design recognizes the importance of the three major geological environment types and uses them as the poles in a triangular relationship. The design also incorporates a subtle relationship among the three major environment types into the diagram. If the three major environment types were independent of each other, they would form an equilateral triangle. However, they are not completely independent of each other. Although the continental environment is quite distinct from the marine environment, the estuarine environment is in somewhat of a transition between continental and marine. Thus, the estuarine environment is related to both the continental and marine, but somewhat more closely to the continental. To reflect this relationship, the pole of the estuarine environment was moved inside the equilateral triangle (the dashed lines) in Figure 6-3 and closer to the continental pole. Within this triangle, the locations of the other three mixed-environment types were defined according to their degree of mixing with the three major types.

Table 6-1: Coding Design for the Geological Environment Parameter

Environment	**X**	**Y**
Continental	1	0
Marine	0	0
Estuarine	0.8	0.8
C/E	0.9	0.4
M/E	0.4	0.4
C/M/E	0.6	0.27

Once the relative locations of the six geological environment types were set, the next step was to code their locations. The continental and marine poles were assigned the positions (1,0) and (0,0) respectively. The projection of the estuarine pole on the edge between the continental and marine poles was designated to be (0.8,0). The height of the estuarine pole turned out to be irrelevant; in the pattern recognition approach the pair of index values are scaled separately. An angle of 45 degrees between the marine-continental edge and the marine-estuarine edge was used to calculate the coordinates presented in Table 6-1.

Preliminary Design of the ANN Models

There were two major tasks at this stage. The first task was to find the appropriate input parameters for the neural network models. The second task was to use the selected input parameters to divide the working extraction database into smaller and more manageable data groups. In other words, the source database had to be categorized into smaller groups possessing distinct similarities.

As was mentioned above, the output parameters of principal interest in the study are related to bitumen recovery during the primary flotation stage. It was decided that the two key performance parameters for the primary flotation stage should be: the fraction of feed bitumen recovered (primary recovery); and, the fraction of bitumen contained in the recovered froth (primary froth quality). These two output parameters were modeled separately.

The next step was to search for the key input parameters that had the greatest impact on the output parameters in the primary flotation stage. At this point, there were twenty-six parameters in the candidate pool. Eleven of them described the oil sand samples, while the remainder described the extraction process. The selection process was divided into two phases. First, an initial screening was performed to remove parameters that contributed little to the output parameters. Parameters that had too many missing data fields were also removed with discretion during this phase. If these parameters were selected as input parameters, it would result in fewer complete records available for the modeling process. In the second phase, the remaining parameters underwent a parameter interaction test to determine whether they were part of multiple interactions that might exist among the parameters. Parameters that showed little interaction with the other input parameters and the selected output parameters were dropped.

When the eleven parameters describing the oil sand samples were examined, six parameters were found to have a lot of missing data. These six parameters were the sodium, potassium, magnesium and calcium in the oil sand, the solid content of the oil sand, and the average particle size for the oil sand samples. The geological facies parameter had fifty-two discrete values and was determined to be an ineffective index for the ANN modeling technique. The water content in the oil sand samples was excluded because it was correlated highly to the bitumen content in the samples. These eight parameters were dropped from the pool. There were three parameters left to describe the oil sand samples; they were the bitumen content, the fines content, and the geological environment. As discussed above, two indexes (an index X and an index Y) were used to describe the environment parameter. Therefore, in the end, four parameters were used to characterize the oil sands in the extraction experiments.

Among the fifteen process parameters in the input candidate pool, the four ion concentration parameters (sodium, potassium, magnesium and calcium in the process water) were excluded because they could be collapsed into a pair of parameters: the monovalent and divalent ion concentrations in the process water. This left eleven process parameters for further examination: the type of mixing unit, the water temperature during the mixing (digestion) stage, the time allowed for the digestion stage, the amount of NaOH added, the amount of low energy extraction (LEE) chemicals added, the water to oil ratio during digestion, the time allowed for the primary flotation stage, the water to oil ratio during primary flotation, the addition of air during the digestion stage, the monovalent ion concentration in the digestion water, and the divalent ion concentration in the digestion water. Further examination of these parameters occurred during the second stage of the parameter selection procedure.

The second stage of the parameter selection procedure fit naturally with the source data categorization aspect of the pattern recognition approach. In this study, the approach employed for categorizing the source data mimicked the decision-making process that would be followed by an oil sands extraction expert when faced with the task of processing an oil sand sample. An expert would first examine the physical characteristics of the oil sand sample. Based on these characteristics, an expert would decide how the ore should be processed for optimum recovery. Consequently, the source data categorization was undertaken as a two-stage process. In the first stage, the whole database was classified into several categories according to the characteristics of the oil sand. Each category represented a major type of oil sand. In the second stage, the data in each category was divided further into classes according to the way in which the oil sand was processed. Each class represented a major option for processing a type of ore.

There were two major benefits associated with this approach. First, this approach allowed further assessment of the parameters selected for the ANN models. If a parameter were not essential for the data categorization, it would be less likely to have a significant impact on the processing operations. Second, this approach not only allowed a rapid comparison of the effects of different processes on similar types of oil sands, but it also allowed a systematic search to be undertaken to determine if an optimal process setting across different types of oil sand existed.

In the first stage of categorization, the source data were classified into three categories by the oil sands characteristic parameters. Each category was dominated by one major type of geological environment. These were marine, continental, and estuarine. The mixed environment types were absorbed straightforwardly into one of these categories.

During this stage, the effects of the environment index coding design on the categorization were examined. Tests were performed to categorize the data separately with four parameters (bitumen content, fines content, environment index X and environment index Y) or three parameters (bitumen content, fines content and environment index X). The use of only the environment index X was equivalent to collapsing the two-dimensional representation (a triangle) of the geological environment parameter into a one-dimensional representation (a line). It was found that when two environment indexes were used for the categorization, the category definition was much sharper than with a categorization using only one environment index. Apparently, a two-dimensional coding design provided much more information on the relationships between the various geological environments than a one-dimensional coding design.

In the second stage of categorization, the data in each category were divided further into operations classes by the extraction process parameters. Each process parameter was assessed in the three categories to determine whether its impact was the same in each category. The study was done in two steps. First, each parameter was taken out of its category one at a time to see how the categorization fared without this parameter. Then, among those parameters with lesser significance, several parameters were taken out together to observe the effects on the categorization. This was done because it is possible that even though some parameters might have a negligible impact on the categorization on an individual basis, they could interact with each other and collectively present a significant impact.

It was found that the interactions among the significant parameters (the parameters that were important in the single parameter test) were strong, as was expected. The interactions among the weak parameters (the parameters that were less important in the single parameter test) also existed, but with less strength. The level of importance of some parameters appeared to be different in the three categories that divided the data according to geological environment. An example was the water to oil ratio during primary flotation. It was a weak parameter in both the marine and continental categories, but appeared to be a strong one in the estuarine category.

As a result of the second stage study, it was decided that the amount of LEE chemicals added should be dropped as an input parameter because it was insignificant most of the time. This was due to the inconsistent usage of LEE chemicals in all three categories. The use of LEE chemicals was minimal in the samples from the marine and estuarine categories. It was more frequent for the samples from the continental category. Essentially, the use of LEE chemicals was an afterthought in the extraction experiments. Thus, it was excluded in the final input parameter selection. The rest of the ten process parameters were selected as input parameters. It was decided to preserve the same input parameters for the ANN models in all three categories. This allowed for an easy comparison of the effectiveness of the extraction process across different types of oil sand samples.

Detailed ANN Model Design and Evaluation

Once the input parameters were selected, the next step was to assign the data properly into training, testing, and production data sets. The main issue in channeling data into these sets was that each of the three sets should span the study domain. To achieve this, further data classification was needed within each operations class. Classification at this level made pattern comparison possible in a small and clearly-defined group. Within each group, patterns with similar properties could be identified and then distributed as equally as possible among the three data sets. When the three representative data sets had been prepared, the ANN models were built. Separate ANN models were constructed for each of the three categories (marine, continental and estuarine) into which the data had been divided.

The models were evaluated by statistical measurements and graphical analysis, such as plotting the ANN model predictions vs. the actual data values and plotting the predicted residuals vs. the actual data values. The statistical measurements of the ANN model performance are summarized in Tables 6-2 and 6-3.

Table 6-2: ANN Model Performance for Primary Recovery

Category	Statistical Measure	Training Set	Testing Set	Production Set
Marine	R Squared	0.92	0.72	0.62
	Mean Absolute Error*	4.5	8.2	9.8
Continental	R Squared	0.86	0.76	0.64
	Mean Absolute Error	5.3	6.2	7.6
Estuarine	R Squared	0.85	0.81	0.83
	Mean Absolute Error	7.4	7.5	7.6

*The entries for mean absolute error are expressed in units of percent, since primary recovery is expressed in these units.

Table 6-3: ANN Model Performance for Primary Froth Quality

Category	Statistical Measure	Training Set	Testing Set	Production Set
Marine	R Squared	0.84	0.71	0.61
	Mean Absolute Error*	3.5	4.3	5.2
Continental	R Squared	0.87	0.80	0.81
	Mean Absolute Error	4.2	4.8	5.2
Estuarine	R Squared	0.92	0.83	0.84
	Mean Absolute Error	3.6	4.9	5.0

*The entries for mean absolute error are expressed in units of percent, since primary froth quality is expressed in these units.

The statistical measurements listed in Tables 6-2 and 6-3 provided a general evaluation of the models. The model performances were considered to be quite satisfactory. The mean absolute error in predicting primary recovery was 10% or less. The mean absolute error in predicting primary froth quality was 5% or less. Typically, the models performed better on the training and testing data sets than on the production set. This occurred because the more noisy data in the working database was deliberately excluded from the training and testing sets to encourage more rapid convergence of the ANN models. The more noisy data was placed purposely in the production set to test the predictive capabilities of the models. Consequently, lower R squared values appeared in the statistics for the production set. In this study, it was very encouraging to observe that not only were the R squared values for the training and testing data sets high, but that the R squared values for the noisy production sets approached values that were similar to those of the training and testing sets. In the estuarine category, there was very little noisy data in the production set. Thus, the R squared values for the training, testing and production data sets were very similar in magnitude.

The performance of the ANN models was also examined graphically. Figure 6-4 and Figure 6-5 show the predicted values vs. actual data values on the production set, which was never used in the training or testing process. There was an outlier in Figure 6-4 with a primary froth recovery at 140%. This record was purposely put in the production set to test the model. Model predicted a 90% recovery, which is what the value should be under normal circumstance. In general, it was found that the models tended to under-predict high values and over-predict low values in some instances. There are two aspects to this phenomenon. First, noisy patterns and outliers were retained intentionally in the data sets to examine the noise-tolerance capabilities of the models. For these patterns, under-prediction or over-prediction is quite possible. However, it can be argued that in these cases, under-prediction or over-prediction is the correct model response. Second, the best way to evaluate whether there is a general trend in a model for over-prediction or under-prediction is to examine the plots of predicted residuals vs. actual data values. If the residuals are generally distributed randomly around their mean (in this case, the perfect mean is 0), then it can be concluded that there is not a trend to either over-prediction or under-prediction. The residual plots corresponding to the model predictions shown in Figures 6-4 and 6-5 are presented in Figures 6-6 and 6-7. After examining the residual plots for all of the ANN models, it was concluded that there is no general tendency on the part of the models to over-predict or under-predict the extraction data.

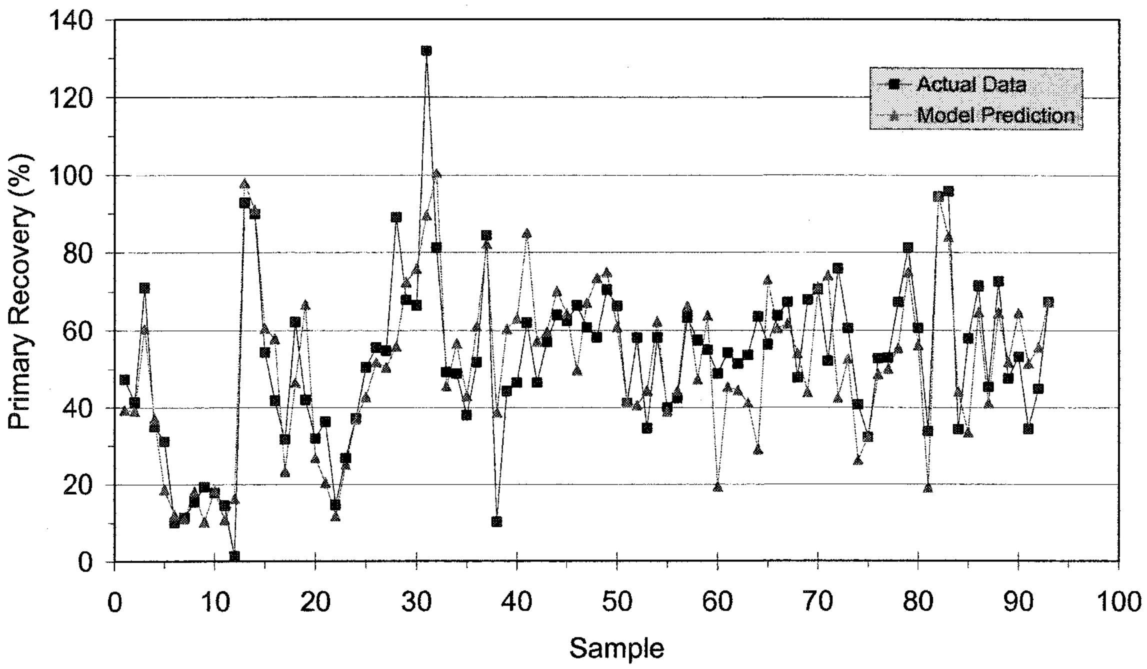

Figure 6-4: ANN Prediction of Primary Recovery for the Marine Category

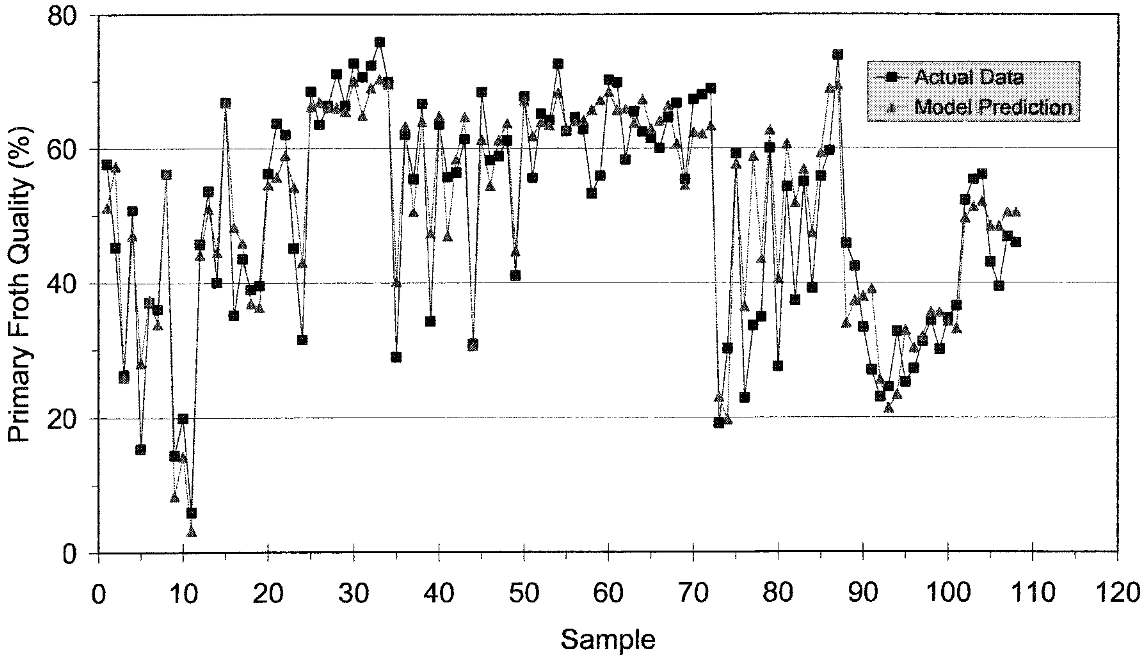

Figure 6-5: ANN Prediction of Primary Froth Quality for the Estuarine Category

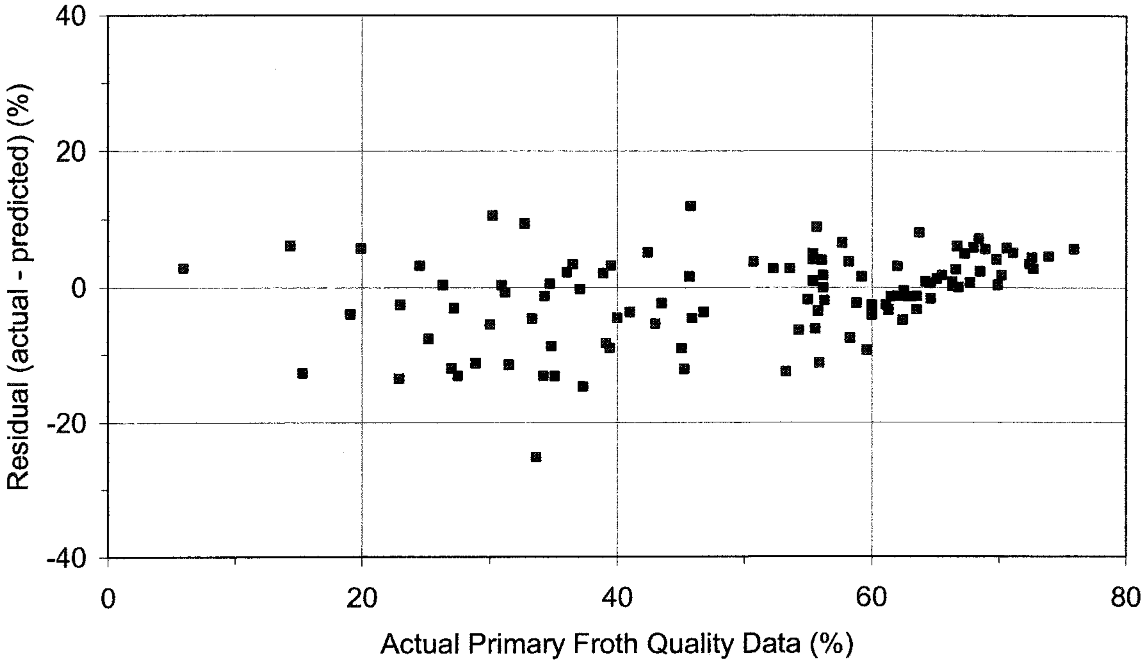

Figure 6-6: Primary Recovery Residuals for the Marine Category

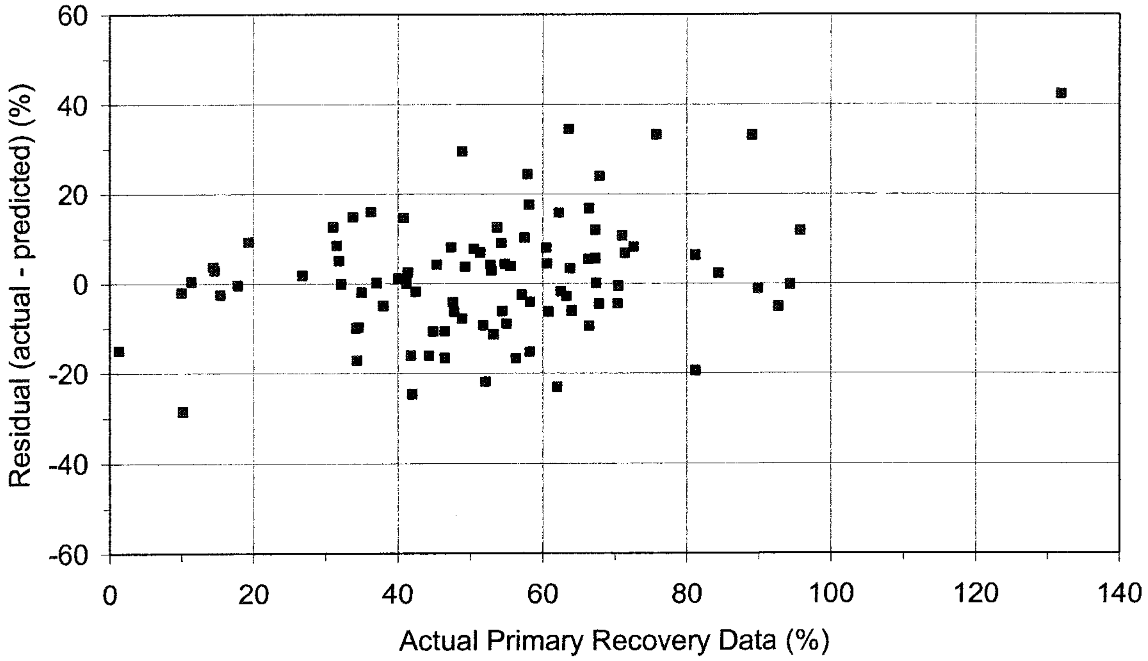

Figure 6-7: Primary Froth Quality Residuals for the Estuarine Category

Sensitivity Study

The sensitivity study was designed to test the neural network models in terms of their response logic, parameter interaction, and extrapolation capability. Since there are a myriad of possibilities to test, it is usually desirable to design a schedule to coordinate the tests. A test schedule normally specifies the key input parameters to examine, the range of parameter values to be examined, and the approach for setting fixed values for the other input parameters that are not under scrutiny.

Since a neural network model has multiple input parameters (in general, n of them), the study domain is an n-dimensional space. In order to examine the key parameters, the values of the other input parameters have to be fixed. This is equivalent to examining the key parameters in specific sub-spaces of the n-dimensional space. Obviously, for the examination to be effective, the locations of these sub-spaces must be well chosen. With the pattern recognition approach, the n-dimensional space is divided first into categories and then further into classes. Each class is a collection of sample data representing one type of processing operation. From this point of view, it is best to situate the test locations within the data clusters defining the classes. In this way, not only can the center of the data cluster be used for the fixed values of the parameters not under scrutiny, but also the data cluster provides a reasonable delineation of a boundary for testing extrapolation capabilities. The response of the model in the tests is also likely to be more meaningful because the test location is supported by many data points. This will not be nearly as likely if the test location is a random one in the n-dimensional space.

Some background to the specific sensitivity tests that were carried out is in order. It was noted in the introduction to this chapter that the ANN modeling technique bypasses the step of establishing a physical model for a process. However, some of the input variables that were recommended for inclusion in the neural network models were known, from basic physical chemical principles, to interact and affect the output of the extraction process.

For example, it is well known that the state of dispersion of clays in an aqueous medium is related to the water chemistry. High divalent ion concentrations result in a collapse of the electrical double layer around clay particles, a net attraction between particles results, and individual particles coagulate to form a gel that can interfere with the flotation of bitumen. Even in the absence of coagulation effects, the viscosity of the aqueous medium through which the bitumen must float in the flotation vessel increases with clay content. This, too, tends to impede the flotation of bitumen, suggesting a decrease in bitumen recovery is likely to occur.

The principles and observations described above were used as a guide in selecting the key input parameters to be examined in the sensitivity study. It was decided to focus the sensitivity study on the effects of clay content and salinity (ion concentration) on the two output variables, primary bitumen recovery and primary froth quality, from the ANN models. Therefore, the key input variables that were chosen to be examined were fines content (a surrogate measure of clay content), and the monovalent and divalent ion concentrations in the digestion water.

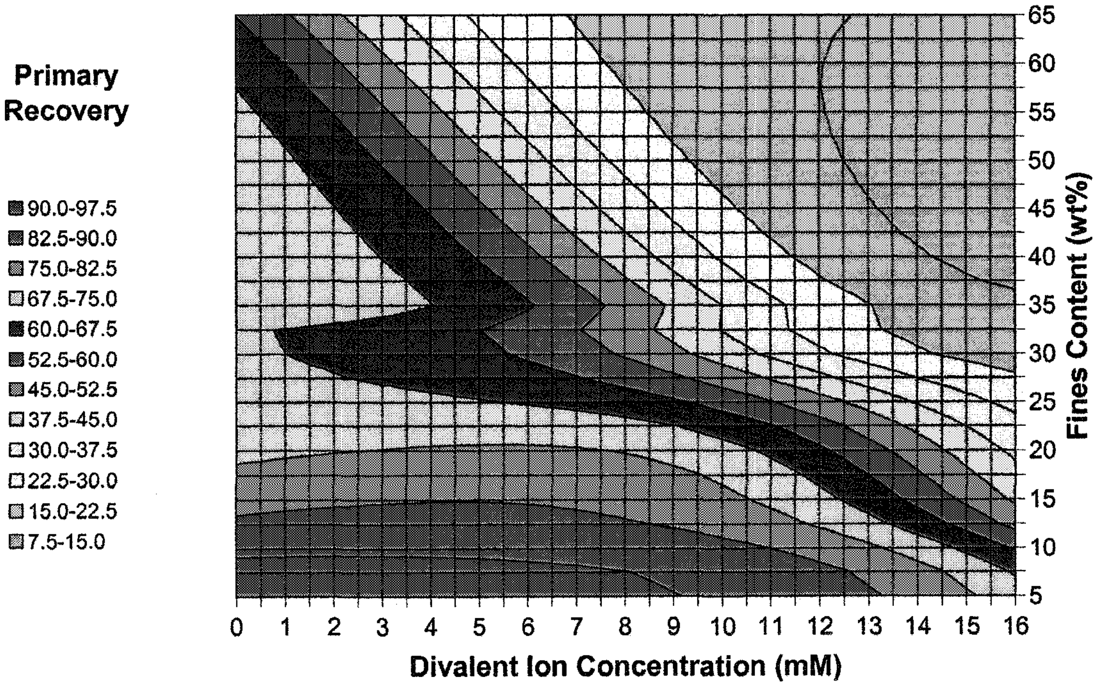

Figure 6-8: Primary Recovery Response to Fines Content and Divalent Ion Concentration

Figure 6-8 shows an example of the predicted response of primary recovery to changes in fines content and divalent ion concentration. The high percentage recovery is at the lower left corner of the figure. The relationship between fines content, divalent ion concentration, and bitumen recovery is entirely consistent with the physical chemical principles described above. Not only was the trend in bitumen recovery modeled by the neural network the same as the trend predicted from theoretical principles, but the abrupt decrease in bitumen recovery was identified as well by the ANN model.

The ANN model also successfully predicted a decrease in the quality of the recovered froth with increasing divalent ion concentration, as shown in Figure 6-9. However, the increase in froth quality with increasing fines content predicted by the model is not consistent with the physical chemical principles under discussion. A more detailed examination of the response of the model in this example would have to be carried out to determine the cause of this discrepancy.

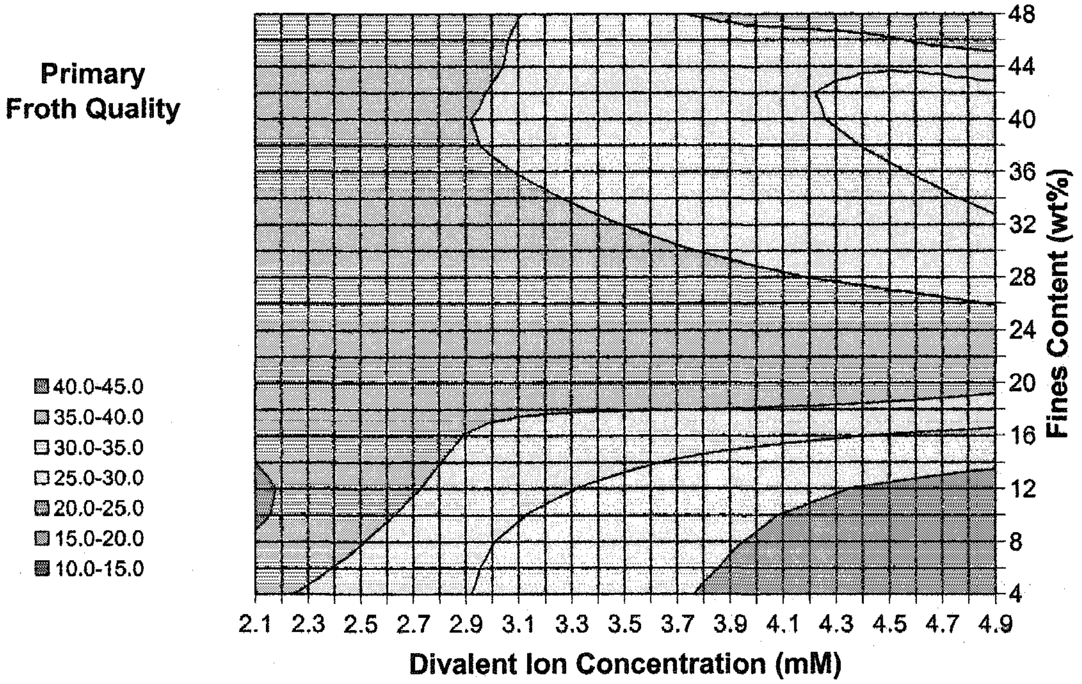

Figure 6-9: Primary Froth Quality Response to Fines Content and Divalent Ion Concentration

The final example discussed here, shown in Figure 6-10, is similar to the example shown in Figure 6-8 except that the input parameter divalent ion concentration is replaced by the parameter monovalent ion concentration. The low froth quality is located at the lower right corner of the graph. Monovalent ion concentration varies primarily as a consequence of the addition of sodium hydroxide (caustic) in the extraction process. The classic recovery response, characterized by relatively low recoveries at low levels of caustic addition, followed by a maximum and then a subsequent loss of recovery as caustic addition is increased, is well-known to those knowledgeable in the art of extraction. The response of the ANN model shown in Figure 6-10 is consistent with this popular belief.

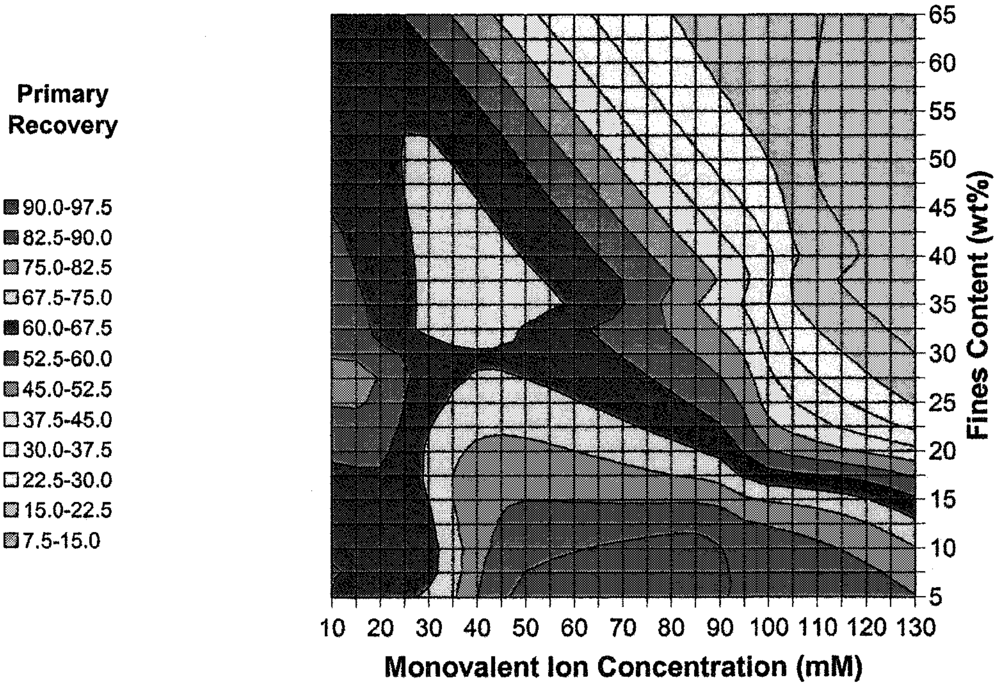

Figure 6-10: Primary Recovery Response to Fines Content and Monovalent Ion Concentration

The examples shown in these figures are not intended to demonstrate that, as a result of the ANN modeling, we completely understand the oil sands extraction process now. Rather, they illustrate how, out of a very complex set of data gathered over a twenty year period and not simply for the purpose of identifying clay/salt interactions, an ANN modeling approach was able to predict some of the relationships and interactions that were known to be present. This provides a level of confidence in the ANN approach, and suggests it has the potential to improve our understanding of the physical and chemical mechanisms of oil sands extraction.

Conclusions

The purpose of the work reported in this chapter was to apply the ANN modeling technique to analyze oil sands extraction data, generated from laboratory experiments testing various process designs. A new ANN modeling protocol, the pattern recognition technique, was utilized in the project to improve the neural network modeling procedures. As discussed in the previous two sections, the modeling results were very satisfactory. Although there is still room for fine-tuning the models, the current models have sufficient accuracy and robustness for various research purposes.

A key component of the project was the sensitivity analysis that could be performed using the ANN models. The sensitivity study examined whether the performance of the models was consistent with known oil sands extraction behaviour. For the most part it was, providing a level of confidence in the ANN approach. Moreover, the ANN models were successful, where standard statistical linear regression methods had failed, in identifying the presence of expected interactions among key input parameters from the extraction database.

The ANN models that were constructed for this project could continue to be used in a wide range of scenario studies. The models could be used to evaluate further aspects of the qualitative and quantitative response of important output variables in oil sands extraction to the behaviour of key input parameters. For example, they could be used to generate response surfaces and families of response curves to study the interactions among a pair of input parameters and a single output parameter. These response surfaces and curves could be used as tools in trying to improve the understanding of the physical and chemical mechanisms of oil sands extraction, allowing better engineering of extraction processes.

Acknowledgements

This chapter is a revised version of a paper, which I co-authored with Dr. Sawatzky, Dr. E. D. Wallace, Mr. Micheal London, and Dr. Steve Stanley. Dr. Ronald P. Sawatzky is a senior research scientist in the Petroleum Recovery Institute, Alberta Research Council (ARC), Edmonton, Alberta, Canada. Dr. E. Dean Wallace is a manager of strategic planning and development in ARC. Mr. Michael J. London is a research engineering consultant to the Petroleum Recovery Institute of ARC. Dr. Stephen. J. Stanley is the general manager for EPCOR Water Services, Edmonton, Alberta, Canada. Dr. Stanley is also an adjunct professor in the Department of Civil and Environmental Engineering, University of Alberta, Edmonton, Alberta, Canada.

The author would like to thank Mr. Gary Kovacik with the Alberta Research Council for his active and enthusiastic support for the project.

References

Anxionnaz, H.A., Delhomme, J.P. and Haan, S. D. (1999). Reconstructing Petrophysical Borehole Images: Their Potential for Evaluating Permeability Distribution in Heterogeneous Formations, SPE Paper 56786, 1999 SPE Annual Technical Conference and Exhibition in Houston, Texas.

Arehart, R.A. (1989). Drill bit diagnosis using neural networks. SPE Paper 19558, 1989 the 64th Annual Technical Conference and Exhibition of the Society of Petroleum Engineers, in San Antonio, TX.

Balch, R. S., Stubbs, B. S., Weiss, W. W., and Wo, S. (1999). Using artificial intelligence to correlate multiple seismic attributes to reservoir properties. SPE Paper 56733, 1999 SPE Annual Technical Conference and Exhibition in Houston, Texas.

Bowen, B.A. (1991). A neural network design methodology: considerations and issues for design and project management. *AGARD Lecture Series 179 - Artificial Neural Network Approaches in Guidance and Control*, 7 Rue Ancelle 92200 Neuilly Sur Seine, France, pp.3-1 -pp.3-20.

Dashevskiy, D., Dubinsky, V., and Macpherson, J.D. (1999). Application of Neural Networks for Predictive Control in Drilling Dynamics, SPE Paper 56442, 1999 SPE Annual Technical Conference and Exhibition, in Houston, Texas.

Flood, I. and Kartam, N. (1997). Neural networks in civil engineering I: principles and understanding. *Journal of Computing in Civil Engineering* 8(2): 131-163.

Mohaghegh, S. (1995). Neural network: what it can do for petroleum engineers. *Journal of Petroleum Technology 47(1) pp. 42.*

Murray, A., Neuroth, M., Stronach, A.F., and MacConnell, P.F.A. (1999). The Applications of Advanced Computing Techniques to Oil and Gas Facility Optimization. SPE Paper, 56904, 1999 SPE Offshore Europe Conference in Aberdeen, Scotland, 7-9 September 1999.

Olson, T. M. (1998). Porosity and Permeability Prediction in Low-permeability Gas Reservoirs from Well Logs Using Neural Networks, SPE Paper 39964, 1998 SPE Rocky Mountain Regional/Low Permeability Reservoirs Symposium and Exhibition in Denver, Colorado.

Reymond, S.B., Soenneland, L., and Strudley, A. (1999). Mapping Reservoir Saturation with Seismic Resolution to Improve Reservoir Model, SPE Paper 57259, 1999 SPE Asia Pacific Improved Oil Recovery Conference in Kuala Lumpur, Malaysia.

Shelley, R.F. (1999). Artificial Neural Networks Identify Restimulation Candidates in the Red Oak Field, SPE Paper 52190, 1999 SPE Midcontinent Operations Symposium, Oklahoma City, OK, March.

Simpson, P.K. (1991). Neural Network Paradigms, in *Artificial Neural Network Approaches in Guidance and Control* (pp. 2-1 to 2-33). AGARD Lecture Series 179, Advisory Group for Aerospace Research & Development, Neuilly sur Seine, France.

Zhang, Q. and Stanley, S. J. (1997). Forecasting raw-water quality parameters for the North Saskatchewan River by neural network modeling. *Water Research* 31(9): 2340-2350.

Zhang, Q. and Stanley, S.J. (2000). Modeling Artificial Neural Networks with Pattern Recognition Approach. *Journal of Computing in Civil Engineering,* submitted.

Chapter VII Conclusions

The major contributions of this thesis are advancing the theoretical research on the internal workings of artificial neural network, formulating an advanced ANN modeling protocol for complex study domains, and rectifying the proposed protocol in actual applications. In this thesis, a graphical analysis technique was first introduced to interpret the detailed internal activities of artificial neural networks at the level of artificial neurons and their connection links. Based on the new understandings of artificial neural networks, an ANN modeling protocol that integrates the pattern recognition technique with ANN modeling to effectively simulate complex domains was developed. With this protocol, much better accuracy in modeling and an additional capacity to detect the out-of-bound data can be achieved. The effectiveness of the protocol was further tested and demonstrated in two actual applications.

Compared to the rapid development of ANN applications in various fields of science and engineering, the research progress on the theories of artificial neural networks has been slow, particularly in the area of internal workings of ANN. This is mostly due to the complex networking of many non-linear processors (artificial neurons) inside a neural network. Without a clear understanding of the activities inside neuron networks, many explanations related to the behaviors of ANN models have to rely on weak assumptions, and thus are not very convincing. Lack of knowledge on the internal workings of ANN hinders the advancement of ANN modeling.

The breakthrough in deciphering the ANN inside structure was found recently as illustrated in Chapter 3. It was discovered that a feed-forward neural network could be considered as a function, though a very complex one. This assertion can be proven in two steps. First, it can be proved that any artificial neuron is a function provided that its activation function is a function. Second, the networking of neurons is equivalent to combine functions under the algebra rule and the composition rule of function to form a final function. Therefore, the overall network is a function.

Since a neural network is a function, many properties of function can be applied to clarify the behavior of neural networks. For example, by the definition of function, the mapping between the input domain and the output domain is limited to a one-to-one or a many-to-one relationship. The mapping is quantitatively affected by the coefficients of a function. Based on these key characteristics, a graphical mapping technique was developed to study the internal workings of neural network.

The studies are accomplished through several case studies. First case study is a simple case to explain the purpose of ANN inner components, such as the connection weights and the thresholds. Mathematically, the connection weights and the threshold are the coefficients of a function, thus directly impact the function mapping. Connection weights represent the links into or exit from a neuron. A threshold resides in a neuron. Both connection weight and threshold can shape a function curve. Connection weight can stretch, compress, and reflect a function curve either horizontally or vertically. Threshold can shift a function curve horizontally or vertically. As shown in Chapter 3, the learning algorithm manipulates these coefficients to shape each output function curve of every neuron in the network for final desired profiles. Clear understanding of these coefficients' functionality allows further manipulation of the internal components of neural network to achieve a specific goal.

In the second case study, the principles learnt in the first case were applied to study the impact of noise data in the ANN learning process. As shown in Chapter 3, each significant noise is purposely memorized within the network. The memorized noise establishes its impact zone center around the noise. From this study, it can be concluded that ANN models do memorize the features of a training data set. Furthermore, how to adjust the learning of the features and to control their impact zones becomes an important subject in the ANN model training process.

Thus, the last case study is used to demonstrate that the impact zone can be contained by more learning patterns in the neighborhood. The more correct learning patterns present in the training set surround the original noise location, the smaller impact zone would be. Therefore, when a training data set is free of noise, only the few most representative learning patterns are needed to train a good ANN model. On the other hand, if a training data set is noisy, significantly more data points are needed in the training to achieve a comparable accuracy because more correct data are needed to surround each noisy pattern to limit its impact zone. More data used in the training also increase the difficulty of the training process and the complexity of ANN architecture.

The case studies also illustrate the interpolation and extrapolation ability of ANN models. As shown previously, the noises affect the interpolation through their impact zones. Similarly the useful features also affect the interpolation through their impact zones. The closer to a feature center, the stronger the influence of that feature. The extrapolation with ANN models is mostly influenced by the shapes of the activation functions and in part by the scaling function of the source data. The ANN model prediction has a tendency to saturate in extrapolation. Generally, as an extrapolation point is progressively moving outward from the learned boundary, it falls into a flat response zone at the tail ends of the activation functions. In other words, the ANN model prediction would be the same at the saturation zone. This is a general phenomenon for the monotonic activation function family. Although the non-monotonic activation function family (such as the Gaussian function) has a wider range of active prediction zone, the saturation zones still exist but with a lesser range compared to those of the monotonic functions.

The best practical solution to the saturation of prediction in extrapolation is to scale the range of the original training data as far away from the saturation zone as possible. For example, during the scaling of the training data set, shrink the data range to (0.2 to 0.8) to allow 20% of potential extrapolation range in either end. This will effectively mitigate some of the extrapolation saturation problems.

Robustness of neural network was also examined in this thesis. This feature of neural network is mainly due to its non-linear networking structure of neurons. The networking structure provides a lot of flexibility to learn features of a study domain and store the features in its neurons. Any prediction from the network is a deduction resulting from consulting all the features stored in the network. Non-linear structure allows more flexible means to deduct a prediction and minimize the potential impact of a faulty input.

The ability to interpret an ANN model allows us to discuss many important issues in the ANN modeling protocol, such as under-training and over-training, from the perspective of internal architecture. As demonstrated in Chapter 3, under-training is most likely to happen in the minimum solution architecture, which is the simplest architecture to model the training patterns with an acceptable accuracy. Under such a scenario, the ANN network optimizes its storage capacity to learn every individual pattern. However, it has few resources left to model the underlying general trend. That is why the example shown in Figure 3-22 has some very poor interpolations. Under-training in the extrapolation zone is an intrinsic problem of ANN models. The only viable solution seems to be adding more learning data in the saturation zone.

Over-training happens when too many similar data patterns are in the training data set. If the ANN models are driven to reproduce the minor difference among the similar patterns, all the available resource will be devoted to learn the minor differences. Again there would be few resource left to detect and preserve the overall trend and main features, and the priority of the learning is disrupted.

To avoid under-training and over-training, the solution is to search for an optimum number of representative data points to cover all sections of a study domain. In this thesis, the pattern recognition approach is introduced to complete this task. The fundamental concept of pattern recognition is to cluster similar data points together to identify their common characteristics and to expose the differences among clusters. The clustering process divides the original complex domain into smaller and cohesive sub-domains. In the model developing stage, the data clustering process can be continued until the number of data points in each group is small enough and similar enough, such that selecting a data point into the training set is like selecting a sibling to represent a family. The advantage of this approach in selecting representative data is that it is systematic, consistent, fast, and fairly objective.

To further expand the classification concept into ANN modeling, the author designed an ANN modeling system with a hierarchical structure. Instead of building one single model to simulate a whole complex domain, a series of ANN models were built to cover each sub-domain. The ANN models for sub-domains form the modules of an overall system and are linked together with a data filter.

When the ANN system receives a data pattern, the data filter first identifies which category the data pattern belongs to, then it redirects the data pattern to the category it belongs. The ANN module responsible for that category takes the data pattern and produces a prediction. If the data point does not belong to any of the existing categories, a flag is raised to warn a potential new data in a new category. This data point is set aside in a special group. There would be no prediction available from the ANN system at this time. However, if sufficient number of data points accumulated in that special group over time, it is necessary to create an ANN model for the new category. The newly built ANN model would be just another module to the system. There is no need to scrape the whole model and rebuild again. With this module design, any model upgrade for the ANN system can be completed for individual components without causing major disruptions to the rest of the system. This is a very desirable feature for a system operating in an on-line real time environment.

The theories discussed above were put to test in the raw water color model upgrade project and the oil sands treatment process modeling project. The color model upgrade project was guided by an early version of the theories. The oil sands project was finished with the improved theories. As a result, the oil sands project not only took much less time to complete, but also achieved much higher accuracy. The sensitivity study immediately followed the model building process shows the ANN system not only responds logically and perceptively, but also detects and exhibits many interesting features buried within the source data.

In conclusion, the ability to interpret the internal structure of neural networks allows us to greatly improve the ANN modeling protocol. The improvements in the protocol in turn enable us to build ANN models with higher accuracy, shorter time frame, and higher degree of confidence. The successful ANN applications are, in return, the best evidences that confirm the achievements in the theoretical research. The advances in these three fronts not only provides a solid theoretical support and increases the transparency of the ANN modeling technique, but also build a solid foundation for industrial application and commercialization of the ANN modeling technique. The advanced ANN modeling technique discussed in this thesis have been applied to create a real-time ANN control system for the dissolved air flotation process in the Port Hardy Water Treatment Plant in Port Hardy, British Columbia, Canada. This is the first real-time on-line ANN control system in the world. The technology developed for the Port Hardy Water Treatment Plant is suitable for any small, medium, or large water plants. The technology is especially suitable for small plants because the ANN system allows remote monitoring and control of plants, and operates the system 24 hours a day like an expert. The technology will significantly improve the water quality and the operation reliability, and reduce the operation cost for small plants.