School of Electrical and Computer Engineering

A Neural Fuzzy Approach For Well Log And Hydrocyclone Data Interpretation

Kok Wai Wong

"This thesis is presented as part of the requirement for
the award of the Degree of Doctor of Philosophy
of the
Curtin University of Technology"

December 1999

ACKNOWLEDGEMENT

I would like to express my immense gratitude and appreciation to my supervisors; Dr Lance Fung and Dr Doug Myers; for their advice, support, guidance and inspiration throughout my course of study. I would also like to thank all the staff in the School of Electrical and Computer Engineering for their friendship and assistance over the period of my research. Special thanks to Dr Halit Eren for his assistance and expertise.

I would also like to thank Hugh Crocker and Robert Charlebois from Crocker Data Processing in supporting the project and supplying the test samples.

Finally, I am grateful to my fiancée, Elizabeth, whose understanding and patience made all these work possible.

ABSTRACT

A novel data analysis approach that is automatic, self-learning and self-explained, and which provides accurate and reliable results is reported. The data analysis tool is capable of performing multivariate non-parametric regression analysis, as well as quantitative inferential analysis using predictive learning. Statistical approaches such as multiple regression or discriminant analysis are usually used to perform this kind of analysis. However, they lack universal capabilities and their success in any particular application is directly affected by the problem complexity.

The approach employs the use of Artificial Neural Networks (ANNs) and Fuzzy Logic to perform the data analysis. The features of these two techniques are the means by which the developed data analysis approach has the ability to perform self-learning as well as allowing user interaction in the learning process. Further, they offer a means by which rules may be generated to assist human understanding of the learned analysis model, and so enable an analyst to include external knowledge.

Two problems in the resource industry have been used to illustrate the proposed method, as these applications contain non-linearity in the data that is unknown and difficult to derive. They are well log data analysis in petroleum exploration and hydrocyclone data analysis in mineral processing. This research also explores how this proposed data analysis approach could enhance the analysis process for problems of this type.

CONTRIBUTIONS

This research contributes to the field of data analysis by proposing an automatic, self-learning and self-explained data analysis approach. In achieving this, it is shown that statistical theory can be applied to analyse the functionality of Artificial Neural Networks (ANNs). The thesis extends the generalisation capability of the Backpropagation Neural Network (BPNN) and highlights the factors that affect it. New approaches to improve the generalisation confidence of these networks are also proposed by employing a Self-organising Map (SOM) data-splitting validation and the interactive reinforcement learning approach. It is shown that the learning of an ANN can be controlled in such a way as to allow human interaction via a Modular Neural Network and through controlling the distribution of data in interactive reinforcement learning. In addition, a new input contribution method is proposed to identify significant input parameters.

It is also shown that combining the advantages of both ANNs and Fuzzy logic allows rules to be created to describe the generalised function of what an ANN has learned. This permits human interaction in the learning.

The value of this data analysis approach proposed is demonstrated in two complex application areas. It is shown they offer a significant improvement over previous methods and are an attractive option for the resource industry.

AUTHOR'S NOTE

Parts of this thesis have been previously reported in the following conference, newsletters and journal papers:

1. C.C. Fung, K.W. Wong, H. Eren and R.Charlebois, "Lithology Classification Using Self-Organising Map," in *Proceedings of IEEE International Conference on Neural Networks*, December 1995, Perth, pp. 526 - 531.

2. C.C. Fung, K.W. Wong, H. Eren, R. Charlebois and H. Crocker, "Modular Artificial Neural Network for Prediction of Petrophysical Properties from Well Log Data," in *Proceeding of IEEE Instrumentation and Measurement Technology Conference*, June 1996, Brussels, pp. 1010 – 1014. And in *IEEE Transactions on Instrumentation & Measurement*, 46(6), December 1997, pp. 1259-1263.

3. H. Eren, C.C. Fung, K.W. Wong and A. Gupta, "Use of Artificial Neural Networks in Estimation of Hydrocyclone Parameters with Unusual Input Variables," in *Proceedings of IEEE Instrumentation and Measurement Technology Conference*, June 1996, Brussels, pp. 1015-1019.

4. K.W. Wong, C.C. Fung and H. Eren, " A Study of the Use of Self-Organising Map for Splitting Training and Validation Sets for Backpropagation Neural Network," in *Proceedings of IEEE Region Ten Conference (TENCON) - Digital Signal Processing Applications*, November 1996, Perth, pp. 157 - 162.

5. H. Eren, C.C. Fung and K.W. Wong, " Back Propagation Neural Network in Determination of Parameter d50c of Hydrocyclones," in *Proceedings of IEEE Region Ten Conference (TENCON) - Digital Signal Processing Applications*, November 1996, Perth, pp. 163 - 166.

6. H. Eren, C.C. Fung and K.W. Wong, "An Application of Artificial Neural Network for Prediction of Densities and Particle Size Distributions in Mineral Processing Industry," in *Proceedings of IEEE Instrumentation and Measurement Technology Conference*, May 1997, Ottawa, pp. 1118 - 1121.

7. C.C. Fung, K.W. Wong and H. Crocker, "Determining Input Contributions for a Neural Network Based Porosity Prediction Model," in *Proceedings of Eighth Australian Conference on Neural Network (ACNN'97)*, June 1997, Melbourne, pp. 35 - 39.

8. C.C. Fung, K.W. Wong and H. Eren, "Determination of a Generalised BPNN using SOM Data-splitting and Early Stopping Validation Approach," in *Proceedings of Eighth Australian Conference on Neural Network (ACNN'97)*, June 1997, Melbourne, pp. 129 - 133

9. P.M. Wong, K.W. Wong, C.C. Fung and T.D. Gedeon, "A Neural-Fuzzy Technique for Interpolating Spatial Data via the Use of Learning Curve," in *Proceedings of International Work-Conference on Artificial and Natural Neural Networks (IWANN'97)*, June 1997, Canary Islands. And in J. Mira, R. Moreno-Diaz, J. Cabestany, *Biological and Artificial Computation: From Neuroscience to Technology*, Springer-Verlag, pp. 323 - 329.

10. C.C. Fung, K.W. Wong and H. Crocker, "Using Neural Networks for Log Analysis?" in *SPE News Australasia*, July 1997, Issue 18, pp. 12 - 13.

11. H. Eren, C.C. Fung, K.W. Wong and A. Gupta, "Artificial Neural Networks in Estimation of Hydrocyclone Parameter d50c with Unusual Input Variables," in *IEEE Transactions on Instrumentation & Measurement*, 46(4), August 1997, pp. 908 - 912.

12. C.C. Fung, K.W. Wong and P.M. Wong, "A Self-generating Fuzzy Rules Inference Systems for Petrophysical Properties Prediction," in *Proceedings of IEEE International Conference on Intelligent Processing Systems*, October 1997, Beijing, pp. 205 - 208.

13. K.W. Wong, C.C. Fung and H.Eren, "Modifying the Generalisation Characteristics of a Neural Network with Interactive Reinforcement Training," in *Proceedings of IEEE International Conference on Intelligent Processing Systems*, October 1997, Beijing, pp. 472 - 476.

14. C.C. Fung, H.Eren, K.W. Wong and C. Maynard, "Determining Significant Input Contributions to an Artificial Neural Network Model of a Hydrocyclone used for Particle Separation," in *Proceedings of the International Symposium on Manufacturing Technology ISMT'97*, Novermber 1997, Auckland, pp. 223-227.

15. C.C. Fung, and K.W. Wong, "Establishing a Generalised Fuzzy Interpretation System Using Artificial Neural Network," in *Proceeding of the International Conference on Computational Intelligence and Multimedia Applications 1998*, February 1998, Melbourne, pp. 330-335.

16. C.C. Fung, K.W. Wong, H. Eren, "Developing a Generalised Neural-Fuzzy Hydrocyclone Model for Particle Separation", in *Proceedings of IEEE Instrumentation and Measurement Technology Conference*, May 1998, Minnesota, pp. 334-337.

17. M. Jang, S. Cho, P. Wong, C.C. Fung and K.W. Wong, "Rock Porosity Prediction using Multilayer Predictions", *The Fifth International Conference on Neural Information Processing ICONIP*, October 1998, vol. 2, Kitakyushu, pp. 1016-1019.

18. C.C. Fung and K.W. Wong, "Petrophysical Properties Interpretation Modelling: An Integrated Artificial Neural Network Approach," *International Journal of Systems Research and Information Science*, 1999, vol. 8, pp. 203-220.

19. H. Crocker, C.C. Fung, K.W. Wong, "The STAG Oilfield Formation Evaluation: A Neural Network Approach", *APPEA'99 Journal*, Vol 39, Part 1, April 1999, pp. 451-460.

20. Wong, K.W., "A Reduced Rule Base Neural-Fuzzy Well Log Interpretation Model", *Inter-University Postgraduate Electrical Engineering Symposium*, July 1999, pp. 37-38.

21. Fung, C.C., Wong, K.W., and Myers, D. "An Intelligent Data Analysis Approach Using Self-Organising Maps", *The Sixth International Conference of Neural Information Processing ICONIP*, November 1999, vol. 2, pp. 735 - 738.

22. Wong, K.W., Myers, D., and Fung, C.C. "A Generalised Neural-Fuzzy Well Log Interpretation Model With A Reduced Rule Base", *The Sixth International Conference of Neural Information Processing ICONIP*, November 1999, vol. 1, pp. 188 - 191.

TABLE OF CONTENTS

CHAPTER 4:

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 THE DATA ANALYSIS PROBLEM

In most engineering applications, the role of data analysis is critically important. The data analysis approach used must be able to provide a reasonable summary as well as an analysis of the data. There are two broad categories of data analysis; descriptive and inferential (Mendenhall and Sincich, 1992; Phipps and Quine, 1998). Descriptive analysis simply aims to find a description of the data as presented solution. No prediction of what might have been achieved outside the range is expected, nor should it be undertaken. For inferential analysis, however, the analysis tool is expected to derive the underlying function from which the data derives and therefore allow the prediction of data that could be expected in the experiment for different input values.

Clearly, inferential analysis is the more complex problem. It faces the difficulty that it may only ever process a sample and that may be an incomplete description of the population. By implication, inferential analysis must extract as much information as possible from the sample and draw sound inferences about the population.

In most applications, whatever data analysis approach is adopted, it is required to offer reasonable interpolation performance and provide some indication when extrapolation is appropriate. Given the diversity of potential problems, it is inappropriate to consider a generic data analysis approach. However, with slight modification, any new data analysis technique should obviously be applicable to a particular class of applications.

This thesis presents a new quantitative inferential analysis technique using predictive learning. Predictive learning systems (Keans, 1994; Vapnik, 1995) attempt to construct useful prediction functions purely by processing data taken from past successfully resolved problems. They assume, as they must, that all useful information is available in the supplied data. However, being a learning system, their analysis can shift in the light of new information.

The inferential analysis technique presented can solve multivariate non-parametric regression problems. Hence it can be used to deal with non-linear or random data (sometimes with bias). It is robust in the presence of noise. Inferential analysis normally makes use of a predictive learning algorithm to extract knowledge from the supplied sample when dealing with non-linear, random, noisy and heterogeneous data. In statistics, the empirical model, multivariate non-parametric regression analysis and discriminant analysis are usually employed (Hardle, 1990; MacLachlan, 1992). Although these approaches are widely used, they do have their limitations (Cherkassky et. al, 1994). They can normally deal with only small amount of training data and as some prior assumption need to be made, it is very difficult to analyze complex problems. Statistical approaches are based on structured models and therefore they are very computationally complex. Further, it is difficult for non-statisticians to understand and use them. Statistical approaches tend to be inflexible, as it is very difficult to find an analysis model that applies universally to any class of problem. Most of the time, the operating conditions can change from one operation to another. It is also tedious to build another model every time the operating condition changes. All these problems present an argument for the search for a better data analysis approach to handle the same degree of analysis.

1.2 EMPLOYING ARTIFICIAL NEURAL NETWORKS AND FUZZY LOGIC IN DATA ANALYSIS

Recently, Artificial Neural Networks (ANNs) have emerged as an option for inferential data analysis (Friedman, 1994; Bruce and Robert, 1994). The observation sample that is used to derive the predictive model is known as training data in an ANN development. The independent variables, or the predictor variables, are known as the input variables and the dependent variables, or the responses, are known as the output variables.

In supervised learning (Kartalopulos, 1996), an ANN makes use of the input variables and their corresponding output variables to learn the relationship between them. Once found, the learned ANN is then used to predict values for the output variables given some new input data set. For unsupervised learning, an ANN will only make use of the input variables and attempts to arrange them in a way that is meaningful to the analyst.

ANN analysis is quite similar to statistical approaches in that both have learning algorithm to help them realise the data analysis model. However, an ANN has the advantages of being robust with the ability to handle large amounts of data. Novice users can also easily understand the use of an ANN. An ANN also has the ability to handle very complex functions (Cherkassky et. al, 1994). There are some limitations. For example, the quality of the results predicted cannot be assured and the data analysis model built may not be able to be interpreted. A more detailed analysis of ANN limitations is discussed in Chapter 2.

Fuzzy Logic (FL) is also becoming popular in dealing with data analysis problems that are normally handled by statistical approaches or ANNs (Kosko, 1997). However, traditional FL data analysis systems do not have any learning algorithm to build the analysis model. Rather, they make use of human knowledge, past experience or detailed analysis of the available data by other means in order to build the fuzzy rules for the data analysis. The advantages of using FL are the ability to interpret the analysis model built and to handle fuzzy data. The data analysis model can also be changed easily by modifying the fuzzy rule base. The major limitation is the difficulty in building the fuzzy rules due to lack of learning capability.

ANNs and FL are complementary technologies in designing an intelligent data analysis approach (Williams, 1994). That suggests combining the two (Nauck, 1995). For example, fuzzy logic could be used to enhance the learning capabilities or performance of the neural network. In another approach, a neural network and fuzzy system could be integrated into a single architecture. However, a human analyst may still have difficulties understanding the analysis model computed. Analysis of the prediction model is also very time consuming. Therefore, it was one of the prime objectives of the research presented to find a better way of combining the advantages of the ANN and FL such that these particular problems could be overcome. What has been achieved is a novel data analysis approach that is automatic, self-learning and self-explained that can provide accurate and reliable results.

1.3 THE PROBLEM OF WELL LOG AND HYDROCYCLONE DATA ANALYSIS

In engineering, the most important criteria in developing a data analysis approach is that it gives reasonable prediction results for practical problems. To validate the model presented in this thesis, two problems from the resource industry were closely examined. They are the problems of well log data analysis in petroleum exploration and hydrocyclone control in mineral processing.

Well log data analysis plays an important role in petroleum exploration. It is used to identify the potential for oil production at a given source and so forms the basis for estimating the financial returns and economic benefits. More specifically, it is the means of predicting the petrophysical properties of each well. That has a significant impact on the total budget spent on coring. More details on well log data analysis are given in Chapter 2.

Hydrocyclones find extensive applications in mineral processing for the classification and separation of solids suspended in fluids. This task is important, as any mistake in classification will result in huge losses. Due to the complexity of the separation mechanism in the hydrocyclone, the interpretation of the physical behaviour and forces acting on the particles is not clear. The task of hydrocyclone data analysis is to describe this performance. More details are given in Chapter 2.

1.4 THE CONTRIBUTIONS OF THIS THESIS

This research has examined most of the factors that contribute to the successful application of ANNs to any data analysis problem. That is to say, the ability to handle a very large amount of training data, the generalisation capability of the ANN, and the reduction of the unnecessary input variables to reduce the network's parameters. Most of the processes in the proposed approach have made to be automatic. This is of benefit to any novice users who have little knowledge about ANNs.

The integration of an ANN with a Fuzzy logic system is also proposed. The objective in this case was to allow user interaction in the development of the data analysis model.

It is shown the proposed data analysis approach is able to address the limitations of the statistical, ANN and FL approaches raised in the previous sections for at least the two problem classes analysed. This research has also identified that in order to obtain reliable results, a more integrated approach than has been customary is needed. Consequently, the proposed approach includes a variety of different components.

For cases where there are very large amounts of available sample data, it is always safe to assume that the underlying functions are very complex for an ANN to learn. This research has shown that with some ANNs, it is possible to learn the complex functions perfectly in a much shorter period of time than usual. This leads to a structure that has been termed a Modular Neural Network.

The most important feature of an ANN is the ability to generalise. For that reason, an intensive study was made of this issue. It was found that statistical approaches could be a very powerful and useful tool for analysing the generalisation capability of an ANN and lead to a good insight into the process. Factors that contribute to the generalisation capability of the ANN are also highlighted by this approach. This has lead to new approaches being formulated in this thesis to ensure that the generalisation of an ANN can be achieved. One of these new approaches also allows a human analyst to control how an ANN learns. Results presented show that these proposed techniques provide better results.

When dealing with large number of input variables, a human analyst would appreciate a straightforward technique to identify the significant input variables. An input contribution measure is proposed for this task. This is easy to use and quick to determine the significant input variables. Results presented show that it is a reliable way of identifying significant input variables.

Finally, a Generalised Neural-Fuzzy System has been developed to allow an ANN data analysis tool to provide a self-explanation function. It makes use of the fuzzy rules to explain the generalised function of the ANN such that a human analyst can understand how the data analysis model derives inferential results. This developed Generalised Neural-Fuzzy System will also allow the analyst to modify or add in knowledge or past experience into the model. It is shown that this proposed data analysis approach is an automatic, self-learning and self-explained that can provide accurate and reliable analysis results.

1.5 AN OVERVIEW OF THE THESIS

Chapter 2 overviews the problem of well log data analysis in the petroleum industry and hydrocyclone control parameters identification in mineral processing. It also reviews the problems of current data analysis methods used in these fields. This chapter also frames the objectives of this research and its significance.

The first part of Chapter 3 examines the possibility of applying an ANN to hydrocyclone data analysis. This chapter also makes a comparison between an ANN approach and conventional approaches used currently in the field. The second part of this chapter examines the advantages of using a Modular Neural Network (MNN) over a single Neural Network in data analysis.

Chapter 4 examines the generalisation capability of the Backpropagation Neural Network (BPNN) and the factors that affect it. New techniques and approaches are presented to extend that capability. Some of the formulated techniques also allow a human analyst to control how a BPNN should learn. This chapter demonstrates that statistical analysis of the BPNN can be used to better understand the generalisation capability of a BPNN. Statistical analysis may also verify the factors that affect the generalisation capability.

After investigating the use of an ANN as a data analysis approach, the uses of fuzzy logic are examined in chapter 5. A self-generating fuzzy rule algorithm is presented to extract rules from data. This chapter shows that combining the advantages of the integrated ANN data analysis tool and the self-generating fuzzy rule technique produces a Compact Generalised Neural Fuzzy Rule System. An efficient way of

identifying the significant input variables for predicting the corresponding output variables is proposed. This input contribution measure effectively reduces the number of fuzzy rules. As the number of fuzzy rules presented could be quite large, an algorithm to reduce the fuzzy rule base is also proposed.

Chapter 6 concludes the thesis. It examines how the objectives of this research have been accomplished. Future directions in this field of study are also suggested.

CHAPTER 2:

PROBLEMS OF THE ANALYSIS OF WELL LOG AND HYDROCYCLONE DATA

2.1 SIMILARITIES IN ANALYSISING WELL LOG AND HYDROCYCLONE DATA

Well log data analysis in the petroleum industry (Crain, 1986; Rider, 1996) and hydrocyclone data analysis in the mineral industry (Bradley, 1965) fall into the same class of non-linear data analysis problems. There are a large number of well-developed techniques for solving linear problems and some classes of nonlinear data analysis techniques (Mendenhall and Sincich, 1992). Nonlinear data analysis, especially where the nature of the nonlinearity is unknown, is far more difficult to deal with.

The problem in this case is an identification problem. There are known inputs to some 'black box' plus measured outputs. The problem is to determine a function that describes the link between the two.

In most instances, the techniques that approximate well to non-linear functions are those which can be generalised from the given set of input and output pairs. However, that set may not be perfect due to human or measurement error. That is to say, the data set is noisy. The main objective for data analysis is to make use of the given noisy and imprecise nonlinear data to enhance the desired output responses. This analysis also tries to reduce irrelevant and unwanted responses. In the past,

parametric or semi parametric approaches with some prior assumptions have been used to handle this form of data analysis. Non-parametric techniques have becoming more popular in recent decade due to the improvement in computing power.

In this thesis, well log data analysis in petroleum industry and hydrocyclone data analysis in mineral industry are used as examples to illustrate the power of the data analysis approach proposed. Although the collection of the data in these two fields is different, they both fall into the same category of inferential nonlinear data analysis problems.

A well or drill hole is made in order to gain information on some region. Samples extracted from the underground cores are examined intensively to obtain the desired outputs; the petrophysical properties of the well. Hopefully, this well log data will then allow a good prediction of the petrophysical properties of the area as a whole. Well log data analysis is largely concerned with forming such predictions.

For hydrocyclones, the input and output parameters are measured in an experimental laboratory and used to form the final design of the system. The objective here is to predict the output parameters and so the function of the system.

Although these problems seem to be different, they have many similarities. The data involved in both cases are non-linear, random, noisy, and sometime may be heterogeneous. In both cases, too, the desirable form of a data analysis tool is a system which is automatic, self-learning, and self-explaining that can provide accurate and reliable prediction results. In neither case is the objective to replace the

human analyst involved. Rather, it is to provide assistance to them to make their broader task easier. Those analysts need to be able to examine and understand the designed data analysis model. Further, as will be indicated, it is extremely useful if they can also manipulate and incorporate prior knowledge or experience into the model.

2.2 THE PROBLEM OF WELL LOG DATA ANALYSIS

The cost of developing a petroleum reservoir now requires exploration to be a very carefully managed and controlled process. The initial phase normally involves a series of boreholes being drilled at different locations around the region believed to hold the reservoir. Then well logging instruments are then lowered into each borehole to collect data typically every 150mm or so of depth. These data are known in the industry as w*ell log data*. Now follows a very intense processing of this data in order to commence an evaluation of the reservoir's potential.

Well logging instruments used for this data acquisition broadly fall into three categories: electrical, nuclear and acoustic (Rider, 1996). Examples of the measurements obtained are Gamma Ray (GR), Resistivity (RT), Spontaneous Potential (SP), Neutron Density (NPHI) and Sonic interval transit time (DT). There are over fifty different types of logging tools available for different requirements. Measurements of the formation and fluid properties in and around the well bore location are also usually included in the well log data.

Physical rock samples from various depths are obtained by using a coring barrel to recover intact cylindrical samples of reservoir rock. These samples are then sent to a laboratory and examined using various physical and chemical processes. Data obtained from this phase are known as *core data* in the log analysis process. Although core data is the most accurate way of assessing the hydrocarbon of a well, they are very difficult and expensive to obtain. Means of providing good prediction of the petrophysical properties is necessary to avoid spending excessive amounts of money on coring. Therefore it is important to establish an accurate well log data analysis procedure to provide reliable information for the log analyst.

Two key issues in reservoir evaluation using well log data are the characterisation of formation and the prediction of petrophysical properties. Examples of petrophysical properties are porosity, permeability and volume of clay. While a core data set gives an accurate picture of the petrophysical properties at specific depths, it takes a lengthy process and incurs great expense to obtain such data. Hence, only limited core data are available at selected wells and depths. The objective of well log data analysis is therefore to establish an accurate interpretation model which can be used to predict the petrophysical properties for uncored depths and boreholes around that region (Crain, 1986; Asquith and Gibson, 1982).

An accurate prediction is essential to the ultimate determination of the economic viability of the exploration and the production capacity of the particular well or region. Ideally, the model can be used to interpret log data from wells within the neighbouring region without the need to carry out further core analysis. This requires an integrated knowledge of the tool responses and geology together with various

mathematical techniques. This is important in order to derive an interpretation model that relates the log data to the petrophysical properties. However, the establishment of an accurate well log interpretation model is not an easy task due to the complexity of different factors that influence the log responses (Doventon, 1986). This demands a high level of human expertise, experience and knowledge.

A large number of techniques have been introduced in order to establish an adequate interpretation model over the past 50 years (Marett and Kimminau, 1990). The way the well log analysis is carried out has also changed due to the development in logging tools. It has also changed due to the development of the physics of porous media and the development of computer technology. However, the derivation of such interpretation models normally falls into two main approaches: empirical and statistical.

In the empirical approach, mathematical functions relating the desired petrophysical properties based on several well log data inspired by theoretical concepts are used (Coates and Dumanoir, 1974; Kapadia and Menzie, 1985). This approach has long been favoured in the field and much effort has been made to understand petroleum engineering principles. However, in cases where the geological characteristics are different, the empirical models may not perform well. They may also fail in cases where formations are separated by great distances or formed in totally distinct deposition environments. The unique geophysical characteristic of each region prevents a single formula to be universally applicable. As the number of parameters that the mathematical functions can handle is limited, it is also difficult to establish an accurate model. Further, it is inflexible, thus it takes much time and effort to

present another empirical model for the new situation.

Statistical techniques are viewed as a more practical approach to this problem (Wendt et al., 1986; Yao and Holditch, 1993; Condert et al., 1994). The common statistical techniques used are regression analysis and discriminant analysis. The simplest form of regression analysis is to find the relationship on a two dimensional crossplot. The derived regression equations are then used to predict the petrophysical properties. The equations are used to predict in the same well where core data are not available or other wells around the region. However, a number of initial assumptions of the model need to be made. Assumptions must also be made of the statistical characteristics of the data.

These assumptions will normally over-simplify the data and smooth out real variations in it. They also provide bias due to their estimation nature. In cases where complex analysis needs to be exercised, multiple linear regression and non-linear regression techniques can be used. Discriminant analysis is a multivariate technique designed to separate samples into groups based on information presented by the training data. It also requires certain statistical assumptions to be made before analysis is carried out.

Statistical techniques lack universal capabilities and their successful application is an inverse function of the problem complexity. When the problem is complex, the assumptions are more difficult to estimate correctly. Statistical techniques also limit the number of well log data that can be handled at the same time. With the increasing number of instruments and log data, it becomes difficult to apply the traditional

statistical and graphical methods.

Before going further, the factors that could possibly affect the accuracy of the well log analysis should be examined. First, as the logging tools are ever increasing, it is difficult to understand the non-linear response equations of some of the tools. This will make any prior assumptions on the statistical model difficult. Secondly, uncertainties may arise from the errors in log measurement such as depth mismatching and bad hole. Thirdly, human error could occur in the process of core analysis to generate core information. These problems have provided the motivation to explore other analysis approaches.

In the past few years, another technique that has emerged as an option for well log data analysis is the Artificial Neural Network (ANN) (Baldwin et al., 1990; Wiener et al., 1991;Rogers et al., 1992; Osborne, 1992). ANN performs analysis in a fundamentally different way from the traditional empirical and statistical approaches. ANNs can also address most of the mentioned factors that could possibly affect the accuracy of the model. An ANN does not require a prior assumption of the functional form of the dependency. It also offers a numerical model free of estimators and dynamic systems as well as the capability of modeling complex nonlinear processes with acceptable accuracy and has the ability to reject noise.

An ANN is also different from a conventional computer program. A computer program does take in inputs, process them and returns some results, but the computational block constructed about an algorithm or heuristic. However, in an ANN, a carefully selected and representative set of training data is provided for the

ANN to learn the underlying mathematical model intrinsically. It is for this reason that an ANN is suitable for applications whose solution is unknown or difficult to determine.

Research has shown that an ANN can provide better well log data analysis than statistical approaches (Wong et al., 1995a). Most well log data analyses are based on the Multi-layer Neural Network (MLNN) using the backpropagation learning algorithm (Rumelhart et al., 1986), which is commonly known as a Backpropagation Neural Network (BPNN). A BPNN is suited to this application as it resembles the characteristics of regression analysis in statistical approaches. However, a BPNN is a non-linear and non-parametric technique. A BPNN also has the ability to perform pattern classification, function approximation and regression analysis. All these have made BPNNs popular in well log data analysis.

Although ANNs have been found to perform better than traditional statistical and empirical methods, log analysts still have reservations in using them. The main reason for this is that log analysts have little control on how and what the ANN should learn. Then after the ANN has learned the functions, the log analysts have no way to understand how the interpretation model predicts, nor can they perform any modification on that interpretation model. However, ANNs still have considerable advantages if the drawbacks can be overcome. This has been the main motivation for this research.

2.3 HYDROCYCLONE CONTROL MODELING

Hydrocyclones (Bradley, 1965) find extensive application in the mineral process industry where they are used for the classification and separations of solids suspended in fluids. They are manufactured in different shapes and sizes to suit specific purposes. Hydrocyclones normally have no moving parts. The feed slurry containing all sizes of particles enters the hydrocyclone. Inside, due to centrifugal force experienced by the slurry, the heavier particles will be separated from the lighter.

After the particles suspended in the fluid are classified, they are discharged either from the vortex finder as overflow or from the spigot opening as underflow. Due to the complexity of the separation mechanism in the hydrocyclone, the interpretation of the physical behaviour and forces acting on the particles is not clear. Much work has been done on describing hydrocyclone performance using mathematical modelling (Kelsall, 1952; Bradley, 1965; Lynch and Rao, 1975; Plitt, 1976; Gupta and Eren, 1990).

The performance of a hydrocyclone is normally described by a parameter known as d50. This parameter determines the classification efficiency. It represents a particular particle size reporting 50% to the overflow and 50% to the underflow streams. The separation efficiency of hydrocyclones depends on the dimensions of the hydrocyclone and the operational parameters. Examples of the operational parameters are flowrates and densities of slurries. D50 is not a monitored parameter, but determined from separation curves known as *tromp curves*. They are used to

provide the relationship between the weight fraction of each particle size in the overflow and underflow streams.

In practical applications, the d50 curve is corrected by assuming that a fraction of the heavier particles is entering the overflow stream. This is equivalent to the fraction of water in the underflow. This correction of d50 is designated as d50c. The correct estimation of d50c is important since it is directly related to the efficiency of operations. Under normal industrial applications of hydrocyclones, any deviation from a desired d50c value cannot be restored without changing the operation conditions or/and the geometry of the hydrocyclone. Also, sensing the changes in d50c is a difficult task. It requires external interference by taking appropriate samples from the overflow and underflow streams. At the same time conducting lengthy size distribution analyses of these samples.

Gupta and Eren (1990) have discussed the automatic control of hydrocyclones. The output signal d50c cannot be sensed or conditioned directly, thus d50c needs to be calculated from the operation parameters. The automatic control of hydrocyclones can be achieved by manipulation of the operational parameters such as diameters of the spigot opening, the vortex finder height, the inlet flowrate, the density and the temperature of slurries for a set value of d50c. The correct prediction of d50c is essential to generate control signals.

Mathematically, d50c can be estimated from empirical models derived from experimental data by using analytical and statistical techniques. Some of the conventional formulae can be found in the literature (Gupta and Eren, 1990; Lynch

and Rao, 1975; Lynch, 1977; Mizrahi and Cohen, 1966; Plitt, 1976). Nevertheless, these models are hard to derive since the effect of each variable must be separately identified and incorporated into the formula. Most of the models have been derived by using multivariate analysis on the data where just one variable is varied while all others are held constant.

Because of these difficulties, all the conventional models are restricted to a few estimation variables. The common estimation variables are the flowrates and densities of slurries, the height of vortex finder, the fixed dimensions and the pressure differences. Since experimental conditions can change from one operation to another, the empirical models may not be applicable universally. This may explain the existence of many different formulae obtained using similar estimation variables. Even using the same test rig it is difficult to keep consistent operations over a period of time. Variables such as the solid contents and the particle size distribution within the slurry tend to fluctuate from time to time. In order to give a wider applicability to the conventional models, incorporation of additional control parameters, such as water and solid split ratios or densities is necessary. However, when using conventional approaches, it is difficult to include more control parameters and is also time consuming.

As ANNs are not popular in hydrocyclone control parameter identification, one of the objectives of this research was to initiate the use of ANNs in this field. The next chapter will examine this in more detail.

2.4 THE LIMITATIONS OF ARTIFICIAL NEURAL NETWORKS IN SOLVING THESE DATA ANALYSIS PROBLEMS

In most practical cases, using an ANN is not simple. In particular, if the training of the ANN is not handled properly, the results can be very disastrous.

When the amount of input data is large, the underlying function that the ANN needs to learn is normally very complex. Therefore, in order for the ANN to fully identify the function, learning should take a very long time. That aside, if the number of iterations is insufficient, the ANN may not be able to fully learn the underlying function. This may greatly affect the prediction accuracy of the model. This problem is discussed in Chapter 3, and a new technique is proposed to handle it.

A second issue is that the trained network should have learnt the underlying generalised function of the data instead of memorising the training data. Although ANNs are known to have a generalisation capability that sees a rejection of noise, this is not always achieved if the training is not handled properly. The capability of generalisation is particularly important in the situation where the data are very noisy. In most practical cases, noise exists in both the input variables and output variables. Noisy data normally means that there is an irregular mapping between the input and output. In designing a generalised ANN data analysis tool, the factors that will affect the generalisation capability need to be investigated intensively. Chapter 4 gives an intensive study of the generalisation capability of ANNs and current approaches to handling this problem are also reviewed at the beginning of that chapter. After investigating the factors that affect the generalisation capability; with the objective of

designing an automatic data analysis approach in mind; some straightforward and new approaches are proposed to ensure better generalisation capability.

Third, if the number of available input parameters is large, there must be some indications of the significance of each input to the output. This is especially important in well log data analysis, given the number of logging tools which may potentially be employed. As an ANN only consists of weights, it just acts like a "black box" to the analyst. Analysts need to spend a considerable time in understanding the configuration of the ANN before they can perform an input contribution measure. Garson (1991) and Wong et. al (1995c) have proposed an input contribution measurement in analysing the weights. This requires relatively complex analysis beyond the scope of a novice user of ANNs. Besides, the complexity of the weight vector in the ANN may prevent the derivation of an accurate input contribution measure.

Fourth, after an ANN is trained, it acts like a "black box" with only weight connections between the nodes. Unlike an empirical expression with limited terms and coefficients, the analyst would have difficulty in understanding the vast number of weights involved and how the network performs a task. In addition, if some weights of the ANN are modified, the effects on the output are unpredictable. Some kind of approach to represent the underlying function learned by the ANN is needed to provide a better understanding of the model. This problem may be solved by another technique that could express the function in human understandable rules known as Fuzzy Logic (FL) (Zadeh, 1965).

A fuzzy set allows for the degree of membership of an item in a set to be any real number between 0 and 1 (Zadeh, 1965). This allows human observations, expressions and expertise to be modelled more closely. Once the fuzzy sets have been defined, it is possible to use them in constructing rules for fuzzy expert systems and in performing fuzzy inference (Wang, 1991). Fuzzy reasoning is expressed as linguistic rules in the form "If x is A, then y is B", where x and y are fuzzy variables, and A and B are fuzzy values. This form of description corresponds well to the rules expressed by humans.

The use of Fuzzy Logic (FL) simplifies the development of an intelligent data analysis tool with the following features:-

1. Sophisticated knowledge and rich human experience can be incorporated into the fuzzy knowledge base in a form that is close to natural language.
2. The incorporated knowledge is not necessarily precise and complete.
3. The input facts to be accessed in fuzzy inference are not necessarily clear cut nor do they have to match the given knowledge exactly.
4. Partially matched conclusions can be inferred from the fuzzy facts and the established fuzzy knowledge base.

This approach is suitable to the application of most data analysis problem as the model for each situation may vary greatly and it allows the incorporation of intelligent and human knowledge to deal with each individual case. However, the extracting of fuzzy rules for the data analysis problem could be a nightmare to analysts with little experience. This could be a major drawback for use in well log

data analysis and hydrocyclone control parameters identification.

Several researchers have used ANN and FL together to perform a task (Lin and George, 1995; Jian, 1993; Zhang and Kandel, 1998). There are several ways where these two techniques can be made to work hand in hand to perform a specific data analysis task (Nauck, 1995). However, the analysts still have difficulties understanding the analysis model. Analysis of the prediction model is also very time consuming. Besides, the integration of the human experience and knowledge may still be impossible. Thus one of the major objectives of this research was to look for a straightforward Neural-Fuzzy data analysis approach. It should be automatic, self-learning and self-explained so that it could be used by any novice users. This data analysis approach should also be able to allow users to control the learning, and as well as be able to integrate knowledge or modify the model without too much difficulty.

CHAPTER 3:

A DATA ANALYSIS TOOL EMPLOYING AN ARTIFICIAL NEURAL NETWORK

3.1 INTRODUCTION

ANNs has many advantages over conventional approaches that use empirical formulae or statistical methods. ANNs have also shown successful application in well log data analysis. However, they are not popular for hydrocyclone data analysis. The first part of this chapter examines the feasibility of using ANNs in hydrocyclone control parameters identification. It also shows that the same ANN approach could be used in hydrocyclone data analysis problem. This approach has already been investigated for well log data analysis.

The shortcomings of ANN's need to be addressed and this is done later in the chapter. The most significant of these is that an ANN may not perform well when the data volume is very large. One important reason for this behaviour is that the underlying function is very complex. Another is that the number of iterations required in learning the complex function was underestimated or that the ANN learning time was insufficient. Possibly the distribution of the large data volume may have affected the ANN's ability to fully generalise the "truth" function. In the second part of this chapter, a new Modular Neural Network (MNN) is proposed that overcomes some of these difficulties.

3.2 EMPLOYING ARTIFICIAL NEURAL NETWORKS IN HYDROCYCLONE DATA ANALYSIS

3.2.1 ESTIMATING THE PARAMETER D50C

For the purpose of this study, a Kerb hydrocyclone model D6B-12°-839 was chosen. The slurry that was fed into the hydrocyclone for classification was made from a stock sample of -212μm sized quartz. The slurry was suspended in water circulating through the system. The quartz was mixed thoroughly each time before the slurry was made. Samples were taken manually from the overflow and underflow streams. Size analysis was then performed. After that, d50c can be estimated to assess the performance of the hydrocyclone. However, the size analysis is a time consuming and tedious task. Mathematically, the d50c can also be estimated from empirical models derived from analytical and statistical techniques. Some of the typical conventional formulae can be found in Gupta and Eren (1990), Plitt (1976), Lynch and Rao (1975) and Mizrahi and Cohen (1966). The five conventional hydrocyclone variables that were used to estimate d50c are the inlet flowrate, inlet density, the vortex finder height, the spigot-opening diameter, and the operating temperature.

3.2.2 BACKPROPAGATION NEURAL NETWORK (BPNN)

When a BPNN is used in hydrocyclone data analysis, the results from the physical size analysis are used as the training data. The input neurons of the BPNN correspond to the five hydrocyclone variables, and the output neuron is assigned to the d50c. The BPNN has a number of layers. The input layer consists of all the input

neurons and the output layer just the output neuron. There are also one or more hidden layers. All the neurons in each layer are connected to all the neurons in next layer with the connection between two neurons in different layers represented by a weight factor.

The objective of training the BPNN is to adjust the weights so that the application of a set of inputs produces the desired output. The training set consists of a number of desired input and output pairs. The input set is presented to the input layer of the BPNN. A calculation is done to obtain the actual output set by proceeding in order from the input layer to the output layer. At the output, the total error on each output neuron, which is the sum of squares of the differences between the desired output and the computed output is calculated. This value is used in a learning algorithm to update the weights and the process is back propagated through the network.

Once the modification of all the connection weights is done, a new set of outputs can be computed and subsequently a new total error will be obtained. This back-propagated process repeats until the value of the total error is below some particular threshold. At this stage, the BPNN is considered to have learned the function. After the BPNN has learned and generalised from the training data, it is then used to predict d50c under the same operational conditions.

3.2.3 A COMPARISON BETWEEN RESULTS OBTAINED THROUGH CONVENTIONAL METHODS AGAINST THOSE WITH THE BPNN

To see that a BPNN can be used in hydrocyclone control parameters identification, results obtained from such a network were compared to those generated from the Gupta's model (1990) and the Plitt's model (1976). Figure 3.1 illustrates these as compared to the observed d50c. It can be seen that BPNN's results are better for most values of d50c. Further statistical analysis of the data used to produce this figure gave the following. For Gupta's model, the correlation coefficient was 0.983 with r-squared value of 96.66%. For Plitt's model, the correlation coefficient was 0.895 with r-squared value of 80.14%. For the BPNN, in contrast, the correlation coefficient was 0.986 with an r-squared value of 97.17%. These are summarised in Table 3.1.

Table 3.1: Results from different analysis approach.

Analysis Approach	Correlation Coefficient	r-squared value
Gupta	0.983	96.66%
Plitt	0.895	80.14%
BPNN	0.986	97.17%

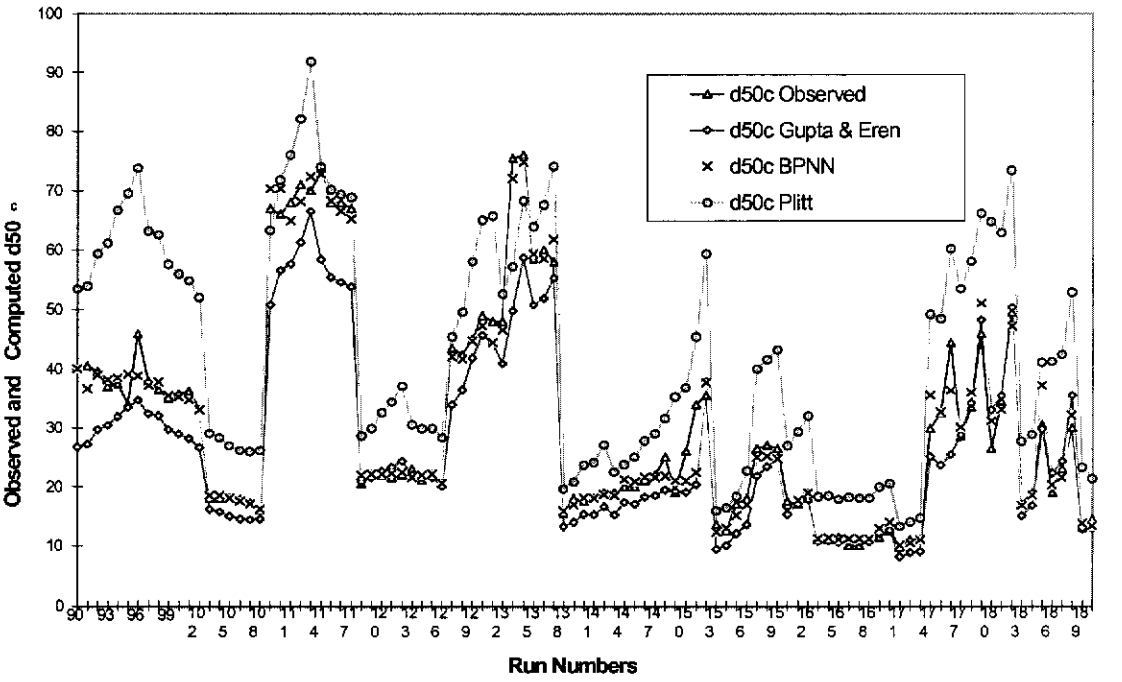


Figure 3.1: Results of the d50c using different approaches as compared to the observed data.

The BPNN approach clearly out-performs the other two. The reason why the results generated by Plitt's model have the worst accuracy is mainly due the inconsistency between the models. This is due to the fact that the hydrocyclone set-up condition for realising the Plitt model may be quite different to that of the Gupta model. While this BPNN analysis is based on the hydrocyclone set-up condition used to realise Gupta's model, that is why it performs well compared to BPNN.

This shows that the conventional approaches cannot be applied universally. They will only perform well when the operational condition is almost the same as that for which the model is derived. A BPNN is more flexible and should be able to be applied universally. A BPNN will generate promising results as long as the predicted results yield the same condition as those in the training data. Moreover, as a BPNN

has a learning capability, it is easy for it to re-build another model under a new operational condition. Table 3.2 also gives a summary of the comparisons between the conventional approach and the BPNN.

Table 3.2: Comparisons of conventional approach and BPNN.

	Conventional Approach	**BPNN**
Model assumption	Yes	No
Applied universally	No	Yes
Learning Ability	No	Yes
Rebuild another model	Difficult	Easy

3.2.4 ADDITIONAL ESTIMATION PARAMETERS AND RESULTS

Another advantage of using a BPNN is that incorporating new input variables is easy to do. The changes that need to be made to the BPNN are to increase the number of input neurons and present the extra input parameters in training. The learning capability of the BPNN will take care of the addition input parameters in the final function. In the conventional approach, new input variables often take a very long time to incorporate into the analysis. As the number of input variables increases, the conventional analysis process will become much more complicated. With an increasing number of input parameters, a stage will be reached where the analysis is so complex that it is not possible to incorporate them. However, due to the difference in the operational condition in different problems, it is important to include as many input parameters in the prediction as possible. Each input parameter, though, may not

be universally important in all, but it would be expected that inclusion of most of the available input parameters in the prediction model should lead to greater accuracy.

In Figure 3.2, three further parameters are included as input parameters in this analysis. The five conventional hydrocyclone variables that were used in the last section to estimate d50c are the inlet flowrate, inlet density, the vortex finder height, the spigot-opening diameter, and the operating temperature. With the three additional parameters, this makes the total number of input parameters used in predicting d50c to eight. The additional parameters were the underflow and overflow flowrates, and the ratios of the two flowrates. In this case, a statistical analysis indicated that the correlation coefficient of trained results increased to 0.995 giving an r-squared value of 98.9%. This gives a better prediction as compared with the conventional five input parameters. However, the training time for the BPNN is longer compared to the five input parameter problem. Based on a Pentium 90 PC, it took about 14 minutes to train for 8 inputs and 16 hidden neurons, but only 4 minutes for five inputs and 10 hidden neurons. The selected number of hidden neurons used in these two cases generated the best results. Although, the training for eight inputs is slower, once it has learned the function, d50c can still be generated in a very short time. It takes about 30 seconds to predict d50c for both cases.

Figure 3.3 depicts the trained BPNN results with fourteen input variables with the addition of water and solid split ratios, the overflow and underflow densities and the pressure difference between the inlet and the overflow streams. In this case, the correlation coefficient has further improved to 0.995 with an r-squared value of 98.97%. In this case, it takes about 20 minutes to train the network, while the

prediction still takes about 30 seconds.

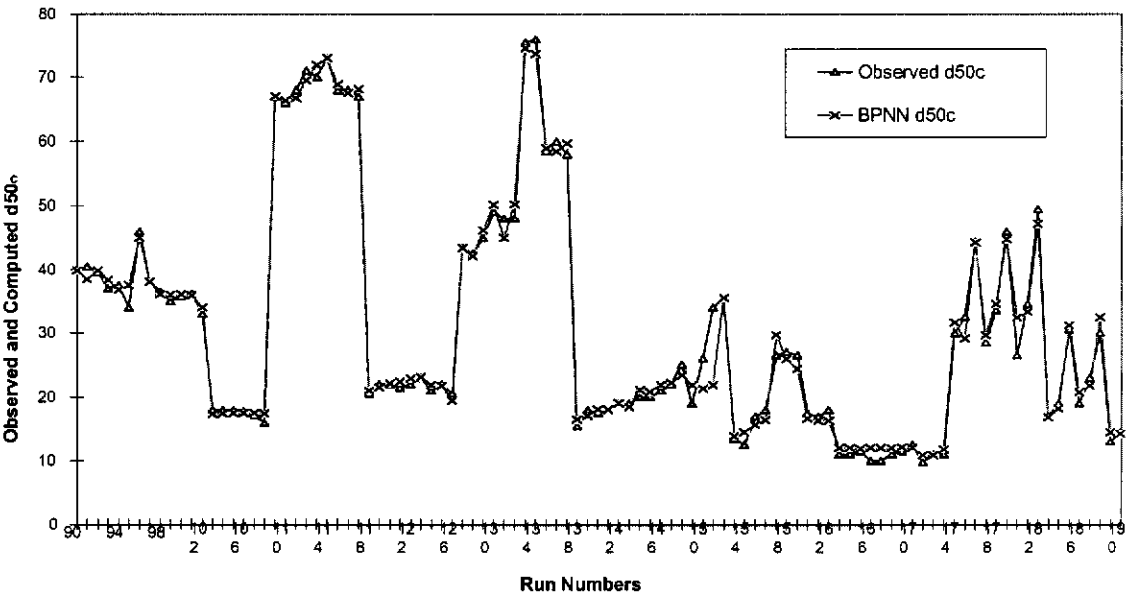


Figure 3.2: The data and predicted results with eight parameters

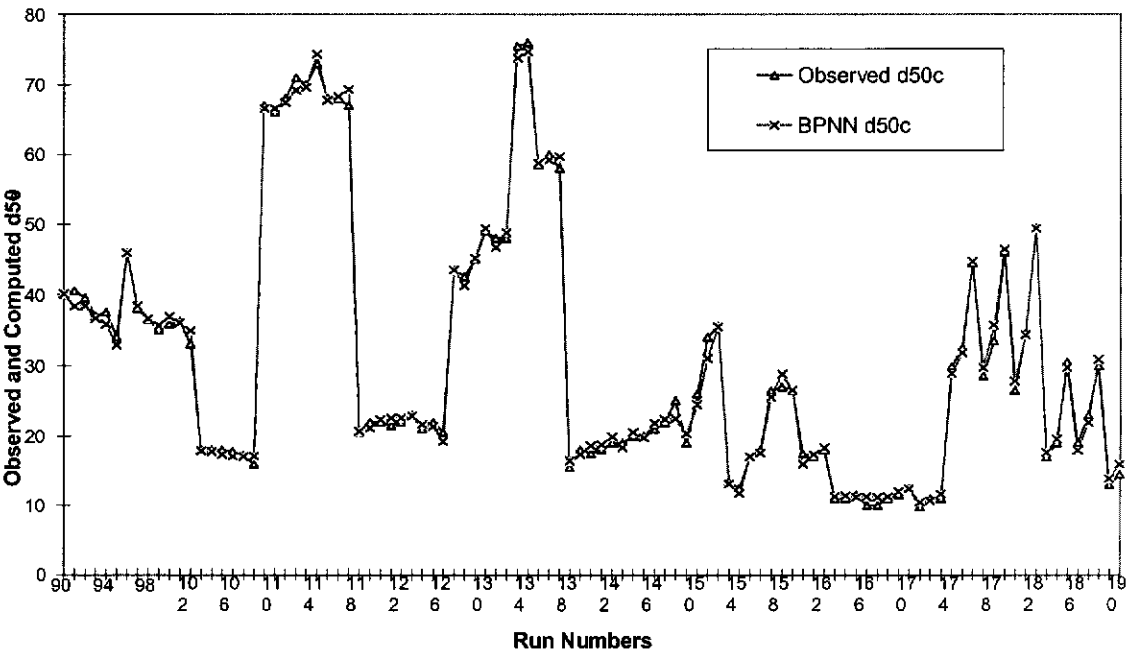


Figure 3.3: The data and predicted results with fourteen parameters

3.2.5 BENCHMARKING BPNN'S PREDICTION CAPABILITY

To really benchmark the performance of the BPNN, it should be used on data not used in the training process. Once a BPNN network is trained, the learning of the

network is assumed to be holding for any future data generated under the same operational conditions. The easiest way to test the BPNN is to arbitrarily select half of the available data from measurements for training purposes and use the other half for testing.

In this case study, the five input parameters were selected as those in Figure 3.1. Graphical results of tests are illustrated in Figure 3.4. Here, the correlation coefficient was found to be to 0.98 with an r-squared value of 97.67%, which is exceptional. This shows that the BPNN has the ability to make accurate predictions as long as the operational condition is the same as for the training data.

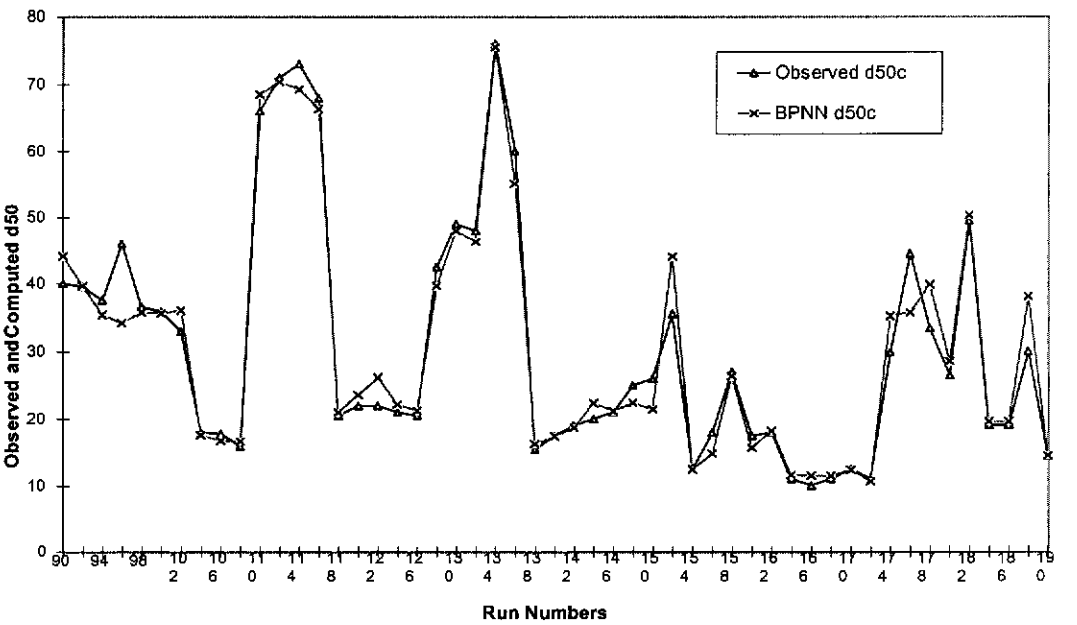


Figure 3.4 The results of the testing data.

3.2.6 AN ANALYSIS OF THE IMPROVEMENTS ACHIEVED THROUGH USING A BPNN

In the prediction of the hydrocyclone parameter d50c, the results from two best-known conventional models have been compared to those obtained by the application of a BPNN. It has been shown that in this case the BPNN gives better results that fit well with the original data. Application of the BPNN is also universal while conventional models only work well in particular situations. To re-build a new model for other operational conditions, BPNN can do that easily. Unlike conventional models, BPNN can also incorporate additional hydrocyclone input parameters easily. This will sometimes help to improve the prediction performance. There is no limit on how many input parameters the BPNN can handle outside of the problem of training time. Once the BPNN has learned the function, the prediction time is largely independent of the number of inputs. This first part of the chapter has also shown that BPNN is capable of producing accurate results to cases that are not seen in the training process. If the operational conditions remain the same as those for the training data, then the prediction generated from a BPNN is very promising. This study has indicated that the use of BPNN can lead to a more effective and efficient automatic control of hydrocyclones.

In spite of this, the drawbacks mentioned in chapter 2 would discourage the use of a BPNN in hydrocyclone control parameters identification. Further refinement of the approach is needed in order to win the confidence of potential users. The subsequent sections of this chapter and future chapters will examine these problems in details. New methods are proposed to handle them.

3.3 THE MODULAR NEURAL NETWORK

A problem that was clearly implied earlier in this chapter is how to deal with the complex functions underlying a large volume of available training data. To achieve this, the Modular Neural Network (MNN) is proposed. The MNN is based on the Self-organising Map (SOM) (Kohonen, 1989; Kohonen, 1990; Kohonen, 1995), Learning Vector Quantisation (LVQ) (Kohonen et al., 1992) and BPNN. Although the development of this network is presented using well log data analysis for illustration, it is easily employed for hydrocyclone control parameters identification or any similar problem. However, this proposed MNN can only be used when the available training data is large.

As compared to the conventional BPNN approach that uses only a single network, the MNN enables the division of a complex network into a number of sub-networks. Initially, a SOM and LVQ are used to classify the data. Several BPNNs corresponding to the number of classes obtained from the SOM are then trained for the purpose of prediction. Since the number of data to be handled by each sub-network is effectively reduced, the training time is significantly shortened. The data that falls into the same sub-network will have similar characteristics, thus effectively reducing the complexity of the function that the ANN needs to learn.

This discussion of the MNN is arranged into two major parts. The first will focus on the classification approach. The second will examine the prediction results of the MNN.

3.4 THE VALUE OF THE MODULAR NEURAL NETWORK

A BPNN is capable of learning any non-linear function from the available training data. However, if the available training data is large and complex as shown in Figure 3.5, the underlying function may be too complex for a single ANN to adequately represent it. This suggests a multiple representation that in turn leads to the prospect of modules. Noting Figure 3.6, if the data can be first classified before the BPNN learning process, the functions handled by each sub-section will be very much simpler compared to the whole set of training data. Further, the overall function should be able to learn in a shorter time. Given that function is better represented, the prediction results should also increase.

There are several ways of performing a classification of the training data. However the objective here is for a technique where this may be done automatically and transparently to the human analyst. A SOM is selected as the best classification approach as it uses unsupervised learning. Further, it has the ability to learn and organise information without being given correct outputs for the inputs.

A SOM network consists of two layers of nodes. Each output node is computed with the dot product of its weight vector and the input vector. The result will reflect the similarity between the two vectors. At the end of training, a SOM will make use of its own learning ability to arrange the available training data into different cluster. After a SOM has classified the training data, an LVQ, which also uses supervised learning, is used to fine-tune the classification process. LVQs are closely related to SOMs, but use the given classification information to define the class regions in the

input space. In this case, the SOM and LVQ will learn from the data and perform their own classification process. This is desirable to meet the objective of automated operations.

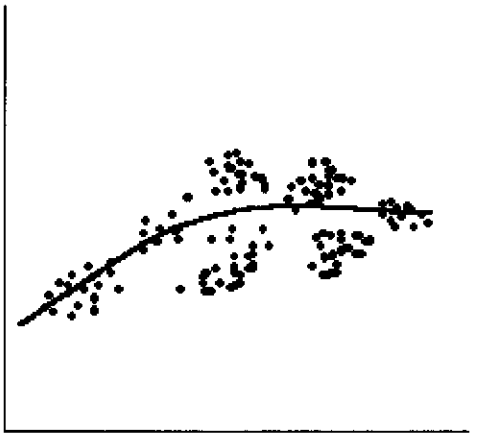


Figure 3.5: Function handle by one BPNN

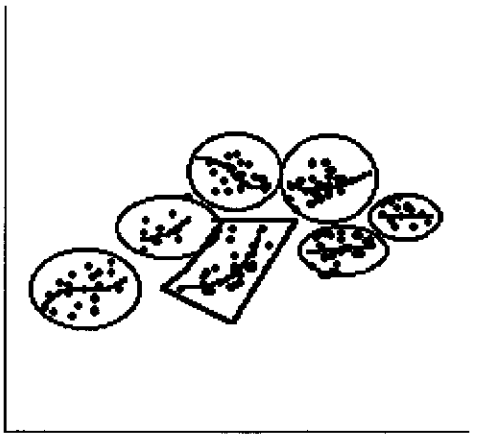


Figure 3.6: Functions handle by MNN

3.5 THE CLASSIFICATION PROCESS

3.5.1 APPLYING SOM AND LVQ ALGORITHMS TO LITHOLOGY CLASSIFICATIONS

Lithology classification involves the deduction of the principal rock matrix composition via data obtained from various well logging instruments. These instruments record the physical properties of earth at specific depths along the well bore. The measured quantities include resistivity, bulk density, gamma ray, neutron porosity, sonic waves and other parameters. The determined lithologies can then be used to answer a variety of questions such as thickness of a specific zone or depth of a particular formation. In addition to classification of lithologies, well logging data are also used to determine or predict petrophysical properties such as porosity, permeability and water saturation. This information is essential for the prediction of reservoir characteristics and subsequently the determination of reservoir production and related economic factors.

To illustrate the approach adopted, an example of 127 training samples comprising six input variables and three rock matrix outputs was used to derive representative results. The trained network was then applied to 378 test data values. In applying the SOM and LVQ algorithms to classification, a number of approaches were followed as listed below.

Approach 1: Apply the SOM algorithm to the input and output data separately and then compare the classification results. If the results are compatible, the

network for the input data can then be taken as the classification model.

Approach 2: Manually classify the input data and train the network with the LVQ algorithm based on the assumed classes.

Approach 3: Classify the output data using SOM algorithm. Unlike the manual approach in 2, the input data is now classified automatically according to their output characteristics. The LVQ algorithm is then applied. The resultant network will be the classification model for subsequent inputs.

In a preliminary examination, results from Approach 1 proved unsatisfactory. An ideal result is one that classifies both sets of input and output data into comparable classes. The matching accuracy between the two classification results in this case was found to be less than 50%. This indicates that the mapping between the two sets of data is not unique. Hence this approach was abandoned.

The second approach assumes that the input data are classified according to the input and/or output characteristics. This requires the experience and knowledge of a human analyst. That individual needs to observe the input data and determine the class patterns. With the manually classified input data, the LVQ algorithm is then applied to train the network. This is time consuming and inefficient. Further, the classification accuracy is very much dependent on the expertise of the personnel involved.

Since it is known that a certain relationship exists between the input vectors and the

characteristics within the output data, the third approach uses the output data as the basis for initial classification. The SOM is first applied to classify the output data. The classes obtained are then used to label the input vectors. The input vectors coupled with the output class labels are then applied to the LVQ algorithm. The process is summarised in the following steps.

Step 1: Normalise the input and output data.

Step 2: Determine the number of classes required and apply SOM algorithm to the output vectors.

Step 3: Label the input vectors according to output classifications from Step 2.

Step 4: Apply the LVQ algorithm to the normalised inputs and establish the network.

Once the network is trained, new input data can be classified by applying the normalised data to the network.

3.5.2 RESULTS AND DISCUSSIONS

The hardware platform employed to derive the results quoted was a PC 486-DX computer running at a clock speed of 33MHz. The SOM and LVQ programs are based on the SOM-PAK and LVQ-PAK obtained from Helsinki University of Technology (Kohonen et al., 1992).

Within the LVQ-PAK, the LVQ algorithms are implemented with a number of variations. They are the LVQ1, OLVQ1, LVQ2 and LVQ3 algorithms. Details of these algorithms are given in Kohonen et al. (1992). In this study, the OLVQ1 (optimised learning rate LVQ1) was used. A set of 127 training samples was used to examine the performance and accuracy of the SOM and LVQ algorithms. Results are graphed in Figure 3.7. Three output variables were used in the classification process as described in Step 2. They were rock matrices MATRIX-1, MATRIX-2 and MATRIX-3, which correspond to sandstone, limestone and dolomite respectively. Three output grid sizes for the SOM output layer were tested. The quantisation error and execution time due to these three configurations are tabulated in Table 3.3.

Table 3.3: Comparison of execution time and quantization errors

Output Grid Size	Time	Quantization Error
1 x 3	5 sec	0.17666
2 x 3	6 sec	0.14822
3 x 3	7 sec	0.11061

As will be noted, the quantization error reduces with an increasing number of classes. However, increasing the number of classes implies a reduction of the generalisation capability. On the other hand, if the number of classes is too low, the network provides poor discrimination. In this study, an output grid size of 2x3 was chosen as it presents better results as compared to the others.

In order to illustrate the characteristics of the rock matrix compositions, samples with the least quantization error in each class are shown in Table 3.4. In the same table,

the number of samples grouped under each class is also illustrated.

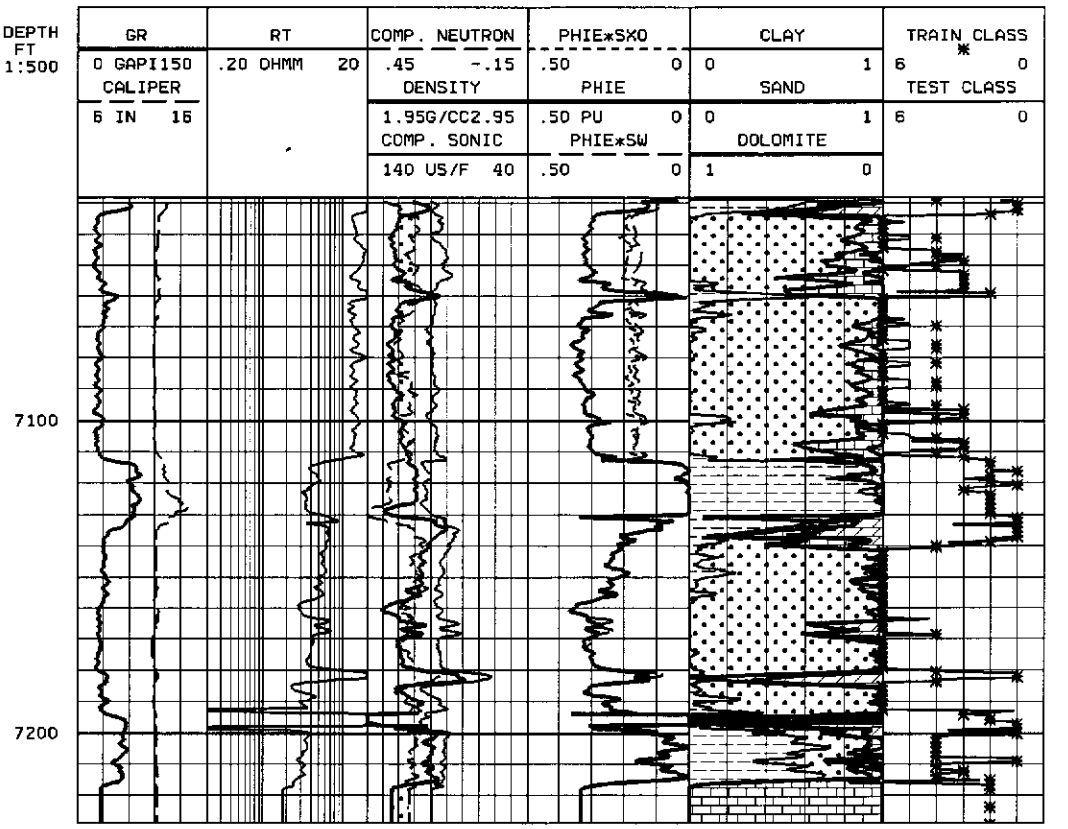


Figure 3.7: Graphical plot of data and results

Table 3.4: Samples of rock matrix composition

Depth	Class	Number	MAT-1	MAT -2	MAT-3
7095.0 ft	1	72	0.9417	0.0583	0
7099.5 ft	2	22	0.8272	0.1353	0.0502
7133.5 ft	3	33	0.5337	0.2802	0.2494

(a) 1x3 grid size

Depth	Class	Number	MAT-1	MAT -2	MAT-3
7137.0 ft	1	14	0.5082	0.2430	0.3333
7033.5 ft	2	16	0.5337	0.2802	0.2494
7054.5 ft	3	13	0.7209	0.2105	0.0918
7047.0 ft	4	14	0.8421	0.0487	0.1463
7077.0 ft	5	12	0.9345	0.0655	0
7102.5 ft	6	58	0.9668	0.0147	0.0248

(b) 2x3 grid size

Depth	Class	Number	MAT-1	MAT -2	MAT-3
7113.0 ft	1	12	0.3686	0.3489	0.1239
7053.0 ft	2	6	0.6011	0.3027	0.1290
7048.5 ft	3	9	0.7990	0.2010	0
7137.0 ft	4	6	0.5082	0.2430	0.3333
7180.5 ft	5	7	0.7390	0.1337	0.1705
7095.0 ft	6	11	0.9417	0.0583	0
7198.5 ft	7	14	0.6722	0.0459	0.3775
7045.5 ft	8	7	0.8819	0.0158	0.1371
7102.5 ft	9	55	0.9668	0.0148	0.0248

(c) 3x3 grid size

In Table 3.4(a), the composition of class 1 for the 1x3 grid-size is predominantly sandstone (MAT-1) and classes 2 and 3 show different ratios of limestone (MAT-2) and dolomite (MAT-3). In the 2x3 grid-size output as shown in Table 3.4(b), samples in classes 5 and 6 have similar characteristics to those shown in 1x3 grid-size class 1. However, class 6 has less sandstone and an increased proportion of limestone. This indicates that the SOM algorithm has subdivided class 1 in the 1x3 grid-size into two regions as shown in the case of the 2x3 grid-size. Another way to relate them is to consider the number of samples grouped under each class. In the 1x3 grid-size class 1, there are 72 samples. Classes 5 and 6 in the 2x3 grid-size have a total of 70 samples. A similar relationship is observed between class 2 in 1x3 and classes 3 and 4 in the 2x3 output. Also, class 3 in 1x3 can be considered to be equivalent to classes 1 and 4 in the 2x3 grid-size. Results from the 3x3 grid-size also exhibit similar characteristics when compared to the other outputs.

The input vectors of the training data are then labelled with the class numbers obtained from the SOM algorithm. Six input variables consisting of data measured from the neutron, density, resistivity, gamma ray, sonic and spontaneous potential instruments are used. They are applied to the OLVQ1 algorithm and the recognition accuracy is obtained. The results are summarised in Table 3.5.

Table 3.5: Recognition accuracy and number of samples in each class

Class	Recognition Accuracy
1	100%
2	100%
3	100%
Average Accuracy	100%

(a) 1x3 grid-size

Class	Recognition Accuracy
1	100%
2	100%
3	100%
4	100%
5	91.67%
6	100%
Average **Accuracy**	99.21%

(b) 2x3 grid- size

Class	Recognition Accuracy
1	100%
2	100%
3	100%
4	100%
5	85.71%
6	90.91%
7	92.86%
8	100%
9	100%
Average Accuracy	97.64%

(c) 3x3 grid- size

Table 3.6: Samples of test results

Depth(ft)	RHOB	NPHI	RT	GR	DT	SP	CLASS
7201.5	*0.8012*	*0.2446*	*0.0027*	*0.5842*	*0.6484*	*0.0613*	*4*
7202.0	0.7711	0.2533	0.0027	0.6006	0.6651	0.0254	4
7202.5	0.7606	0.2610	0.0028	0.6359	0.6512	0.0199	4
7203.0	*0.7835*	*0.2547*	*0.0029*	*0.7241*	*0.6624*	*0.0209*	*4*

(a) consistent lithofacies

Depth(ft)	RHOB	NPHI	RT	GR	DT	SP	CLASS
7198.5	*0.7466*	*0.3446*	*0.0016*	*0.5045*	*0.8711*	*0.0450*	*1*
7199.0	0.8045	0.2959	0.0025	0.5683	0.6579	0.0330	4
7199.5	0.8169	0.2341	0.0033	0.4625	0.5786	0.0330	1
7200.0	*0.8310*	*0.2636*	*0.0030*	*0.5477*	*0.5934*	*0.0311*	*1*

(b) additional class between known classes

Table 3.5 illustrates very high recognition accuracy. However, the value decreases with increasing grid-size. This is due to the existence of data in the overlapped boundaries between classes. As the number of classes is increased, the overlapped areas are also increased and some of these data may be classified incorrectly.

After the network is trained, a set of test data is applied to validate and check the accuracy of the network. A plot of the classification results is also shown in Figure 3.7. The test data were obtained from the same well at an interval of 0.5 ft whereas the training samples were recorded at an interval of 1.5 ft. Samples of the results are shown in Table 3.6. In this table, the entries shown in italics are the original training data while the rest are testing data. In most cases, the test data have shown consistency between two known classes. This is illustrated in Table 3.6(a). However, there are occasions where the network has identified the existence of additional lithofacies. In Table 3.6(b) a class 4 output is identified at 7199 ft. At close examination of the input characteristics, similarity between those at 7199 ft. and

those at 7201.5 ft is observed. This illustrates that the proposed approach in this example is providing a more refined picture of the lithology with higher resolution.

3.6 PREDICTION CAPABILITIES OF A MODULAR NEURAL NETWORK

3.6.1 THE APPROACH

The block diagram of the modular neural network is shown in Fig. 3.8. A number of BPNN networks corresponding to the number of classes obtained from SOM are trained. After the classification process, the data fed into the different BPNNs has similar characteristics. In this way, training of the BPNN takes a shorter time.

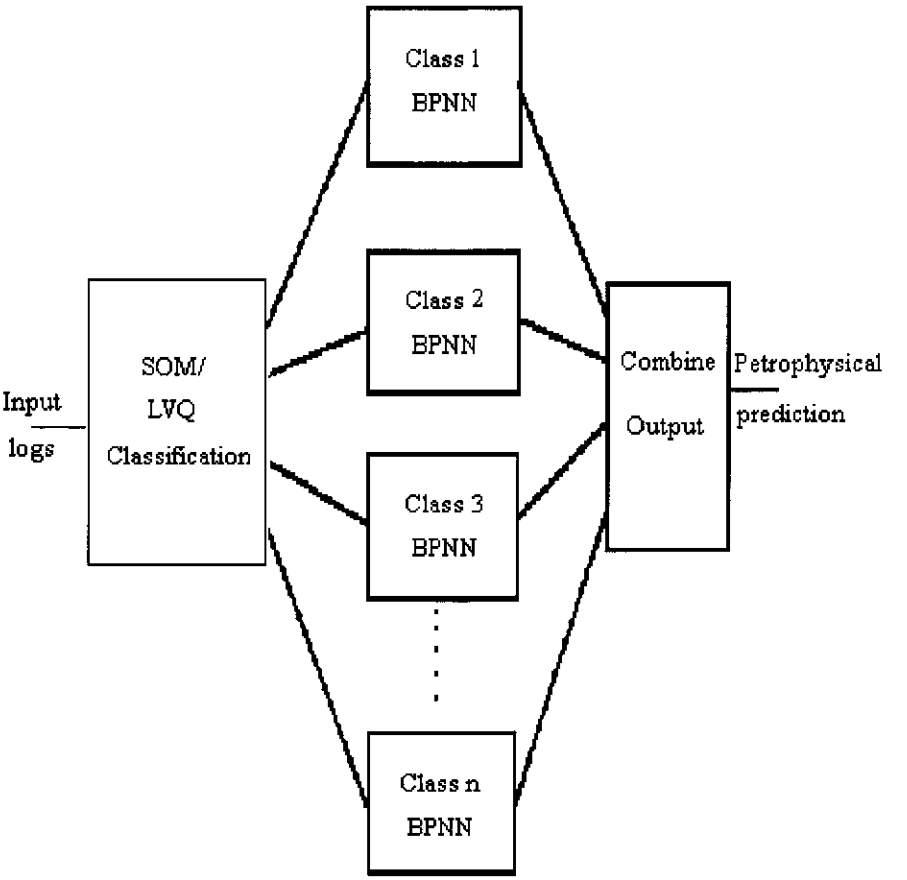


Figure 3.8: Block diagram of Modular Neural Network.

3.6.2 CASE RESULTS AND DISCUSSIONS

Some field data was used to test the procedure. A set of data that contained 127 input logs and corresponding output properties was used for training. Another set of 127 testing data was used to examine the performance of the modular neural network comprising a SOM, LVQ and BPNN. The results obtained were then compared to a conventional single BPNN network. The hardware platform used to derive all results was a PC Pentium-90 computer.

In this study, three output rock matrices were used to demonstrate the prediction ability of the proposed network. The rock matrices are (MAT-1) sandstone, (MAT-2) limestone and (MAT-3) dolomite. The input logs used were (RHOB) bulk density, (NPHI) neutron, (RT) uninvaded zone resistivity, (GR) gamma ray, (DT) sonic travel time and (SP) spontaneous potential.

The BPNN configuration chosen for the single network consisted of six input neurons, five hidden neurons and three output neurons. For the modular network, the SOM was initially used to classify the training data into nine different classes. Nine classes were found to be appropriate for classifying the data. As noted, the quantization error reduces with an increasing number of classes but also reduces the generalisation capability. On the other hand, if the number of classes is too low, the network provides poor discrimination. In this study, an output grid size of 3x3 was chosen as the better option.

These classes were attached to the input logs for the training of the LVQ network. The training data were also divided into the corresponding classes for training of individual BPNN networks. The BPNN configuration chosen for all the 9 sub-networks was the same as the single BPNN network.

Table 3.7 shows the results obtained from the modular network as compared to the results from the single network approach. As expected, the training time for the modular network was much shorter than the single network method. The overall accuracy of the modular network was also better based on the comparison between their mean square errors. The mean square error of the modular network was

calculated by taking the average of the mean square errors from the sub-networks. Figure 3.9 shows the graphical plot of the results generated from the single BPNN as compared to the actual core data. The modular network's output is shown in the graphical plot in Figure 3.10. From these figures, it can be observed that the modular network's output follows closely to the desired output core data. The correlation between the neural network's output and the desired core data are calculated by statistical method using the percent similarity coefficient. For single BPNN method, the percentage similarity for MAT-1, MAT-2 and MAT-3 are 92.4, 41.4 and 53.8 respectively. As for the modular neural network, the similarity is 98.7, 89.9 and 92.5 respectively. Again, these figures have given a clear indication that the modular neural network performs better than single BPNN.

Table 3.7 Comparison of Single BPNN and Modular Neural Network.

	Modular Network	Single Network
BPNN Configuration	9 x 6 input neurons 5 hidden neurons 3 output neurons	6 input neurons 5 hidden neurons 3 output neurons
Training Time		
Network 1:	72 sec	
Network 2:	58 sec	
Network 3:	16 sec	
Network 4:	50 sec	
Network 5:	7 sec	
Network 6:	28 sec	
Network 7:	44 sec	
Network 8:	49 sec	
Network 9:	98 sec	
Total:	**7 minutes**	**34 minutes**
Mean Square Error		
Network 1:	0.001872	
Network 2:	0.0001	
Network 3:	0.000082	
Network 4:	0.0001	
Network 5:	0.000099	
Network 6:	0.0001	
Network 7:	0.0001	
Network 8:	0.0001	
Network 9:	0.0018	
Average:	**0.00048**	**0.0297**

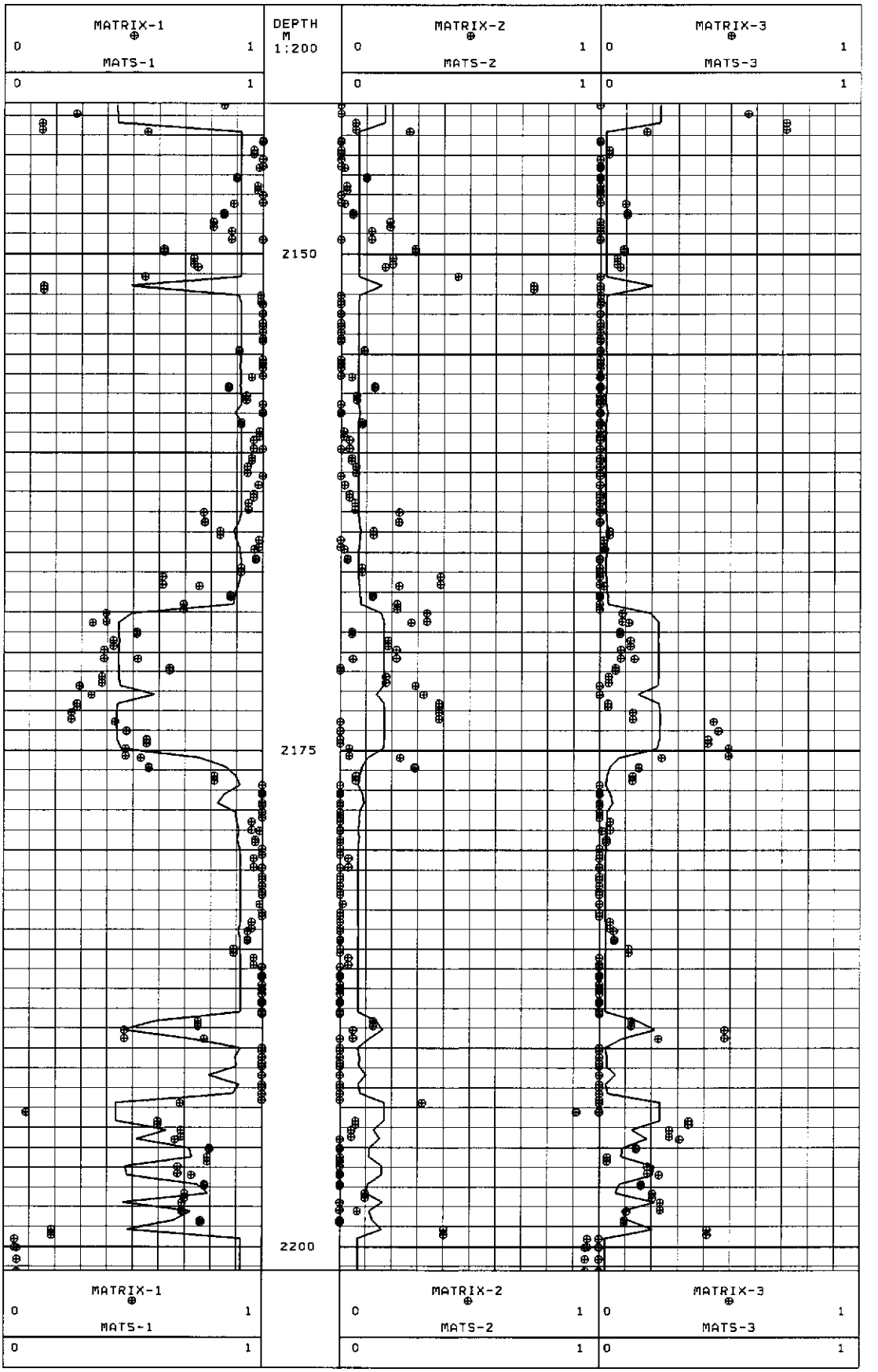


Figure 3.9: Single BPNN output compared to core data.

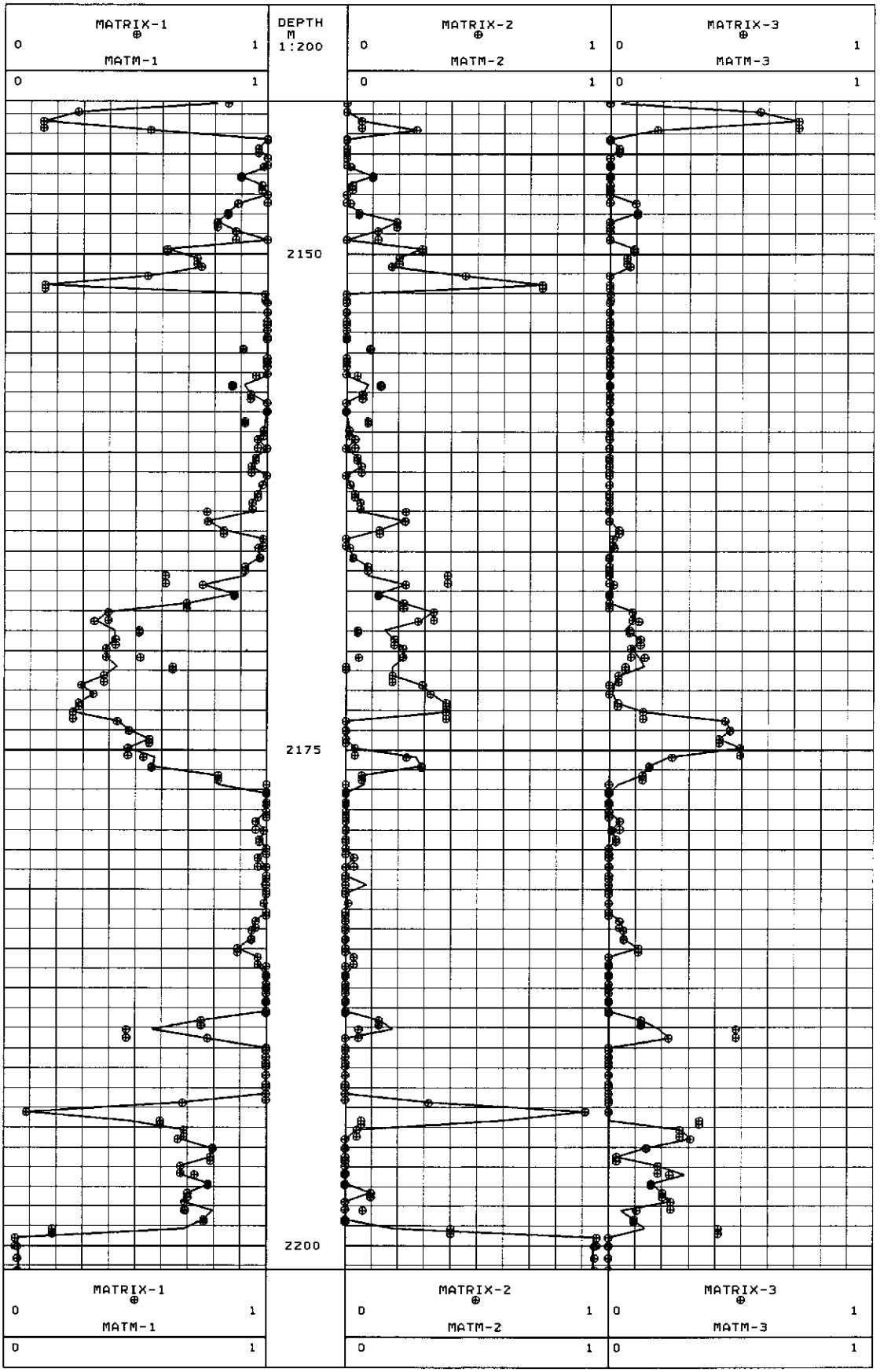


Figure 3.10: MNN output compared to core data.

3.7 ADVANTAGES GAINED BY USING THE MNN OVER AN ANN

A modular neural network that is capable of separating the complex function into different smaller module has been proposed and tested. The MNN not only perform the separation automatically, but also provide a more accurate and reliable alternative to the single BPNN approach. As the MNN makes use of several BPNNs, the learning process is more accurate in realising the underlying function. This is mainly due to the fact that each BPNN only learns from data that has similar characteristics.

First, SOM and LVQ algorithms have been used to classify the set of data from the input variables. After the classification process, a number of BPNNs are then used. The test results have shown that this approach to petrophysical prediction generates more accurate prediction compared to the conventional single BPNN approach. Results from the case study have also shown that the training time of this MNN is shorter. This approach could be used as an alternative method for the analysis tool in addition to the conventional methods.

CHAPTER 4:

GENERALISATION OF A BPNN

4.1 INTRODUCTION

As suggested in chapter 2, the generalisation capability of an ANN has a great effect on its prediction accuracy. This chapter will examine some of the problems in obtaining the best generalisation capability and new techniques are examined which achieve this.

Previously, discussion has centred on the use of a BPNN in function approximation. The most important feature of a BPNN is its ability to generalise. After a network has been trained with the available data, it is naturally desirable that the network provides reasonable performance for input data other than the training data set. However, without a systematic approach in design, that may not be the outcome.

Two primary conditions contribute to the failure to achieve a generalised network. First, the training data set does not possess all the characteristics of the population. Second, the test data used to set the generalisation function of the BPNN are statistically different from the training data.

Poor generalisation may also occur due to underfitting or overfitting. In the first case, the network is undertrained such that the system error remains high at the end of the training process. This may be due to insufficient iterations or the number of hidden units in the network configuration is too small. In these cases, the problem can be

overcome with an increased number of training iterations or to use an alternative network configuration. In the case of overfitting, this phenomenon (Weigend et. al, 1991) occurs when the network tries to fit all the data that may include substantial noise signals on the underlying function. Overfitting may also occur due to the use of large number of hidden units. One of the ways to avoid it is to use all the available training data while reducing the number of hidden units. However, a question arises: how many hidden units are appropriate to avoid overfitting or underfitting? Lawrence et. al (1996) have shown that a large number of hidden units can reduce the training and generalisation errors. However, this may lead to overfitting if no measure is taken to avoid this particular problem.

Most of the research in determining the best generalisation ability of an BPNN has focused on estimating the complexity of the network (Moody, 1992; Solla, 1993) and the network size (Yu, 1992; Baum and Haussler, 1989). Some effective approaches used to avoid underfitting and overfitting of the network include weight decay (Weigend et. al, 1991), early stopping (Wang et. al, 1994; Sarle, 1995) and utilising hint (Abu-Mostafa, 1990). Here, the early stopping approach (Nelson and Illingworth, 1991; Wang et. al, 1994) is investigated.

The early stopping approach has a number of advantages including that it is fast in determining the generalisation point and it can be applied successfully to networks in which the number of weights far exceeds the sample size. In this approach, available data set must be divided into a training data set and a validation data set. This, however, leads to the question of what proportion of available samples should be used as the validation data as well as how to extract or select these validation cases?

A straightforward technique is to divide the available data by trial and error. This is known as the spilt-sample validation approach (Weiss and Kulikowski, 1991). The error calculated on the validation set is used to determine when to stop the training process. In this case, the ability of providing a generalised BPNN is very much dependent on the validation set. Typically, several independent splits are performed and then the results averaged to obtain an overall estimate of the network performance. This method of validation is widely accepted, but it does suffer from the disadvantage of a long training time due to multiple training sessions on different splits. In addition to being sensitive to the specific method of splitting the data and the long training time, it also requires a large number of available data. Furthermore, the number of hidden units and the distribution of the data contribute to the effects on the generalisation ability of the network. In subsequent sections of this chapter, the problem of determining the training and validation data will be examined more fully. It is assumed that there are sufficient observed data.

In terms of the problem of data distribution, the validation error will start to rise when the BPNN tries to fit all the minority data. In effect, a generalised BPNN will treat those minority data as noise and they are not included in the underlying function. However, in cases where the training data are difficult and expensive to obtain, some of the minority data may be significant and should be included in the final generalisation curve of the BPNN. Under these situations, it is difficult to allow a BPNN to include these small numbers of significant training data and at the same time be able to reject those noisy data. An investigation into this problem is also presented within the chapter.

This chapter additionally analyses the factors that affect the generalisation capability of a BPNN from a statistical viewpoint. This enables a better understanding of them. These statistical insights are also used to provide a better understanding of the new proposed techniques, and to show their feasibility.

4.2 SPLIT-SAMPLE VALIDATION

Split-sample validation is the most commonly used method for estimating the generalisation capability of a BPNN using the early-stopping approach (Weiss and Kilikowski, 1991). A set of validation data that is not used in the training process is used to calculate the validation error. The validation error is found in the same way as the average training system error of the BPNN:

$$Ve = \frac{1}{2} P(\sum (Tp - Op)^2 \qquad (4.1)$$

where Ve = average validation error

P = no. of patterns

Tp = target patterns

Op = output patterns

The stopping point in this method is suggested to be the point when the validation error starts to rise as this suggests it is the point where the generalisation ability starts to degrade. Figure 4.1 shows a typical plot of the training and validation errors. When training starts, the errors for both data sets will normally reduce. After much training iteration, the validation error normally starts to rise although the training error may continue to fall. This suggests that the BPNN starts to overfit. The BPNN

training process can be stopped at this point, as further training will result in overfitting.

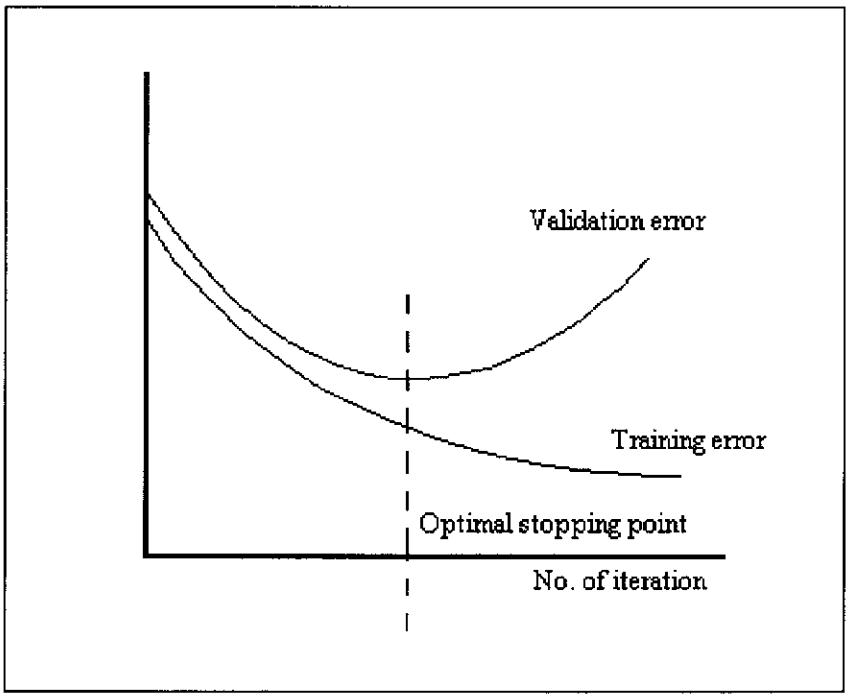


Figure 4.1: Typical plot of training and validation error

Using the early-stopping validation technique, the generalisation ability of a BPNN is highly dependent on the validation data set. Hence, the splitting method used is important. However, there are no rules to suggest the best splitting methods. Nevertheless, the validation data set should demonstrate two characteristics: (1) the validation set should be statistically close to the training set, and (2), the validation error should indicate the generalisation ability of the final BPNN as it is used as the stopping criteria for the training process. In order to address these two characteristics, a SOM data-splitting approach has been proposed and examined.

4.3 SOM DATA-SPLITTING

4.3.1 THE CONCEPT OF SOM DATA SPLITTING

If U is the universal sample space of all the cases of data to be processed by the network, then the training set TR should be statistically similar to U:-

$$s(TR) \subseteq s(U) \tag{4.2}$$

where s( ) indicates the statistical characteristics of a data set.

If s(TR) covers the complete sample space, the validation set (VA) and testing set (TE) should be statistically similar to the training set,. That is,

$$s(VA) \subseteq s(TR) \tag{4.3}$$

$$s(TE) \subseteq s(TR) \tag{4.4}$$

and with the condition: $VA \cap TE = \varnothing$.

However, if the conventional random approach of data splitting is used, this may result in a worst-case situation defined by the following equations.

$$s(TR) \subseteq s(U) \tag{4.5}$$

$$s(VA) \subseteq s(U) \tag{4.6}$$

$$s(TE) \subseteq s(U) \qquad (4.7)$$

and conditions:

$$TR \cap VA \cap TE = \varnothing ,$$

$$s(TR) \neq s(VA) \neq s(TE)$$

In this case, the statistical characteristics of the three data sets are mutually exclusive. The training set does not cover all the sample space, and the validation and testing sets will not be able to give a fair indication of the generalisation ability of the network. It is therefore essential to ensure the similarity of the statistical characteristics of these three data sets.

In SOM data-splitting, the available data are first classified into different clusters using unsupervised learning. If U is classified into C_1 to C_n clusters, then U can be written as:

$$U = \{ C_1 + C_2 + C_3 + \ldots . C_n \} \qquad (4.8)$$

If the training data set is selected from each one of the *n* clusters and some are left for testing and validation, then the conditions on equation (4.3) and (4.4) are satisfied. In this case, the training set will cover all the desired underlying cases. The validation set and testing set are subsets from the clusters from which the training set is selected.

In data splitting, after the SOM has classified the available data, a quantization error corresponding to each data point is generated. A number of splitting approaches on this classified data set can be adopted:-

1. *Lowest QE*

Select the data in each class which has the lowest quantization error and forms the training set. The remaining data are used as the validation set.

2. *Low-High QE*

Select all the data with the lowest and highest quantization error in each class and form the training set. The remaining data form the validation set.

3. *Mean QE*

The training set comprises of data from each class with the mean quantization error. Similar to above, the remaining data form the validation set.

With this SOM data-splitting approach, data from each class are selected for training or validation.

4.3.2 A CASE STUDY ON THE SOM DATA SPLITTING TECHNIQUE

In this case study, the problem of predicting petrophysical properties from well log data has been selected to test the proposed SOM data-splitting approach. Core data from five wells within a particular region are used. It was assumed that all these

wells exhibit similar petrophysical properties. Core data from four wells were used as training data. The core data in the fifth well were reserved as a testing set to verify the accuracy of the trained BPNN. There were a total of 85 training core data values and 32 test data values. A total of nine input logs were available and the target petrophysical property to be predicted was porosity. In this case, all the available input logs were assumed to be important. The network configuration selected for this study comprised of nine input nodes and one output node.

For comparison purposes, two other splitting approaches had been used. The splitting approaches were:

1. **Select one skip one**: Select the first core data as training data and the next one as validation data. Repeat this selection until the end of the set of core data.

2. **Block selection**: Select the first half of the available data set as training and the second half as the validation set.

A BPNN was trained and tested without the use of any validation set. The training process was stopped when the average system error was reduced to the minimum. This is referred to subsequently as Test 1 and it was used to compare the results obtained from subsequent networks trained with the data-splitting approaches.

Test 2A to Test 4B were based on the data-splitting methods without the use of SOM. Tests 2 and 3 were based on the "Select one skip one" approach described above. The differences between the two were the swapping of the test and validation

data sets. Test 4 was based on the "Block selection" approach. In each test, two stopping criteria had been used. Test A means that the training was stopped when the minimum error or maximum number of iterations was reached. Test B means that the training was stopped when the validation error started to rise. A summary of these tests is listed below.

Test 1: Train with all available core data and aims to reduce the average system error to minimum.

Test 2A: Use the "Select one skip one" approach and aims to reduce the system error to minimum.

Test 2B: Same as Test 2A, but stop training when validation error starts to rise.

Test 3A: Same as Test 2A but the training data and validation data sets are interchanged.

Test 3B: Same as Test 3A, but stop training when validation error starts to rise.

Test 4A: Using "Block selection" approach and aims to reduce the system error to minimum.

Test 4B: Same as Test 4A but stop training when validation error starts to rise.

The total number of training and validation data used in the above tests are shown in Table 4.1.

TABLE 4.1: Number of training and validation data for Test 1 to Test 4B.

	Training	Validation
Test 1	85	0
Test 2A & 2B	43	42
Test 3A & 3B	42	43
Test 4A & 4B	43	42

For SOM data-splitting, the 85 training core data were classified into predefined classes. The maps selected were 6-by-6 (36 classes), 7-by-7 (49 classes) and 8-by-8 (64 classes). These dimensions were chosen because it was intended to keep the number of training data between one-third to two-third of all the available data. After classification, quantization errors for each data were generated.

Based on the 6-by-6 output classes, several tests had been carried out. A description of these tests is as follows:-

Test 5A: Select one data from each class and two from those classes that had more data as the training set. The purpose of this selection was to maintain the same number of training and validation data as in Test 2A and Test 2B. The BPNN was trained to the minimum system error.

Test 5B: Same as Test 5B but stop training when the validation error started to rise.

Test 6A: Use Lowest QE approach and reduce the system error to a minimum.

Test 6B: Same as Test 6A but training was stopped when the validation error started to rise.

Test 7A: Low-High QE approach was used and the system error was reduced to a minimum.

Test 7B: Same as Test 7A but stop training when the validation error started to rise.

Test 8A: Use Mean QE approach and reduce the system error to a minimum.

Test 8B: Same as Test 8A but stop training when the validation error started to rise.

As for classifications based on 7-by-7 and 8-by-8 maps, only the Mean QE approach was used. This was because the number of data in each class had been reduced and there was no need to use more than one data point from each class. Tests 9A and 9B were carried out on the 7-by-7 class data. Finally, Tests 10A and 10B were performed on the core data classified into the 8-by-8 map.

Test 9A: Use Mean QE approach and reduce the system error to a minimum.

Test 9B: Same as Test 9A but stop training when the validation error started to rise.

Test 10A: Use Mean QE approach and reduce the system error to a minimum.

Test 10B: Same as Test 10A but stop training when the validation error started to rise.

The total numbers of the training and validation data used in all these SOM data-splitting methods are shown in Table 4.2.

TABLE 4.2: Number of training and validation data used from Test 5A to Test 10B.

	Training	Validation
Test 5A & 5B	43	42
Test 6A & 6B	30	55
Test 7A & 7B	53	32
Test 8A & 8B	30	55
Test 9A & 9B	38	47
Test 10A & 10B	42	43

4.3.3 A DISCUSSION OF THE STUDY RESULTS

The tests performed in this case study were carried out on a Pentium-90 computer. All the application software was developed using the C programming language. Having trained with data prepared from the tests mentioned in the previous section, the BPNNs were tested with the 32 core data in the fifth well for the prediction of porosity. During the training stage, all the tests were aimed to reduce the system error to 0.001 or stop after 50,000 iterations. The results obtained from these BPNNs were then compared with the core porosity values. Two statistical similarity and dissimilarity measures were calculated for comparison purposes (Kovach, 1993), they are:

Percent similarity coefficient (PERCENT):

$$PSC_{ij} = 100\frac{\sum_k \min(X_{ik}, X_{jk})}{\sum_k (X_{ik} \_ X_{jk})}$$

Euclidean distance (EUCLID):

$$ED_{ij} = (\sum_k (X_{ik} - X_{jk})^2)^{\frac{1}{2}}$$

where i and j represent the two data to be compared and k represents the pattern rows.

The results from Test 1 to Test 4B are shown in Table 4.3, and the results for various SOM data-splitting method are shown in Table 4.4. Test 4B could not be carried out because the validation error started to rise from the beginning of the training. This may suggest that the validation data and training data are statistically dissimilar.

TABLE 4.3: Results and training time for Test 1 to Test 4B.

TEST	PERCENT	EUCLID	Training Time
1	91.865	0.8	25 min
2A	89.727	1.046	10 min
2B	93.199	0.627	13 sec
3A	92.842	0.67	2.8 min
3B	91.17	0.723	4 sec
4A	85.369	1.299	6.4 min
4B	NIL	NIL	NIL

From Table 4.3, Test 1 gave a relatively good result of 91.9% of similarity. However, the training time was close to half an hour and the number of training data used was 85. The system error did not reach 0.001 and the test stopped at 50,000 iterations. For cases using the data-splitting approach, Test 2B gave the best result and the training time was 13 seconds. Test 3B also gave a result that is compatible to Test 1 and only 4 seconds were used. However, it can be observed that the split-sample validation based on the *select-one-leave-one* and the *block-select* methods could not guarantee a better result for early stopping. The table shows that the result from Test 3A is better than Test 3B and Test 1. Hence a better method is required.

From a practical viewpoint, this suggests that the user has to repeat the splitting process in order to find the best splitting arrangement. This is commonly done, but it can be very time-consuming, especially when there is a large amount of data. Some of the data may be statistically similar while others exhibit different characteristics. This will lead to grossly incorrect results. This phenomenon was demonstrated in the case of Tests 4A and 4B.

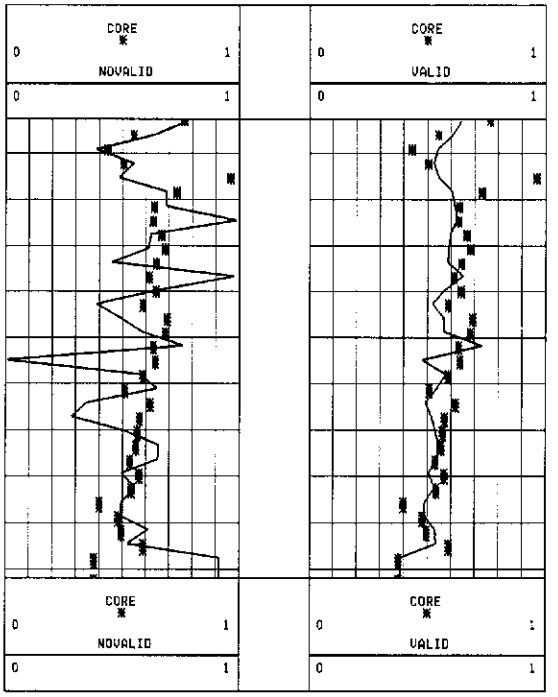
TABLE 4.4: Results and training time using SOM data-splitting

TEST	PERCENT	EUCLID	Training Time
5A	86.748	1.261	23.8 min
5B	93.731	0.591	1.3 min
6A	88.236	1.034	4.8 min
6B	91.998	0.662	3 sec
7A	89.941	0.956	24.5 min
7B	92.572	0.653	48 sec
8A	90.935	0.849	23 min
8B	91.678	0.693	4 sec
9A	89.11	1.22	23.7 min
9B	91.74	0.762	34 sec
10A	90.917	0.735	5.5 min
10B	93.33	0.637	5 sec

As shown in the results from Test 5 to Test 10, the SOM classification approach for data splitting performed better than Tests 1 to 4. Disregarding the ways that the training data were selected from each class, an overall improvement in the results can be observed. The best result is obtained from Test 5B with a percentage of similarity of 93.7% and a training time of 1.3 minutes. In Tests 6B and 8B, the results were 92% and 91.7% respectively. It should be noted that only 30 training data were used in both cases and the training time was less than 4 seconds. Test 10B used 42 training

data and it gives a result of 93.3%. The value is again better than the result from Test 2B above.

Although their similarity coefficients do vary slightly between different splitting methods, the results from the SOM data-splitting and early stopping approaches are constantly better than the others. It is important that the training data must include all the essential characteristics of the population and that statistically similar validation data be used for verification of the network's generalisation ability. Data obtained from the SOM approach fulfils these requirements. Another advantage is that the overall training time is greatly reduced, as it is not necessary to repeat and try different data-splitting processes. In order to illustrate the generalisation capability of the networks, Figure 4.2 shows a plot from Test 5 comparing the predicted porosity with the core porosity. Test 5A is the plot without early stopping while Test 5B shows the result of using the SOM approach. It can be observed that the predicted porosity with validation gives a better result compared to the one without validation.



Test 5A Test 5B

Figure 4.2: Comparing predicted porosity with core porosity.

Figure 4.3(a) and 4.3(b) are cross-plots of the trained network outputs with respect to the core training data also from Test 5. Figure 4.3(a) shows that the output from Test 5A without any validation gives better correlation between the training data and the network output. Figure 4.4(a) and 4.4(b) are cross-plots of the predicted outputs from Test 5A and 5B with respect to the testing core data in the fifth well. These data have not been presented to the network during the training or validation phases. It can be observed that overfitting has taken place in Test 5A as shown in Figure 4.4(a). Test 5A performed well in the training process as demonstrated in Figure 4.3(a) but failed to predict reasonably for data that were not included in the training process. On the other hand, the SOM data-splitting method provided better results as illustrated in Figure 4.4(b). Similar results were also observed from the other tests (Test 6 to Test 10).

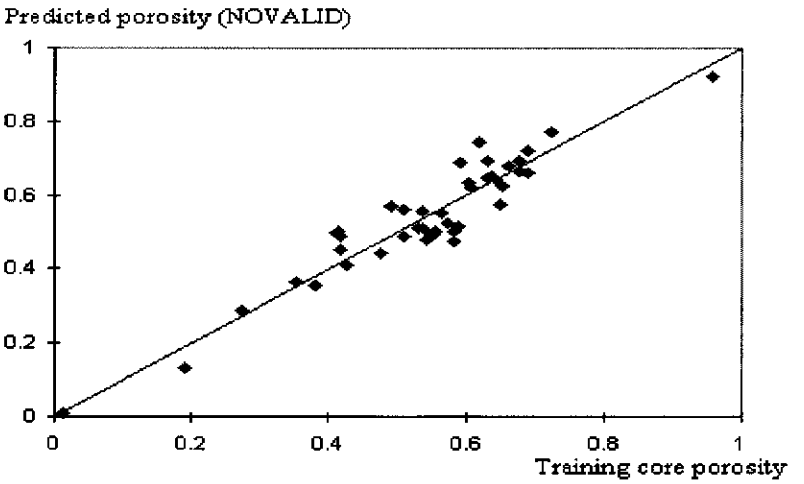


Figure 4.3(a): Cross-plot of predicted porosity Vs training core data from Test 5A without early stopping validation

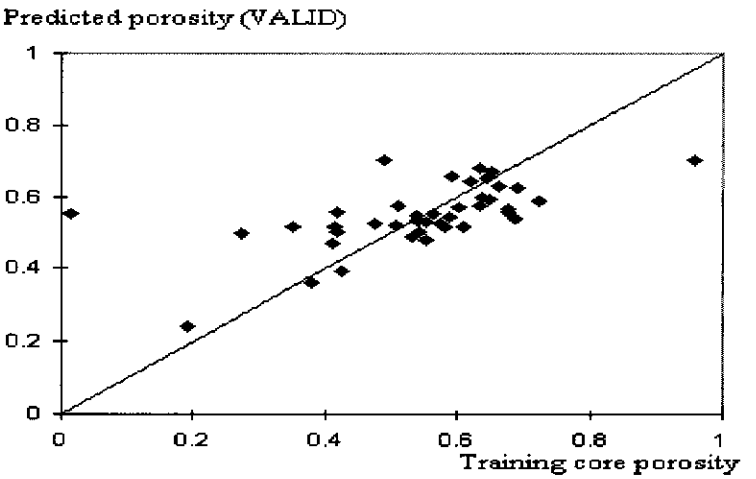


Figure 4.3(b): Cross-plot of predicted porosity Vs training core data from Test 5B with early stopping validation

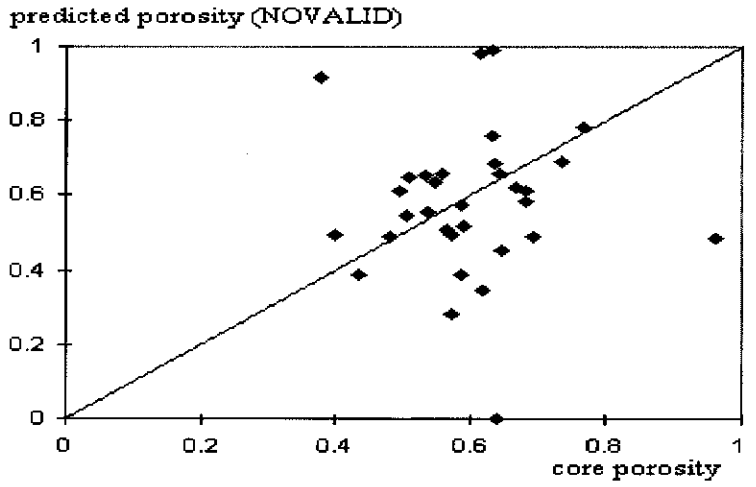


Figure 4.4(a): Cross-plot of predicted porosity Vs testing core data from 5th well in Test 5A without early stopping validation.

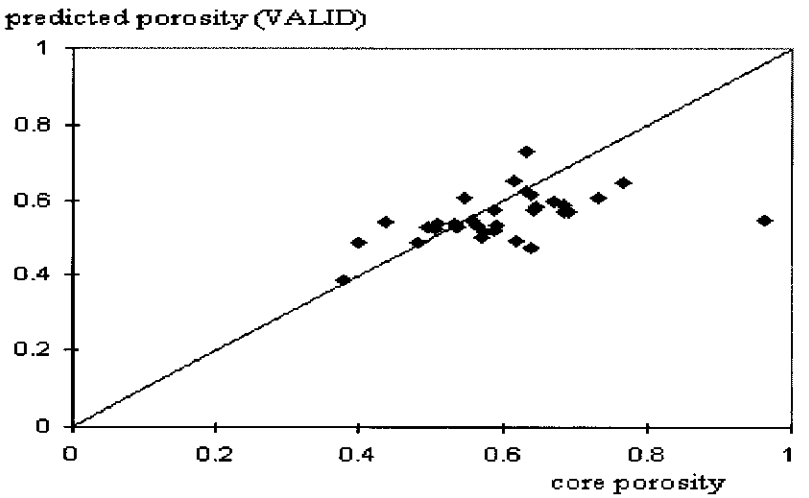


Figure 4.4(b): Cross-plot of predicted porosity Vs testing core data from 5th well in Test 5B with early stopping validation.

4.4 THE NUMBER OF HIDDEN UNITS AND SOM DATA-SPLITTING

4.4.1 USING A LARGE NUMBER OF HIDDEN UNITS

Underfitting normally occurs due to too small a number of hidden units or too little training iteration. This problem should always be avoided by using more training iterations and a larger number of hidden units. However, a large number of hidden units may result in overfitting as the network will try to fit all the data including any noise (Smith, 1993; Weigend et. al, 1991). A commonly used approach to avoid this is the early stopping validation approach mentioned earlier. The proposed SOM data-splitting approach also allows a confident selection of validation set from the available data.

This section compares the performance of networks with different numbers of hidden units using the SOM data-splitting and early stopping validation approaches to enable the BPNN will reach the best generalisation point.

4.4.2 COMPARING A DIFFERENT NUMBER OF HIDDEN UNITS

A problem of well log data analysis was again used to illustrate the approach. A set of typical well-log data comprising 303 core data measurements was used. This set of data consists of 9 input logs (PEF, RHOB, NPHI, CALI, RT, RXO, GR, DT and SP). Typical petrophysical properties to be determined are porosity, permeability, volume of clay (VCL) and a number of other parameters. In this study, results of VCL were examined. The available 303 core data were first classified using the unsupervised SOM method. Testing and validation data sets were then selected from

each cluster. If a cluster contains only one data point, it was selected as the training data. This is to ensure that the training of BPNN covers all possible features. Table 4.5 shows the number of data in each set from the SOM data-splitting approach.

Table 4.5: Number of training, validation and testing data.

Set	No. of data
Training	117
Validation	77
Testing	109

The number of hidden units varied from very small to 8 times the number of training cases. Table 4.6 shows several tests and the corresponding number of hidden units being used.

Table 4.6: Number of hidden units in each test.

Test	No. of hidden units
1	3
2	5
3	18
4	52
5	82

The BPNN was trained with the training data set and the validation error was calculated for every cycle of the training process. The error measure used in this case was as follows:

$$ERROR = \frac{0.5(\sum (Tp - Op)^2)}{P}$$

where Tp = target pattern

Op = output pattern

P = no. of patterns

As the training and validation errors may oscillate during training, the process was allowed to run until the network converged and the validation error rose steadily indicating the network was overfitted. At this point, training was stopped and the network configuration with the lowest validation error was used. After the training and validation processes had been completed, the testing data set was then used to generate an unbiased estimation of the BPNN's generalisation ability.

4.4.3 DISCUSSION OF THE CASE STUDY RESULTS

The BPNNs from Tests 1 to 5 in Table 4.6 were trained and stopped at the lowest validation error point. The error formula, given above was used to calculate the errors. Table 4.7 shows the training and validation errors of five networks.

Table 4.7: Training and Validation Error.

Test	Training error	Validation error
1	0.00269	0.00404
2	0.00198	0.00557
3	0.00200	0.00548
4	0.00213	0.00340
5	0.00191	0.00362

In order to assess the generalisation ability, the training and the validation data set were combined to form Data Set P, which was used previously during the training process. The Testing Data Set T, was one that had not been applied to the network during training. Results from these two data sets were used to compare the performance of the networks. The results are tabulated in Table 4.8.

Table 4.8: Comparison of Errors for different BPNNs.

Test	No. of Hidden units	Error from Data Set P	Error from Data Set T
1	3	0.00345	0.00591
2	5	0.00362	0.00579
3	18	0.00359	0.00571
4	52	0.00275	0.00431
5	82	0.00262	0.00392

From these results, errors due to Data Set P in Table 4.8 were dependent on both training and validation errors in Table 4.7. The value was approximately the average of the training and validation errors. In Test 2, although the training error is the lowest, it has a high validation error. The overall error therefore became the largest compared to the other test cases. This shows that the validation error has a significant effect in determining the overall error. It also suggests that using the training or validation error alone is not a good estimation of the generalisation ability of the BPNN. The use of an unbiased testing set such as Data Set T, which was not used previously in the training process, should be used to estimate the generalisation ability. From Table 4.8, it is observed that the overall error decreases as the number of hidden units increases. Test 5 with the largest number of hidden units gives the lowest errors from both data sets P and T. This suggests that when the number of hidden units increases, it can fit the underlying function better and at the same time avoids underfitting. However, it may be argued that an over-sized network may result in overfitting. Using the early-stopping validation approach to locate the best generalisation point solves this problem, and using the SOM data-splitting approach ensures similarity of the data sets.

Figures 4.5 and 4.6 are cross-plots of the predicted outputs generated by the BPNN in Test 1 as compared to core data. Figure 4.5 shows the comparison with the core data set P while Figure 4.6 is the comparison with core data set T. Figures 4.7 and 4.8 show similar cross-plots from the outputs generated from the BPNN in Test 5.

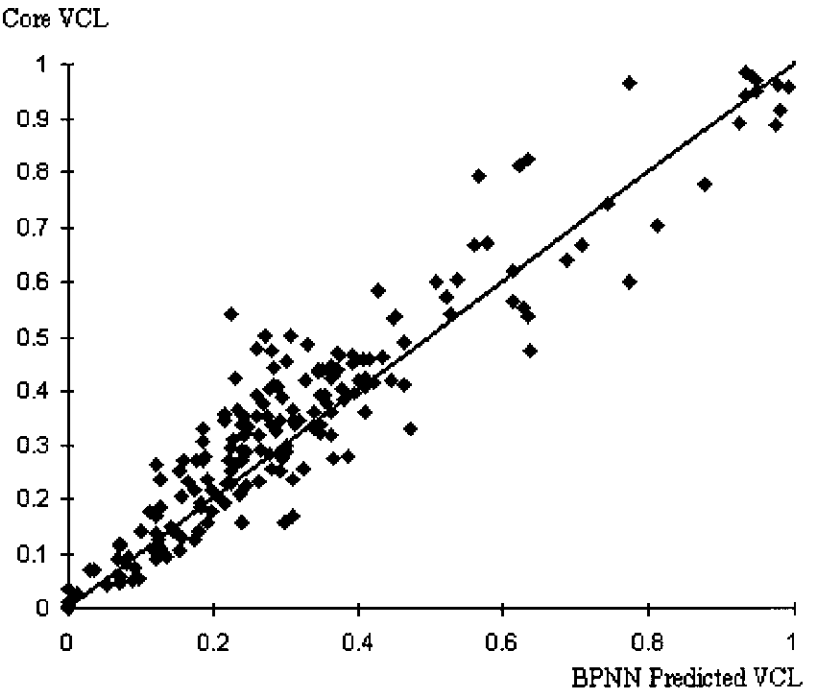


Figure 4.5: Cross-plot of core data set **P** vs BPNN predicted output from Test 1

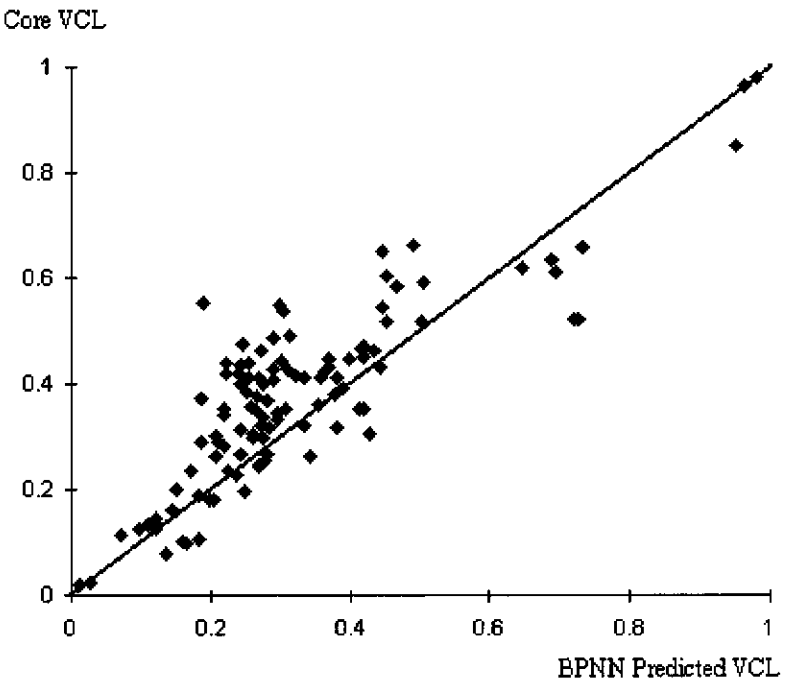


Figure 4.6: Cross-plot of core data set **T** vs BPNN predicted output from Test 1

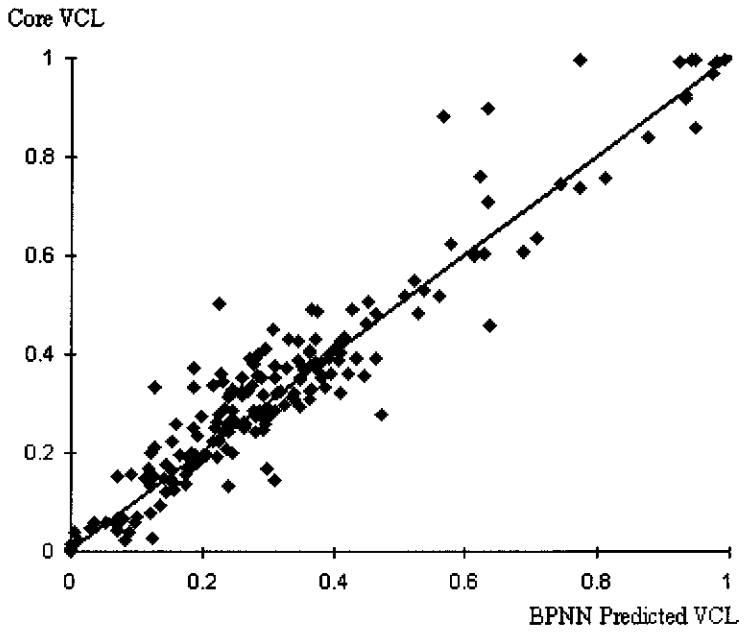


Figure 4.7: Cross-plot of core data set **P** vs BPNN predicted output from Test 5

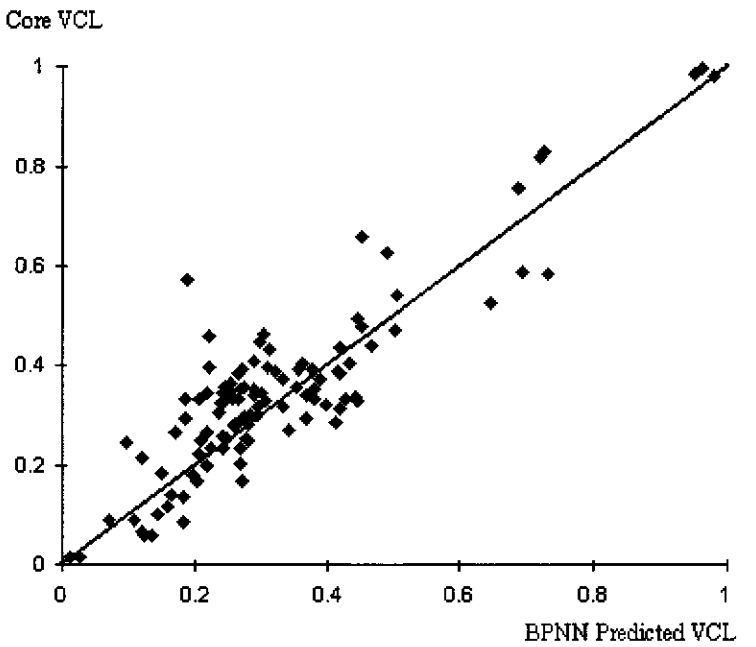


Figure 4.8: Cross-plot of core data set **T** vs BPNN predicted output from Test 5

4.5 GENERALISATION BIAS OF BPNN AND SOM DATA-SPLITTING

When the SOM data-splitting early-stopping validation method is used in the training process for the prevention of overfitting, the system errors due to the training and validation data are calculated in each iteration. In the SOM data-splitting approach as discussed in section 4.2, the training and validation sets are split in such a way that both sets are statistically similar. This can ensure that the training set covers the whole sample space of the available data. At the same time, the validation set will give a better indication of the generalisation ability of the BPNN. Minority data normally falls into the training set after the SOM data splitting. This will be deemed as noise by the validation error in the training of a BPNN.

In the early stopping validation approach, the training set is used to train the BPNN and the validation set is used to guide the generalisation ability of the BPNN. Since the training and validation data are different, it is assumed that the generalisation point is reached when the validation error starts to rise. The network will start to memorise all the training data and overfit from henceforth. At this point, the network is characterised by a generalisation curve that provides the best fit for both the training and validation data.

If there are training data that are few in number and located outside of the generalisation curve, they will be treated as noise and be ignored. Figure 4.9 illustrates an example with three points that are treated as noise and are excluded by the generalisation curve. Assuming that the point highlighted with a circle in Figure 4.9 is deemed by the user as a significant data and it is desirable to be included in the

final generalisation curve as shown in Figure 4.10. Using the data splitting early stopping validation approach, it is not possible for the network to recognise that particular data and to include it in the generalisation curve while treating the other two points as noise. In this case, it is known that there is a bias in the generalisation ability of the BPNN. Normally, the BPNN will bias towards the majority of the training data and treat the minority as noise. However, the question need to be resolved of whether those few minorities are significant, and if so, how can they be included in the final generalisation curve of the BPNN?

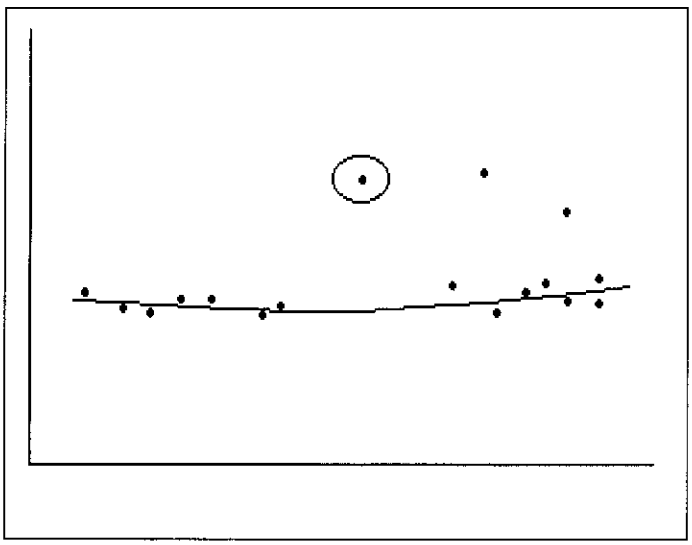


Figure 4.9: Example of a generalisation curve.

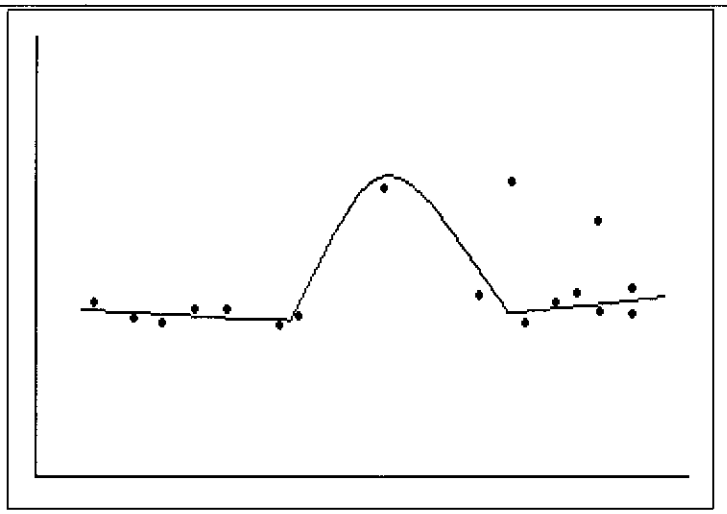


Figure 4.10: Desirable generalisation curve.

4.6 INTERACTIVE REINFORCEMENT TRAINING

4.6.1 DATA BIAS IN BPNN TRAINING

In examining the validation approach and the characteristics of the BPNN training, it can be deduced that:

(1) in order to obtain a low bias and low variance in BPNN training, a large number of training data that contributes to the actual generalisation curve need to be used;

(2) the significant minority data will only appear in the training set, so that when the validation set is used to stop the training, it will normally bias towards the majority data and stop at the point where it predict the best with majority data.

This deduction may be used to solve the problem of how to include, if necessary, the significant minority data in the final generalisation curve while at the same time rejecting the noise present in the training data set. There are two ways this can be done:

(1) Obtain more data that contributes to the characteristics of the minority data.

(2) Manipulate the training and validation process of the BPNN such that it can include those significant minorities.

The first approach is usually not practical as the training data is difficult and expensive to obtain. However, the second approach may be feasible.

An interactive reinforcement training scheme is proposed to address the second approach. The steps involved in this approach are as follow:

(1) Identify the minority data in the data set and examine them.

(2) Pick up the known significant data point that is in the minority of the data set.

(3) Duplicate that significant data point in the training set and validation set.

(4) Train and validate the BPNN with all data including these reinforcement data points.

(5) Check the trained BPNN to see that it can accommodate the characteristics of the minority significant data.

(6) If not, reinforce that data point again by duplicating more data in the training set and validation set.

(7) Re-do Steps (4).

In this proposed interactive reinforcement training approach, the user can modify the BPNN generalisation curve easily by just duplicating the significant data point. This will force the BPNN to recognise that point in the training and the validation sets.

4.6.2 CASE STUDY

A case study using the BPNN to predict the porosity from well log data in petroleum industries was used in this investigation. A typical well with 41 core data was used. After the SOM data splitting, the available data were divided into training and validation set. Twenty points were used for training while the remainder were used for validation. In this well, the majority of the core data had been clustered around the same region with exception of three data points. This suggests that these three points could be noise. Under normal situation, they will appear only in the training set with SOM data splitting. Hence, the generalisation curve will leave them out when the validation set is used to stop the training.

However, if one of the three points is recognised to be significant, it is difficult to modify the generalisation curve by using normal BPNN training and validation. With interactive reinforcement training, it is possible to incorporate that point easily.

Some tests were carried out to illustrate the effects of the interactive reinforcement training approach. The numbers of the duplicated significant data used in each test are as tabulated in Table 4.9.

Test 1 used normal BPNN training without any reinforcement involved in the training process. Tests 2, 3 and 6 had only reinforcement in the validation set. Tests 4 and 7 had only reinforcement in the training set. In Tests 5 and 8, both the training and validation sets were reinforced by the significant point.

Table 4.9: Number of reinforced data in the test cases.

Test	No. of reinforced significant data in training set	No. of reinforced significant data in validation set
1	0	0
2	0	1
3	0	2
4	2	0
5	2	2
6	0	5
7	5	0
8	5	5

4.6.3 CASE RESULTS AND DISCUSSIONS

The output plots of all the tests are shown in Figure 4.11. The predicted outputs from the BPNN are shown in lines and the core data are shown in asterisks on the plot. T1 corresponds to the results from Test 1, T2 corresponds to the results of Test 2 and so on.

The result plotted for T1 are the predicted output when no interactive reinforcement training is involved, which is also the normal training procedure used in training and validating the BPNN. In this case, it can be observed that the BPNN generalised to the majority training samples and left out the three that are different from the generalising curve. In this situation, all the three outliners are treated as noise by the BPNN. However, if upon inspection of this plot the log analyst was sure that the lowest outliner data point at around 740 metres is significant data, that data point could be reinforced and the BPNN re-trained. Results from Test 2 to Test 8 illustrate the modified generalised curve under each condition. When more than two data points were reinforced in the validation set as in the cases of Tests 3, 5, 6 and 8, the BPNN generalisation function moved towards the desired data point. This suggests that the BPNN generalisation curve had began to incorporate that significant point. In effect, the generalisation curve had been modified towards the truth generalisation point. It can also be observed that the highest outlier point around 730 metres has similar characteristics as that of the reinforcement point.

This test also indicates that the validation set plays an important role in deciding the bias of the BPNN generalisation curve. In Test 4 and 7, although that data point had

been duplicated two or more times in the training set but not in the validation set, the BPNN is still unable to recognise it as a significant point as shown in Figure 4.11.

A final Test was performed to include the other two data points around 730 metres using the proposed interactive reinforcement training. The resulting plot is shown in Figure 4.12. In this case, the BPNN recognised all three points as significant data and included them in the final BPNN generalisation curve.

The case study has shown that an interactive reinforcement training approach has successfully led the BPNN to include those data that would otherwise be ignored. It has also shown that the validation set has a great effect in determining the bias of the BPNN generalisation curve.

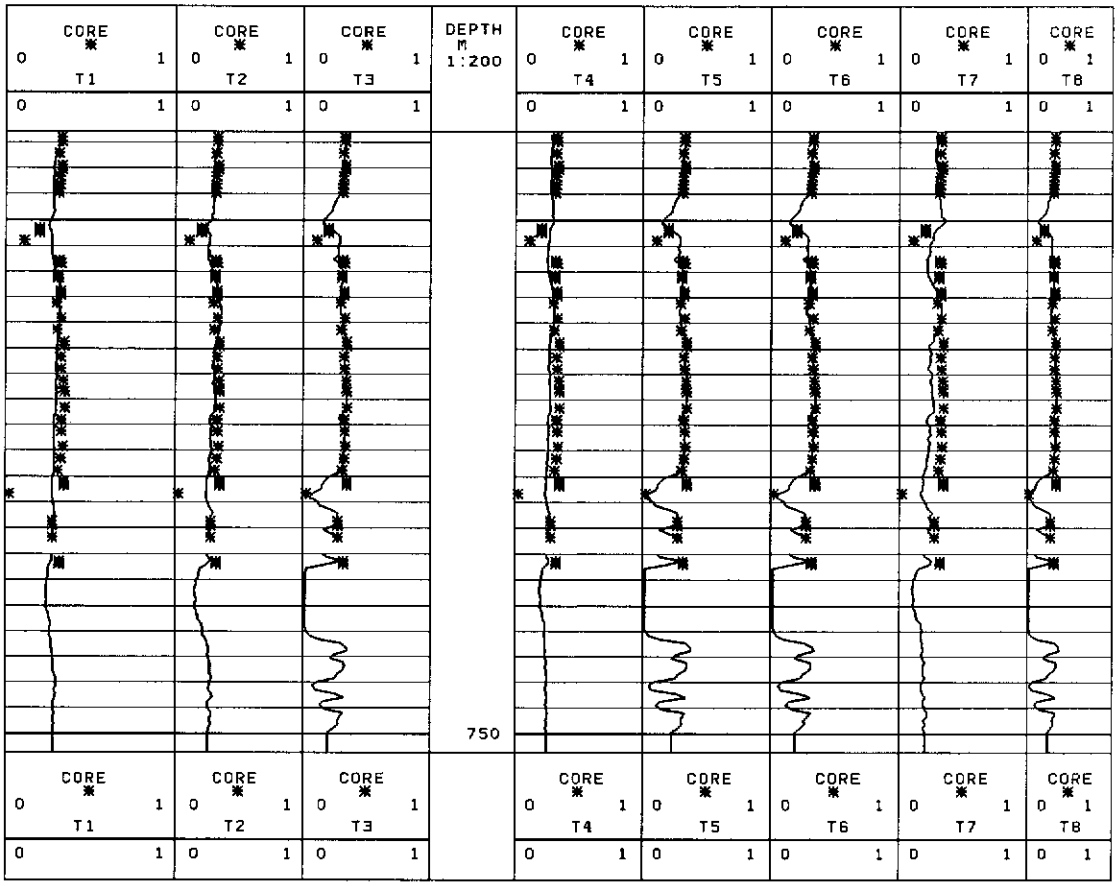


Figure 4.11: Output plots from eight test cases.

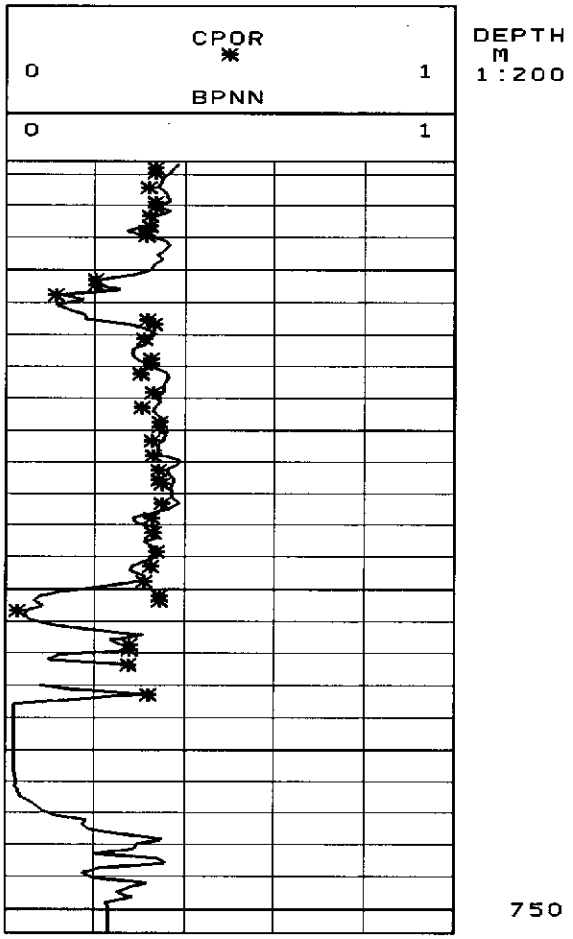


Figure 4.12: Output plot of test case with all three points reinforced.

4.7 TECHNIQUES TO ENSURE GENERALISATION

The SOM data-splitting and early stopping validation approach to determine the generalisation ability of a BPNN has been shown to be reliable. An investigation of the use of SOM as a data-splitting approach for the selection of training and validation data sets has also been reported. These data sets are used to train a BPNN based on a split-sample validation early stopping method. The results derived show that the use of the SOM approach is consistent in providing a good generalised network and the training time is reduced while avoiding the overfitting problem. The

SOM has also ensured that the training data set has enough information to include the underlying function, as well as the generation of a statistically similar validation set.

This is useful in the application of split-sample validation and early stopping for BPNN training. The effect of the number of hidden units on the overall performance has also been investigated in this chapter. It has shown that the overall performance of the BPNN is better with more hidden units. This is because when the number of hidden units increases, it can fit the underlying function better and avoiding underfitting. However, an over-sized network may result in overfitting. This problem is now solved by using this proposed SOM data-splitting and early stopping validation approach to locate the best generalisation point. With this approach, it provides a fast and reliable splitting criterion and prevents the BPNN from underfitting or overfitting when large numbers of hidden units are used.

This chapter has also shown that the generalisation curve may be modified using interactive reinforcement training of the BPNN. In this approach, the validation can still effectively prevent the BPNN from overfitting the noise. At the same time includes the significant minority data in the final generalisation curve of the BPNN. The approach incorporates inputs from a human for the identification of significant training points that may have been missed out due to the generalisation characteristics of the network. This interactive reinforcement training approach is useful when the number of training data are few and difficult to obtain due to high cost involved. It provides a means to allow a user interactively modify the network

characteristics without the need of alternating the weights within the network which are difficult to comprehend.

Now that the generalisation problems in a BPNN have been examined from a network viewpoint, the rest of this chapter examines the problems of generalisations via statistical analysis. This enables the generalisation of a BPNN to be better understood.

4.8 THE RELATIONSHIPS BETWEEN ANNs AND STATISTICS

In recent years, the relationships and overlaps between the fields of neural networks and statistical methods have been explored (Sarle, 1994; Cheng and Titterington, 1994; Ripley, 1993). As statistical methods are mainly concerned with data analysis, it may seem that they have little connections with neural networks which were originally developed to model biological systems. However, in terms of applications and characteristics, there are considerable similarities between these techniques. For example, the feedforward BPNN is similar to projection pursuit regression, the Hebbian neural network is similar to principal component analysis, and the Kohonen net is similar to k-means cluster analysis (Sarle, 1994; Cheng and Titterington, 1994; Ripley, 1993). Those without any similarities with statistical techniques are the Kohonen Self-organising Map and the Reinforcement learning net.

Although there are areas of overlap between the two fields of study, there are distinctive research objectives in each discipline. Neural network researchers are trying to design machine intelligence with an ability to adapt and learn. Most likely,

the network is treated like a black box that requires minimum human intervention, and is used to provide behaviour of "data in and prediction out". It gives an impression that anybody without experience should be able to use neural network tools with confidence based on the automatic learning characteristics.

On the other hand, statisticians usually depend on human understanding of the problem under study before designing any estimation model. They then generate hypotheses, test assumptions, and many other parameters to help them to understand the proposed model. From these different objectives, it will be useful if both of the statistical and the neural network disciplines are used hand in hand.

As statisticians have done much research in the field of data analysis over the past few decades, conceptual foundations and analysis techniques are already well established. It would be very useful to employ statistical analysis techniques to help in designing better neural network systems. As the original objective of developing neural networks was to model the way a human brain learns and functions, the notions of learning, self-organising, dynamics and field theory may provide inspiration for future study. The purpose of this part of the thesis is to perform a statistical analysis on the important issue of the BPNN's generalisation ability, so as to provide a better understanding of the factors that affect the generalisation capability of the BPNN.

4.9 STATISTICAL ANALYSIS OF A BPNN

The majority of Artificial Neural Network (ANN) applications can be categorised under two main headings: classification and function approximation. In function approximation, the BPNN's are similar and comparable to non-parametric estimators (White, 1989; German et. al, 1992) in statistics. The objective is to build a model to represent the relationship between the input (independent variable) x and the target (dependent variable) y without any assumed prior parameters. Given that the input vector X and the target vector Y, expression (4.9) can be used to describe the relationship:

$$Y = g(X) \tag{4.9}$$

When obtaining the training set (observations), there will be some environmental factors that affect the measurements. Therefore it is not possible to have an exact function, $g(\ )$, that describes the relationship between X and Y. However, a probabilistic relationship governed by a joint probability law $P(\nu)$ can be used to describe the relative frequency of occurrence of vector pair (X_n, Y_n) for n training set. The joint probability law $P(\nu)$ can be further separated into an environmental probability law $P(\mu)$ and a conditional probability law $P(\gamma)$. For notation expression, the probability law is expressed as:

$$P(\nu) = P(\mu)P(\gamma) \tag{4.10}$$

The environmental probability law $P(\mu)$ describes the occurrence of the input X. The conditional probability law $P(\gamma)$ describes the occurrence of the output Y based on

the given input X. A vector pair (X, Y) is considered as noise if X does not follow the environmental probability law $P(\mu)$, or the output Y based on the given X does not follow the conditional probability law $P(\gamma)$.

From (4.9), the relationship $g(X)$ based on the available training set can be assumed to be analogous to the conditional probability law $P(\gamma)$. Therefore, it is the role of γ that the BPNN is performing. It can also be denoted as $E(Y|X)$ as the Expectation of Y given X. Therefore:

$$g(X) = E(Y \mid X) \tag{4.11}$$

In a BPNN, $g(X)$ is not always obtained directly from the training set (X_n, Y_n). It has to undergo certain training (estimation) process in realising the best $g(X)$. In a BPNN, the best $g(X)$ model is directly related to the internal weights W, which can be expressed as:

$$g(X) \approx f(X, W^*) \tag{4.12}$$

where W^* denotes the set of the weights giving the best estimation

$f()$ is the estimating function of the network.

From the above condition and taking error into account, equation (4.9) is therefore:

$$Y = f(X, W^*) + \theta \tag{4.13}$$

where θ denotes the error.

The output vector (predicted value), O will be:

$$O = f(X,W) \tag{4.14}$$

To find the best weights W^* so as to minimise the error function θ, a BPNN makes use of the error backpropagation learning algorithm (Rumelhart et. al, 1986) to perform the mean square errors minimisation process, $\sum_{i=1}^{n}[Y - f(X,W)]^2$, or $\sum_{i=1}^{n}[Y - O]^2$. As the prediction performance of the BPNN is very much dependent on the weights W, the expected performance functions $\lambda(w)$ could be expressed as:

$$\begin{aligned}
&\lambda(w) \\
&= E([Y - O]^2) \\
&= E([Y - E(Y \mid X) + E(Y \mid X) - O]^2) \\
&= E([Y - E(Y \mid X)]^2) + E([E(Y \mid X) - O]^2) + 2E([Y - E(Y \mid X)][E(Y \mid X) - O]) \\
&= E([Y - E(Y \mid X)]^2) + E([E(Y \mid X) - O]^2)
\end{aligned}$$

As mean square error (MSE) combines the bias and variance into one measures (German et. al, 1992; Wadsworth, 1990). The above expression can then be separated into bias and variance term using the relationship of MSE = bias^2 + variance:

$$\text{BIAS} \quad = E(Y \mid X) - O \qquad \text{or} \qquad = E(Y \mid X) - f(X,W) \tag{4.15}$$

$$\text{VARIANCE} \quad = E([Y - E(Y \mid X)]^2) \tag{4.16}$$

Hence, the set of best weights *(W*)* which minimises the prediction error is effectively dependent on the bias and variance of the training set as demonstrated in the above analysis. Based on these parameters, an indication of the generalisation ability of the network can be derived as shown in the next section.

4.10 STATISTICAL ANALYSIS OF GENERALISATION CAPABILITY

The generalisation ability of the BPNN is the most important feature in most practical applications. It is a factor used to measure how close is the final model $f(X,W^*)$ to the expected model $E(Y|X)$. As the realisation of the best-fit model is dependent on the available training data, it is also regarded as a measure on how good the BPNN can provide reasonable prediction from 'unseen' input data other than the training data set. The BPNN using backpropagation where this learning depends on mean square error to adjust the weights W in order to minimise the prediction error function θ. The objective is to keep the mean square error as small as possible. From equation (4.15) and (4.16), bias and variance directly affect the value of the mean square error. It is therefore important to keep these two components small as well. However, it is difficult to keep them simultaneously small.

From equation (4.15), the bias is also dependent on the weights W, therefore the size of the network plays an important role in enabling the generalisation ability of the BPNN. A small network with only one hidden node will most likely be biased, as the available function $f(X,W)$ has limited span to adjust its weights (German et. al, 1992). In neural network terms, it is underfitting. Figure 4.13 illustrates this. Lawrence et. al

(1996) and Fung et. al (1997) have shown that a large number of hidden nodes can make the learning fast with better training and generalisation errors. Yu (1992) has also shown that with large number of hidden nodes, it is more likely to have no local minima. From these analysis, it is realised that with large hidden nodes, the bias can be reduced, thus improved the model $f(X,W^*)$. Beside the weights relating to bias, the number of training vectors (X_n, Y_n) will also contribute to the amount of bias. The more available training vectors (X_n, Y_n) are available, the less bias is the model. Usually, for most applications where training data is difficult and expensive to obtain, this has little significant in reducing the bias.

It would seem that by reducing the bias, the mean square error can be reduced, but this will normally increase the variance. Therefore, there is a need to keep a balance between the variance and bias. The contribution of the variance is largely dependent on the noise involved and the distribution of the training set. For the case of the noisy data, when a BPNN tries to reduce the mean square error with small amount of bias using large number of hidden nodes, it has the danger that the variance will increase tremendously due to noisy training vectors. In effect, the final BPNN prediction model will not have good generalisation ability due to the high variance. This is the phenomenon of overfitting in neural network, and Figure 4.14 shows the graphical example of overfitting. In order to balance the contribution of the bias and variance in the final model, automatic smoothing technique can be applied (German et. al, 1992). The common smoothing techniques used are cross-validation (Stone, 1974; Plutowski et. al, 1994) and early stopping validation (Wang et. al, 1994; Sarle, 1995).

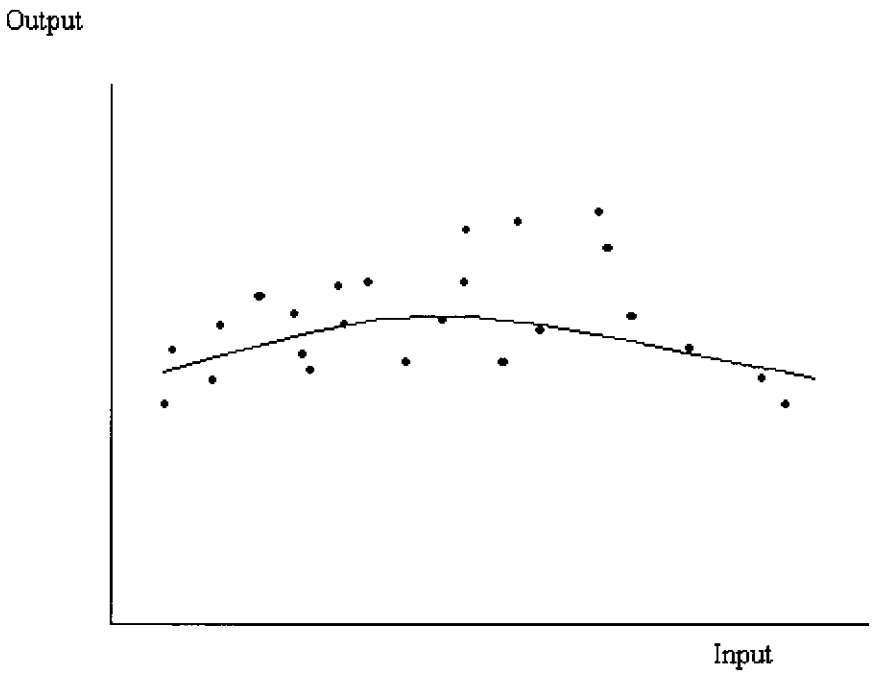


Figure 4.13: Underfitting

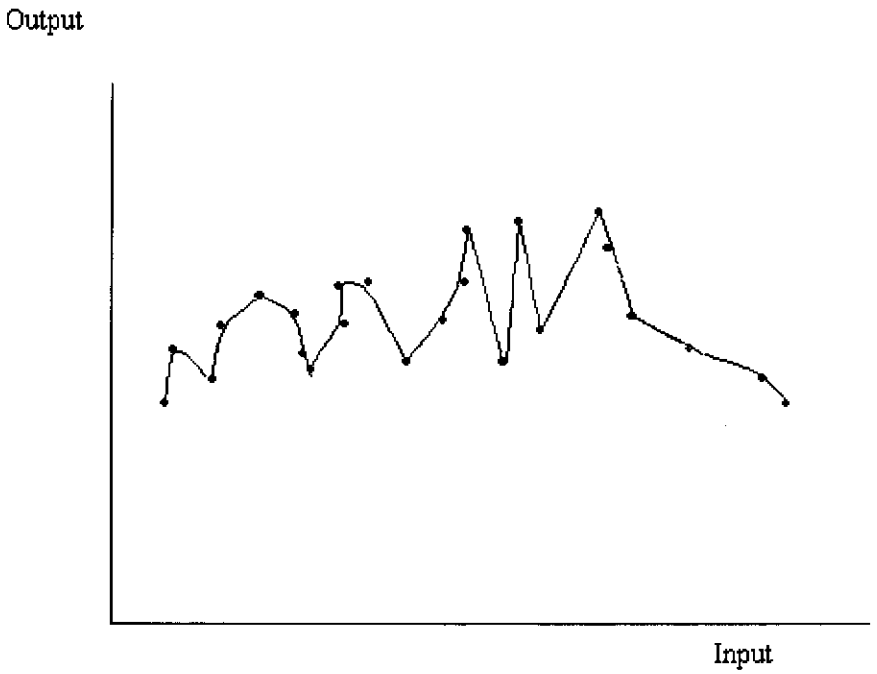


Figure 4.14: Overfitting

A smoothing technique is able to provide better control between bias and variance when the training set is noisy, but the distribution of the training data will generate another problem. Where the 'clean' data is not evenly distributed, the probability law ν will bias towards the majorities (large statistical frequency). As for the minorities (small statistical frequency), they will be smoothened up by the automatic smoothing technique and treated as noise. Figure 4.15 shows an example of a non-evenly distributed 'clean' data set, and how the expected function has been smoothed. This is valid as shown from equation 4.10, conditional probability law γ will affect to some extent on the final prediction model $f(X,W^*)$. In effect, this will increase the bias again. Under this circumstance, a technique needs to be introduced, such that the distribution of the 'clean' data can be evened up, such that the BPNN will be able to accommodate the minority characteristic function.

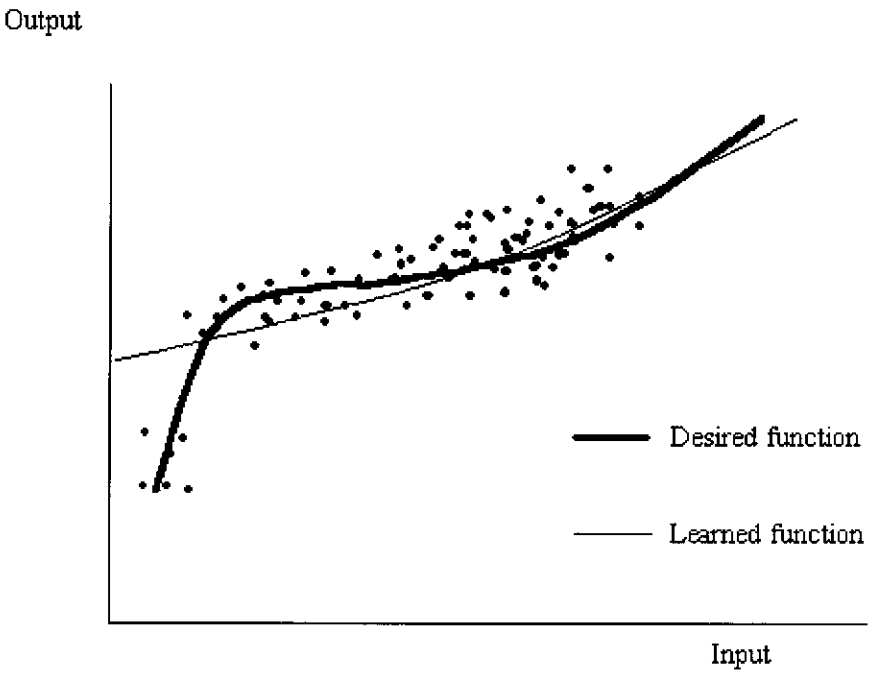


Figure 4.15: Non-Evenly distributed 'clean' data

In this section, the generalisation ability of the BPNN can be concluded that it is largely dependent on the following factors:

1. number of hidden nodes or size of the weights
2. amount of noise in the training set
3. distribution of the 'clean' training data in the training set.

A few points need to be noted from this analysis. First, a large number of hidden nodes or adjustable weights are favourable and can reduce bias. Secondly, the balancing of the bias and variance could be achieved by using some automatic smoothing techniques like cross-validation or early stopping validation. Thirdly, a

technique should be introduced to ensure the distribution of the 'clean' data is not smoothened by the automatic smoothing technique.

4.11 STATISTICAL ANALYSIS OF THE EARLY-STOPPING TECHNIQUE

Although cross-validation and early-splitting validation are considered two different types of automatic smoothing techniques, they do have their similarities. Both of them divide the whole sample of the available data set into training and validation sets. The difference is the way they perform smoothing in training of the BPNN. In cross-validation technique, the available data set is usually divided into k subsets of equal size. A k number of BPNN is set up, each time leaving out one of the subsets from the training. The validation error is then calculated only based on the omitted subset. This is sometime known as 'leave-one-out' cross-validation. However, the main disadvantage of this automatic smoothing technique is the training time needed to train k networks. As for early stopping validation, it works on the basis of split-sample methods. This only requires one network to be trained. This is a more practical and easier automatic smoothing technique.

When applying early stopping validation, the available data set is first split into training and validation sets. A very large number of hidden nodes are used to set up the BPNN. This is considered favourable as discussed in the previous section, as bias will be reduced. By using a small learning rate, the validation error (which is also mean square error) is calculated periodically. The training process is stopped when the validation starts to rise. In this case, the validation is just like a teacher guiding a

student. It therefore plays a very important part in obtaining the best generalisation ability of the BPNN.

Just like the error function used in training the BPNN, the validation process also contributes to the bias/variance dilemma. Beside this, the splitting of the training and validation set is the major contribution of the final generalisation ability of the BPNN. It has been demonstrated by Prechelt (1994) that the validation error will oscillate several times during the process of training. Figure 4.16 shows this situation. In the same paper, it was proposed that training should let the BPNN converge, and then observe the point with the smallest validation error.

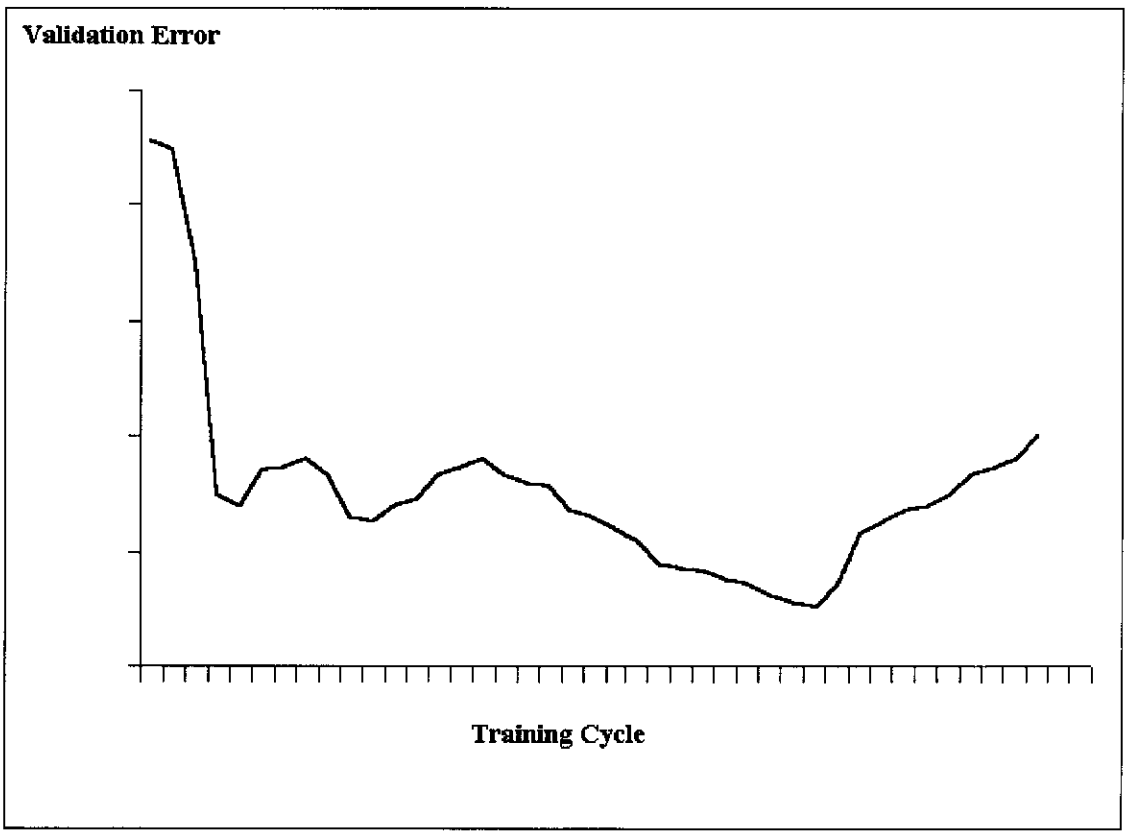


Figure 4.16: Oscillation of validation error.

4.12 FORMULATION OF AN APPROACH

4.12.1 PROBLEMS OF ENSURING GENERALISATION

The factors that directly affect the generalisation ability of a BPNN have been identified. The automatic smoothing technique has also been examined. It can be concluded that to reduce the bias of the mean square error term, a large number of hidden nodes is more practical so that underfitting can be prevented. An automatic smoothing technique is needed to ensure that the variance is kept small as well (to prevent overfitting). Early stopping validation is more preferable as it is fast and can be used in the situation where the number of hidden nodes is large. To find the lowest validation error, the training process is allowed to converge, and then the weights at the lowest validation error is used as the best weights W^*. However, there are still problems that have to be solved:

1. How to split the training and validation set?
2. How to modify the distribution of the 'clean' data?

This section will provide answers to these two problems. Before discussing the solution for the first problem, the characteristics of the Self-organising Map (SOM) algorithm need to be examined.

4.12.2 STATISTICAL ANALYSIS ON SELF-ORGANISING MAP (SOM)

SOM is designed with the intention to closely simulate the various organisations found in various brain structures and has a close relationship to brain maps (Kohonen, 1990; Kohonen, 1995). Its main feature is the ability to visualise high dimensional input spaces onto a smaller dimensional display, usually two-dimensional. In this discussion, only two-dimensional arrays will be of interest. Let the input data space $\mathcal{R}^n$ be mapped by the SOM onto a two-dimensional array with I nodes. Associated with each I node is a parametric reference vector $m_I=[\mu_{I1},\mu_{I2},\ldots,\mu_{I2}]^T \in \mathcal{R}^n$, where μ_{ij} is the connection weights between node I and input j. Therefore, the input data space $\mathcal{R}^n$ consisting of input vector $X=[x_1,x_2,..,x_n]^T$, ie $X \in \mathcal{R}^n$, can be visualised as being connected to all nodes in parallel via a scalar weights μ_{ij}. The aim of the learning is to map all the n input vectors X_n onto m_I by adjusting weights μ_{ij} such that the SOM gives the best match response locations.

SOM can also be said to be a nonlinear projection of the probability density function $p(X)$ of the high dimensional input vector space onto the two-dimensional display map. Normally, to find the best matching node I, the input vector X is compared to all reference vector m_I by searching the smallest Euclidean distances $\| X - m_I \|$, signified by c. Therefore,

$$c = \arg\min_i \{\| X - m_i \|\} \tag{4.17}$$

or

$$\| X - m_c \| = \min_i \{\| X - m_i \|\} \tag{4.18}$$

During the learning process, beside the node that best matches the input vector X is allowed to learn, those nodes that are close to the node up to a certain distance will also be allowed to learn. The learning process is expressed as:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[X(t) - m_i(t)] \tag{4.19}$$

where t is discrete time coordinate

and $h_{ci}(t)$ is the neighbourhood function

After the learning process has converged, the map will display the probability density function $p(X)$ that best describes all the input vectors space. At the end of the learning process, an average quantisation error of the map will be generated to indicate how well the map matches the entire input vectors X_n. The average quantisation error is defined as:

$$E = \int \| X - m_c \|^2 \, p(X) dX \tag{4.20}$$

Beside the average quantisation error, an individual quantisation error is also used to measure how well the input vector matches the closest node I, and is similar to equation (4.18).

4.12.3 STATISTICAL ANALYSIS OF SOM DATA SPLITTING

Section 4.10 showed how important and crucial a task is splitting the available data into training and validation sets. The training set will give information on what the BPNN should learn, and the validation set acts as a teacher to guide the BPNN such that it will learn the correct function. As the BPNN is based on a training set to obtain the underlying knowledge, therefore it should contain more data than the validation set. Although it is known that the training set should be larger than the validation set, the problem of how to effectively split them still exists.

The rule for splitting the available data into a training and validation set is that the training set should be statistically similar to the whole sample space. The validation set should also be statistically similar to the training set as it has to act as a teacher. With this rule, by looking back to SOM algorithm in the last section, SOM can be used as a nonlinear probability density function projection on the two-dimensional map. Therefore in each node I, the probability density function of the input vectors being mapped onto it should have a similar probability density function. This also implies that the input vectors that are mapped onto the same node should have similar relative occurrences as denoted by $P(X)$. This $P(X)$ is similar to the environmental probability law $P(\mu)$ in equation (4.10). From the analysis in section 4.10, the role of training the BPNN can be regarded as a search for the conditional probability law $P(\gamma)$. The formulation of $P(X)$ here has to be extended. Instead of mapping just the input vector X, both input vector X and target vector Y are used in the learning of the SOM. A joint probability between X and Y is assumed and is denoted as $P(X,Y)$. It can further be expressed as:

$$P(X,Y) = P(X \mid Y)P(Y) = P(Y \mid X)P(X) \qquad (4.21)$$

As equation (4.21) is actually similar to equation (4.10), it can then say that the joint probability function density of a SOM is directly related to the joint probability law mentioned in section 4.10. With this, it can also be realised that the joint vectors of X and Y falling in the same node should have very similar statistical characteristics.

In order to satisfy the rule of splitting data, a methodology can be formulated. The n available data set that consists of X input vector and Y output vector are first used to train the SOM. After the map has been trained and individual quantisation errors have been generated, a selection can be made. A data set is selected as validation data if it has a small quantisation error as compared to the other data sets in the same node. This will ensure that the validation set is a sub-set of the training set. However, for cases where there is only one data set in that node, it will be left in the training set. This is to ensure that the training set can covers the whole sample space of the available data, and to ensure that the training set is always larger than the validation set. After all the available data has been split into training and validation sets, the BPNN can start to learn and the process is stopped by using early stopping validation technique. Until now, the user can increase the confidence of the generalisation ability of the BPNN by solving one of the problems mentioned in section 4.12.

4.12.4 STATISTICAL ANALYSIS OF INTERACTIVE REINFORCEMENT LEARNING

To address the second problem mentioned in section 4.12, it is assumed that the user has some knowledge of the problem in order to identify the minority 'clean' data. An interactive reinforcement learning works on the principle that when a child cannot understand a concept, the teacher will repeat the same concept again and again until the child can pick it up.

If the 'clean' data appear to have small statistical frequency, it will not be represented by the joint probability law ν, and eventually be smoothened up by the early stopping validation technique. A straightforward approach in order for the minority 'clean' data to be accommodated in the final generalisation function of the BPNN, is to go back and obtain more data sets. However, it may be impossible for most cases, as data are difficult and expensive to obtain. A methodology has been proposed to increase the statistical frequency of that 'clean' data easily and confidently. By repeating the same data set a number of times, the relative statistical frequency of that 'clean' data set will be increased. This will has an effect on the final generalisation ability of the BPNN as it changes the mean of the whole data set. It will subsequently affect the mean square error from the learning process, and, it also affects the bias and variance of the model. As the early stopping validation technique makes use of training set and validation set to perform the automatic smoothing, it is important that the 'clean' data is repeated in both the training and validation set. This process may need to be repeated a few times until the minority 'clean' data set can be accommodated in the final generalisation function of the

BPNN. With this interactive reinforcement learning method, all the problems mentioned have been solved.

4.13 CONCLUSIONS

As the study of statistics is a powerful tool in data analysis, it is used in this chapter to establish a procedure for determining the factors that contribute to the generalisation ability of a BPNN. Statistics provides a greater understanding and a more meaningful explanation of the factors involved, thus they indicate directions for searching new solutions.

The factors that have been identified t are the size of the weights or the number of hidden nodes, the amount of noise in the training set, and the distribution of the 'clean' data. From this statistical analysis, the techniques have been developed to ensure that the best generalisation function of the BPNN can be obtained. To avoid underfitting, a large number of hidden nodes is used to set up the BPNN. This will also enable that the vector of the weights can learn any complexity inherent in the problem. In order to avoid the BPNN from overfitting, early stopping validation is used to perform automatic smoothing. The SOM data splitting approach was shown to be able to split the training and validation set satisfactorily. In cases where the distribution of the 'clean' data is not even, interactive reinforcement learning may be used to require the BPNN to accommodate them while rejecting noise at the same time. After the statistical analysis of the generalisation problem and the techniques proposed, there can be some confidence that the generalisation ability of a BPNN can be increased.

CHAPTER 5:

A COMPACT GENERALISED NEURAL FUZZY SYSTEM

5.1 THE APPLICATION OF FUZZY LOGIC

As has been shown previously, an Artificial Neural Network (ANN) has the ability to perform non-linear input and output mapping from a training data set. It is also capable of generalisation by rejecting noise and generating results for input data that are new to the network. However, once the ANN is trained, it acts like a "black-box". Further, a user will have some difficulty in comprehending the significance of the vast number of weights involved. Moreover, the effects of the output are unpredictable if some of the weights are modified.

On the other hand, a Fuzzy Logic (FL) system seems more reasonable for expressing knowledge or underlying functions in linguistic terms. Examining the fuzzy rules to understand the behaviour of the analysing system is a relatively easy task for most observers. Besides presenting human understandable rules, a FL system also has the ability to handle fuzzy information and so is capable of handling non-linear functions. As a conventional FL system does not have the ability to learn and adapt from the available data, the setting up of the fuzzy rules can be a very tedious task. It can be even more tedious if a large number of input parameters are involved. A solution to the problem is proposed by extracting the rules from the training data. The proposal is achieved by modifying the self-generating fuzzy rule extraction algorithm of Abe and Lan (1995). However, the extracted rules do not have the ability to reject noise when transforming every set of training data into rules. In this

case, they may not have the best generalisation capability. Besides, the rules generated may not be able to interpret the new input data, as the rules may not cover the whole universe of discourse. When a FL system is used for the purpose of function approximation, it suffers from the curse of dimensionality. The number of fuzzy rules increases exponentially with the number of input variables as well as the number of fuzzy membership functions. Due to these disadvantages, there is a need to integrate the two techniques (ANN and FL) together such that they will complement each other.

A compact generalised fuzzy interpretation system using a Backpropagation Neural Network (BPNN) for ensuring generalisation capability has been developed. The first step in designing the system is to perform an input contribution measure in order to identify the significant input variables. This will recognise those variables that are needed to perform reasonable prediction. At the same time, this reduced set of input variables will also cause the number of rules involved in the final system to be reduced. The next step is to train a BPNN with the available training data using the identified input variables. The approaches outlined previously are used to allow the network to learn the best generalisation point. After the network is trained, it is used to generate training data according to the number of memberships defined in the fuzzy system. The self-generating fuzzy rules algorithm is then used to extract rules from these data. This ensures that the rules generated will incorporate the generalisation capability of the trained BPNN. As the data generated by the BPNN will cover the whole universe of discourse of the fuzzy system, the rules in the final system can be used to interpret any data that are not covered by the original training data.

As the fuzzy rules generated in this system are directly related to the number of input parameters and membership functions, the final rule base could be very large. A user would need to spend considerable time in examining the rule base if changes were planned, and it is unlikely to be practical to examine all the fuzzy rules. As the purpose of generating fuzzy rules in obtaining the generalised underlying function from the training data is to allow for user interaction, an approach to reduce the fuzzy rules is necessary. Therefore, a reduced fuzzy rule-base approach to the development of a generalised neural-fuzzy interpretation model is outlined.

5.2 AN INPUT CONTRIBUTION MEASURE

5.2.1 THE IMPORTANCE OF IDENTIFYING THE SIGNIFICANT INPUT

In most practical data analysis problems, the number of available input variables for analysis can be very large. In most cases, an analyst must rely on their experience to determine the relevant input variables before performing the data analysis. With the increasing sophistication of problems tackled, the number of available input variables may increase tremendously. This is especially true in the case of well log data analysis. Hence, the task of finding the most appropriate input variables can be very tedious.

Although dealing with a large number of inputs and outputs does not pose any particular difficulty for a BPNN, several factors must be taken into account when constructing the network. If a large numbers of input variables are used in the prediction model, the training time will be very long. On the other hand, if too few

input variables or inappropriate variables are used, the interpretation model may not be accurate. Also, when the generalisation function is transformed into fuzzy rules, the number of fuzzy rules will be very large as unnecessary input parameters are incorporated into it. Hence, it is necessary to select the most appropriate input variables for the prediction model. In addition, from a user's viewpoint, it would be useful to know the input-output relationship being established by the model and what input variables are crucial in predicting the desired output.

5.2.2 MEASURING INPUT CONTRIBUTIONS

After a BPNN is trained, the information of the model is represented by the values of the weight connections. In most cases, the measurement of input contributions is determined from the magnitude of these weights. Garson (1991) has proposed an input contribution measurement based on a formula that calculates the size of the input to the hidden weights with respect to the sum of all the weights in the input layer. It is then weighted by the magnitude of the connection to the respective output units. Wong et al (1995c) have used another measure based on the input contributions to the hidden units by using the absolute values of the weights. In both approaches, it is necessary to examine the weights of the network one by one. This requires relatively complex analysis and in some cases, the results may not be accurate.

A straightforward approach to measure input contributions without the need of weights analysis is outlined. A BPNN is treated as a 'black box' with inputs, [i_1, i_2, ... i_n] and outputs, [o_1, o_2, ... o_m]. The measurement is based on the change of output

with respect to the maximum variations in each input. The training set is replaced by cycling each input with its maximum and minimum values. This is illustrated by the following process which assumes a network with n inputs, $[i_1,....i_n]$ and one single output, o.

Step 1: Starting from training pattern 1, i_1 is replaced by its minimum while the other inputs remain unchanged. Calculate the output as $o_1 | \mathit{min}\ i_1$.

Step 2: Similar to Step 1 but the maximum of i_1 is used. The output is denoted as $o_1 | \mathit{max}\ i_1$.

Step 3: The derivative of the first pattern is calculated from the difference of these two values, as

$$\Delta o_1 \mid i_1 = (o_1 \mid \max i_1 - o_1 \mid \min i_1)$$

Step 4: Steps 1 to 3 are repeated for the next pattern until the last training pattern, p.

Step 5: The normalised derivative for input 1 at the output is then calculated from

$$T\Delta o \mid i_1 = \frac{\sum_{j=1}^{p} (\Delta o_j \mid i_1)^2}{p}$$

where p is the number of training data

Step 6: The above 5 steps are repeated for the other inputs until all the normalised input derivatives are obtained, i.e.

$$[T\Delta o \mid i_1, T\Delta o \mid i_2, ..., T\Delta o \mid i_n]$$

Step 7: The input contributions for the output are then calculated in terms of a percentage from the following expression:

$$C_k = \frac{T\Delta o \mid i_k}{\sum_{j=1}^{n} (T\Delta o \mid i_j)} * 100\% \qquad (5.1)$$

where C_k is the % contribution at input k,

n is the number of input nodes, and

o is the output.

If the network has more than one output, the same procedure can be applied by extending it to other outputs.

The case studies following are used to illustrate the benefits gained by employing this input contribution measure.

5.2.3 A CASE STUDY TO ILLUSTRATE THE INPUT MEASURE

5.2.3.1 THE DATA SET

In this section, three sets of test results are presented to illustrate the value of the approach where all are based on a well log problem. In the first, the main objective is to illustrate the performance of this methodology. In the second, it is to study how this input contributions measure can assist in selecting the significant and most appropriate input logs. In the third, known insignificant inputs are artificially generated and included, with the purpose of determining the ability of the input contribution measure to reject these unrelated inputs.

A set of 90 training data from a typical well was used to train a neural network based model using a SOM data-splitting validation approach proposed in chapter 4. Another set of 89 data was then used as the test set for evaluating the prediction performance of the trained network. In this well, there are a total of 12 input logs. They are: (1) photoelectric (PEF), (2) bulk density (RHOB), (3) neutron (NPHI), (4) caliper (CALI), (5) uninvaded resistivity (RT), (6) invaded zone resistivity (RXO), (7) gamma ray (GR), (8) potassium (POTA), (9) thorium (THO), (10) uranium (URA), (11) sonic travel time (DT), and (12) spontaneous potential (SP). The output petrophysical property that is to be predicted is porosity (PHIE).

5.2.3.2 THE INPUT MEASURE TEST

In the first test, all the twelve available input logs were used. The network configuration consisted of 12 input nodes, 24 hidden nodes and 1 output node. After

the network was trained using the SOM data-splitting validation approach, the test set was then used to assess the performance of the network. Based on the input contribution measure, the two most significant input logs were purposely eliminated. The BPNN was then re-trained with the remaining 10 input logs. The purpose of this exercise was to test the effects on performance of the network when the most significant inputs were removed.

The objective of the second test was to illustrate the selection process for the most appropriate inputs. It also gave a suggestion on the number of input logs that will produce the best prediction result. The BPNN trained in the previous test was first trained with all 12 input logs. Subsequent tests reduced a number of logs in turn based on the contribution measure. The tests results obtained are shown in Table 5.1. Ranking of the input contributions in the second column of the Table 5.1 is based on the percentage contributions as calculated from equation 5.1, with 1 giving the highest contribution and 12 as the lowest.

Table 5.1: Case Study Two Tests

Example	Ranked input logs being eliminated	No. of input logs used
1	11 & 12	10
2	9,10,11&12	8
3	8,9,10,11&12	7
4	7,8,9,10,11& 12	6
5	6,7,8,9,10,11&12	5
6	5,6,7,8,9,10,11&12	4
7	3,4,5,6,7,8,9, 10,11&12	2

In the third test, the network with the most appropriate input logs that gave the best prediction results in the previous study was used. Two examples stood out. In the first, an extra input, generated randomly, was added to the training set. In the second, the number of randomly generated inputs was increased to four. The purpose of this test was to observe the values of the input contributions measure. These unrelated inputs were expected to contribute insignificantly if they were not correlated to the output.

Mean square error (MSE) was used to evaluate the performance of the networks where this was calculated according to:.

$$MSE = \frac{(\sum (T_p - O_p)^2)}{2P}$$

where T_p = target pattern

O_p = output pattern

P = number of patterns

5.2.3.3 RESULTS FROM TEST ONE

The input contribution percentages of the network with all 12 input logs are shown in Figure 5.1. Another network was trained with only 10 input logs by eliminating the two most significant contributors. The logs left out in this example were bulk density (RHOB) and uninvaded resistivity (RT). A comparison of the results from these two networks is shown in Table 5.2.

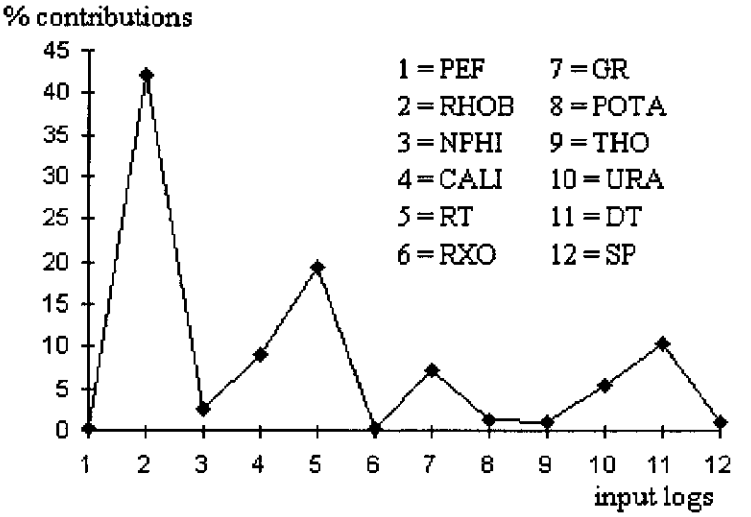


Figure 5.1: Percentage Contributions of Input Logs

Table 5.2: Errors from Test One

Example	No. of Input	ERROR
With all input logs	12	0.0011
Leave out RHOB & RT	10	0.0280

These results show that by leaving out the two most significant input logs, the prediction error based on the testing data set has increased greatly. Further, that the input contributions measure gives an accurate indication of the importance of the input logs used in the porosity prediction model. It does not require any prior knowledge and examination of the weights within the network.

5.2.3.4 RESULTS FROM TEST TWO

Seven networks with a different number of input logs were trained. The input logs used for each example are tabulated in Table 5.3 and the MSE for all the examples are shown in Table 5.4.

Table 5.3: Input logs used in each example.

Example	Input logs used	No. of logs
1	RHOB,NPHI,CALI,RT,GR, POTA,THO,URA,DT,SP	10
2	RHOB,NPHI,CALI,RT,GR, POTA,URA,DT	8
3	RHOB,NPHI,CALI,RT,GR, URA,DT	7
4	RHOB,CALI,RT,GR,URA, DT	6
5	RHOB,CALI,RT,GR,DT	5
6	RHOB,CALI,RT,DT	4
7	RHOB,RT	2

Table 5.4: MSE measure for each example.

Example	ERROR
1	0.0012
2	0.0010
3	0.0008
4	**0.0007**
5	0.0017
6	0.0023
7	0.0035

From the table it can be observed that the error starts to decrease with the reduction of input logs with least significance. However, if too many input logs are eliminated, the prediction accuracy will start to reduce. From Figure 5.1, it can be observed that a number of input logs contribute insignificantly. They are PEF, NPHI, RXO, POTA, THO and SP. By reducing these logs, then as shown in Table 5.4, the overall performance from Example 4 is shown to be the best. This illustrates that the selection of the most appropriate input logs can be easily carried out by inspecting a plot of the input contribution measure similar to Figure 5.1.

5.2.3.5 RESULTS FROM TEST THREE

In this test, a few randomly generated inputs were used to test the ability of the proposed method in rejecting unrelated inputs. Two trials were carried out. The number of logs used in each is tabulated in Table 5.5. Results from Example 4 of

Case Two in Table 5.4 show the best prediction result. The network is therefore used as the basis of this study. The randomly generated inputs are identified as N in Example 1 and N1, N2, N3 and N4 in Example 2 as shown in Table 5.5.

Table 5.5: Input logs used in each Example.

Example	Input logs used	No. of logs
1	RHOB,CALI,RT,GR,URA, DT,N	7
2	RHOB,N1,CALI,N2,RT,GR, N3,URA, DT,N4	10

Figures 5.2 and 5.3 are plots of the percentage contributions of all the input used in these two examples. The randomly generated inputs in both tests contribute insignificantly. By using the proposed selection criteria, these input logs can be ignored without affecting the overall performance of the network.

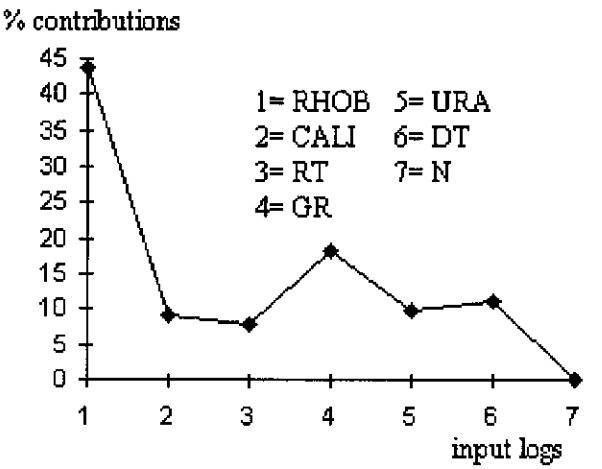


Figure 5.2: Percentage Input Contributions of Test 1.

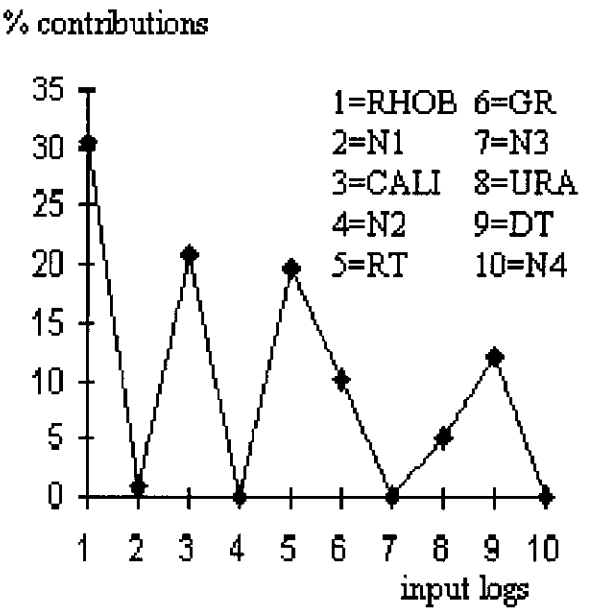


Figure 5.3: Percentage Input Contributions of Test 2.

5.2.4 CASE STUDY 2

5.2.4.1 HYDROCYCLONE ANALYSIS

Data collected from a Krebs hydrocyclone model D6B-12o-839 were used. A BPNN was first trained with 95 samples of training data. The network was then tested with 44 samples of testing data not used in the training process. These data were used to determine the prediction ability of the network. Fourteen input parameters were used initially. The output of the network is the d50c value which determines the separation efficiency. Three networks were trained and the input parameters used in each case are shown in Table 5.6. In Test 1, a network was trained with all the 14 input parameters. They were the inlet flowrate (Qi), overflow flowrate (Qo), underflow flowrate (Qu), ratio of flowrates (Qo/Qu), split ratio (So/Su), solid percentage (Pi), vortex finder height (H), spigot opening diameter (Du), inlet density (Ri), overflow

density (Ro), underflow density (Ru), water split ratio (WS), differential pressure between the inlet and the overflow streams (dP) and temperature of slurry (T).

After completion of the training process, the input contributions measure was then used to examine the relative significance of each parameter. A plot of the percentage contributions for each parameter is shown in Figure 5.4. Based on the results from this measurement, six significant input parameters were selected. They were used to train another network in Test 2. The significant inputs selected were Qi, Ri, Ro, Pi, dP and Du. In order to compare results from previous work, Test 3 was a network which used parameters that were found in an empirical formula reported by Gupta et. al (1990). The parameters used in Test 3 are Qi, Ri, H, Du and T. In Test 3, the most significant parameter Ro is left out.

Table 5.6: Input parameters used in each test.

Test	**No. of input**	**Input parameters used**
1	14	Qi, Qo, Qu, Qo/Qu, So/Su, Ri, Ro, Ru, Pi, dP, WS, H, Du, T
2	6	Qi, Ri, Ro, Pi, dP, Du
3	5	Qi, Ri, H, Du, T

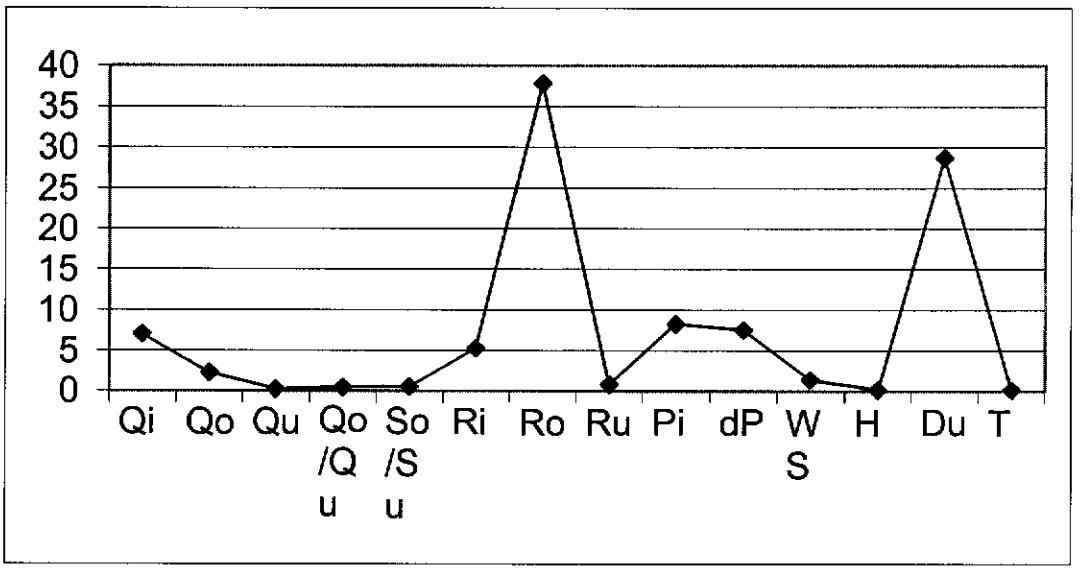


Figure 5.4: Input contributions measure of all 14 input parameters

Results from these tests in terms of training time, training and testing accuracy are summarized in Table 5.7. The last row in Table 5.7 provides accuracy due to the calculated results from the empirical formula. The training time assumes processing on a Pentium 90 Personal Computer where in this case the program was developed in the C++ programming language. From these results, it can be observed that Test 2 has given the best performance. It requires only a third of the training time used in Test 1. Although Test 1 has utilised all 14 parameters, the accuracy is almost identical to that of Test 2. Test 3 required less time than Test 2, but the performance is worse. This is due to the fact the most important parameter has been omitted. The final result due to the empirical formula did not require any training time (but extensive work has been done previously to establish such a formula), and the performance is shown to be lower than the neural network results.

Table 5.7: Results from neural networks and empirical formula

Test	Training time	Correlation of network output against TRAINING data	Correlation of network output against TESTING data
1	11 min	0.989894	0.992331
2	4 min	0.992234	0.993748
3	3 min 25 sec	0.984593	0.985527
Gupta & Eren's formula	NA	0.748121	0.849941

5.3 A SELF-GENERATING FUZZY INTERPRETATION SYSTEM

5.3.1 SETTING UP THE FUZZY RULES

The objective of this self-generating fuzzy system is to aid the user in setting up a fuzzy rules interpretation model by mapping the available data to their corresponding memberships. After this has been done, the user can examine the interpretation model from the fuzzy rules and then modify or add-on to the rule base easily. The fuzzy interpretation model is established in the following manner:

(1) Normalise the data between 0 and 1 by using a linear or logarithmic transformation depending on the nature of the data. This is to ensure that the resolutions of all data are similar.

(2) Define the shape of the membership function, number of fuzzy regions and fuzzy terms for all data.

In this approach, only triangular membership functions are used. The number of fuzzy regions used is the same for all inputs and output. Fuzzy terms used are in the form of L for low, M for medium and H for high. An example of a five fuzzy region term is:

VL, L, M, H, VH.

(3) The space associated with each fuzzy term over the universal discourse for each variable is then calculated and divided evenly. For example a value with range between 0 and 1 with 5 membership-terms is shown in Figure 5.5.

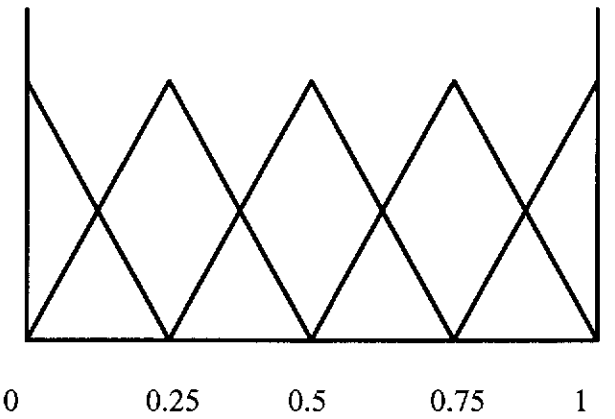


Figure 5.5: Distribution of 5 membership-terms

(4) For each available training data, a fuzzy rule is established by directly mapping the physical value of the variable to the corresponding fuzzy membership function.

Most of the time for a given value, it will normally fall into more than one fuzzy region. In this case, a degree is given to that value in the fuzzy region. The value is then assigned to the fuzzy region with maximum degree.

(5) Go through Steps (1-4) with all the available data and generate one rule for each input-output data pair.

(6) Reduce the fuzzy rule base.

In this step, all rules are examined for similarity. Similar rules are then eliminated and taken out of the rule base.

(7) The set of reduced fuzzy rules together with the centroid defuzzification algorithm now forms the fuzzy interpretation model.

After the fuzzy-rule interpretation model has been set up, it can then be used to predict any from any unknown input data. With the fuzzy interpretation model being set up, based on the analyst's experience, the rules may be manipulated explicitly to incorporate human knowledge and experience.

5.3.2 APPLICATION OF A SELF-GENERATING FUZZY INTERPRETATION SYSTEM

A case study is used to illustrate the application of this proposed approach. Well log data from two typical wells was used to predict the petrophysical property, porosity (PHI). Core data from one well was used to establish a prediction model based on the proposed self-generating fuzzy rules inference system. The model was then used to predict the porosity of the second well. The input logs used in this case study were gamma ray (GR), deep induction resistivity (ILD) and sonic travel time (DT) and all the variables are normalised between the values of 0 and 1. The first well had a total of 71 core data and was used as the training well. The second well had 51 core data and was used as the testing well to test the prediction accuracy of the trained fuzzy interpretation model.

A few tests were carried out to see the effect of the number of memberships. The numbers of rules obtained by varying the number of memberships are shown in Table 5.8. The rules extraction time for the three cases varied from 30 sec to 1 min based on computation using a Pentium 90 Personal Computer. Results showed that the number of fuzzy rules increased with the number of membership terms. The prediction accuracy for both training and testing wells in each test case is tabulated in Table 5.9. It can be observed from Table 5.9 that the results obtained from the self-generating fuzzy rules inference system have high correlation to the original core data. Figures 5.6 and 5.7 are the output plots from the 9 memberships fuzzy inference system for the training and testing wells respectively. Figure 5.8 shows a section of the fuzzy rules extracted from the core data after rule elimination using 5

membership. Figure 5.9 shows the fuzzy membership function for the 5 memberships fuzzy inference system.

Table 5.8: No. of rules generated for each case.

No. of membership	No. of rules extracted
5	29
7	46
9	63

Table 5.9: Prediction accuracy for each case.

No. of membership	Training Correlation	Testing Correlation
5	0.805	0.792
7	0.889	0.853
9	0.917	0.865

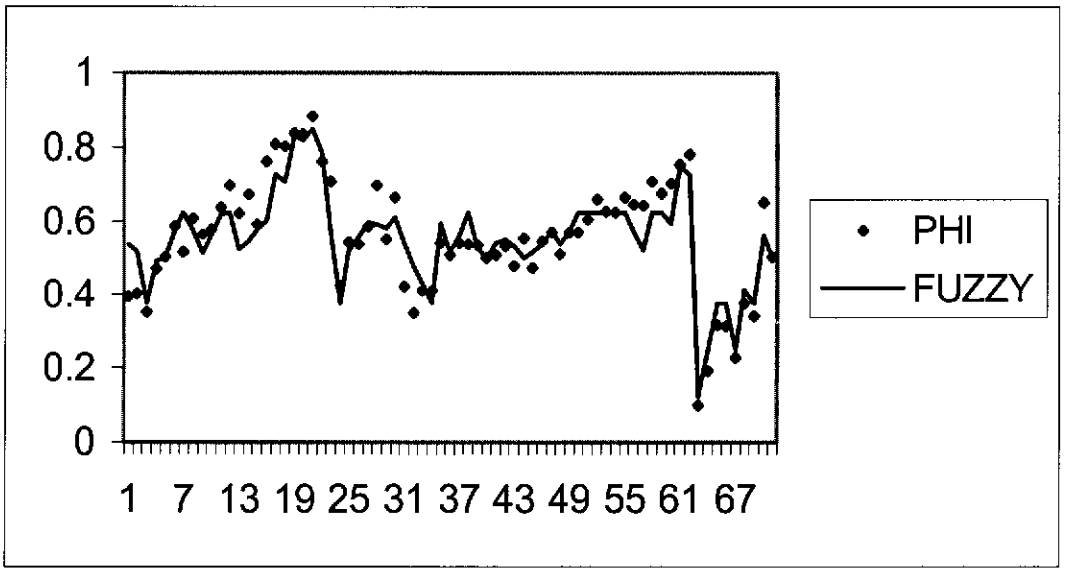


Figure 5.6: Output plot of TRAINING well using 9 membership functions.

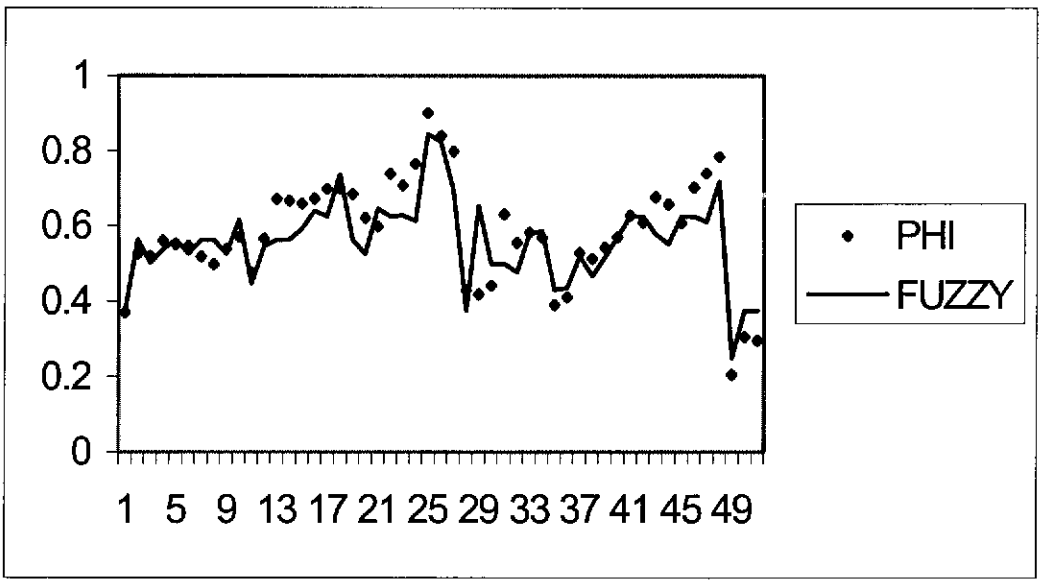


Figure 5.7: Output plot of TESTING well using 9 membership functions.

If GR = h and ILD = l and DT = h then PHI = m

If GR = h and ILD = vl and DT = m then PHI = l

If GR = m and ILD = vl and DT = h then PHI = m

If GR = h and ILD = vl and DT = h then PHI = m

If GR = vh and ILD = vl and DT = h then PHI = m

If GR = vh and ILD = l and DT = h then PHI = m

If GR = h and ILD = l and DT = h then PHI = h

If GR = h and ILD = l and DT = vh then PHI = h

If GR = vh and ILD = l and DT = vh then PHI = h

If GR = m and ILD = m and DT = m then PHI = m

Figure 5.8: Section of rules for 5 membership fuzzy system.

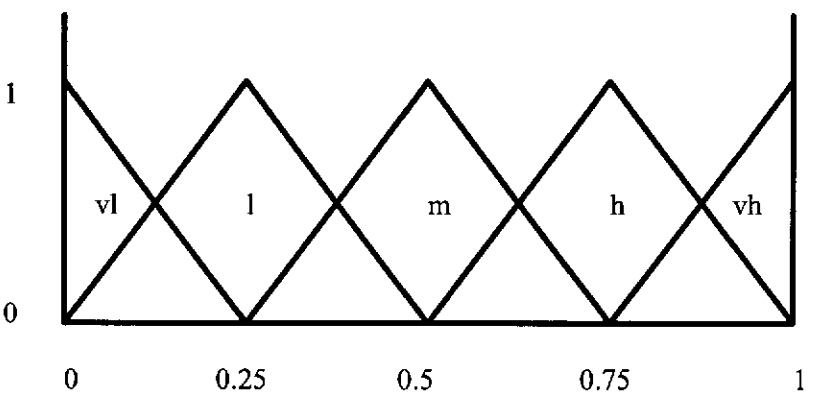


Figure 5.9: Fuzzy terms and regions for 5 memberships fuzzy inference system.

From the results, the fuzzy interpretation model can generate promising predictions. The time taken in setting up the fuzzy interpretation model is also very short. With the understanding of the membership functions such as those in Figure 5.9, the rules can be examined and modified. As the raw data has been first normalised between 0

and 1, the membership functions of all the input and output variables will be similar as those shown in Figure 5.9. This again enables the analyst to easily understand all the fuzzy terms regardless of the number of input and output variables.

5.4 KNOWLEDGE REPRESENTATION OF A BPNN BY FUZZY RULES

5.4.1 COMPACT GENERALISED NEURAL-FUZZY SYSTEM

Although the self-generating fuzzy rules system could extract human understandable rules, it has no generalisation capability and the rules generated cannot cover the whole universe of discourse. A technique is therefore presented to confidently set up a fuzzy system understandable by human users. At the same time, though, the rules involved should best describe the underlying function of the training data by rejecting noise. In reducing the number of rules in the final system, the input contribution measure described earlier has been incorporated to identify the significant input variables.

A BPNN and the self-generating fuzzy rules system have both strengths and weaknesses. A BPNN uses SOM data-splitting and an early stopping validation method to provide generalisation ability from the available training data. However, once the network is trained, it is difficult to understand the operation of the system and a user cannot add or modify the behaviour of the model. In the self-generating fuzzy rules system, the fuzzy rules are extracted from the training data. There is no validation to ensure that the rules extracted are describing the generalised underlying function of the training data. Further, as the rules are extracted for all training points, noise will also be included in the fuzzy rule-base.

The compact generalised neural-fuzzy system outlined combines the two approaches in order to preserve the advantages of each technique. It is based on three broad concepts. First, the input contribution measure based on a BPNN is used to identify the significant input variables. Second, a generalised BPNN is trained according to the techniques outlined in the previous chapter. The third part involves the setting up of a self-generating fuzzy rules system. The following procedure outlines how the overall system is established.

Step 1. Perform an Inputs Contribution Measure based on a BPNN.

Step 2. Train another BPNN using SOM data-splitting and early stopping validation with only the significant input variables.

Step 3. After the network has been trained, determine the number of memberships for the fuzzy system.

Step 4. Generate input variables for all possible memberships.

Step 5. Apply the generated input data to the BPNN and obtain outputs for the corresponding inputs.

Step 6. Use the self-generating fuzzy rules algorithm to establish fuzzy rules based on the input and output data generated from the neural network.

Step 7. The extracted rules form the fuzzy rule-base of the generalised fuzzy interpretation system. The final system uses the centroid defuzzification technique.

To apply the system to any unknown data set, the new incoming data are first normalised. In order to ensure similar responses to those in the training set, identical scaling factors for the training data are used. The input data are then fuzzified according to the predefined membership functions. Each set of input data in fuzzy terms is applied to the fuzzy rule-base and a fuzzy output is obtained through the inference process. The output is then defuzzified to become the crisp output from the generalised fuzzy interpretation model.

5.4.2 USAGE OF THE COMPACT GENERALISED NEURAL-FUZZY SYSTEM

The problem of well log data analysis is used as a case study to examine the performance of this compact generalised neural-fuzzy system. A typical problem comprising two boreholes is used. There is a total of 289 core data in the first borehole that is used as the training well and a total of 140 core data in the second. The second borehole was used as "blind test" to benchmark the performance of this proposed interpretation model and so not used in the training process in any way. For these two boreholes, data from 13 well logging tools were available. They were bulk density correction density (DRHO), photoelectric capture cross section (PEF), bulk density (RHOB), neutron (NPHI), caliper (CALI), uninvaded resistivity (RT), invaded zone resistivity (RXO), gamma ray (GR), potassium (POTA), thorium (THO), uranium (URAN), sonic travel time (DT), and spontaneous potential (SP). The petrophysical property that was of interest in this case was the volume of clay (VCL). The presence of clay has an important effect on the permeability that governs the ease of extraction of fluid. In addition, it also affects the log readings. It is therefore one of the important properties in well log interpretation. In this study, all the data was normalised between 0 and 1.

A BPNN using all 13 input well logs and 1 output was established to perform the input contribution measure. The network was trained and stopped at 5000 iterations despite the fact that the training error may not have reached a sufficiently low value. After the training was stopped, the percentage input contribution measures from all the input well logs was obtained and shown in Figure 5.10. In this case, the input

well logs found to be important were RHOB, NPHI, RT and GR. With the purpose of reducing the number of fuzzy rules in the final system, only the four inputs with highest contribution percentage were selected as the significant inputs. However, if only the GR was selected in designing the system, the prediction would not be good even though it has the highest percentage and clearly stands out from the others. This is due the fact that the volume of clay is not solely dependent on GR, but also has some degree of dependence on RHOB, NPHI and RT. Although DRHO and PEF may have effects on the final system, their presence may not improve the accuracy, as their percentages are considerably lower. However, if they are also selected, the fuzzy rule-base will have very large number of fuzzy rules.

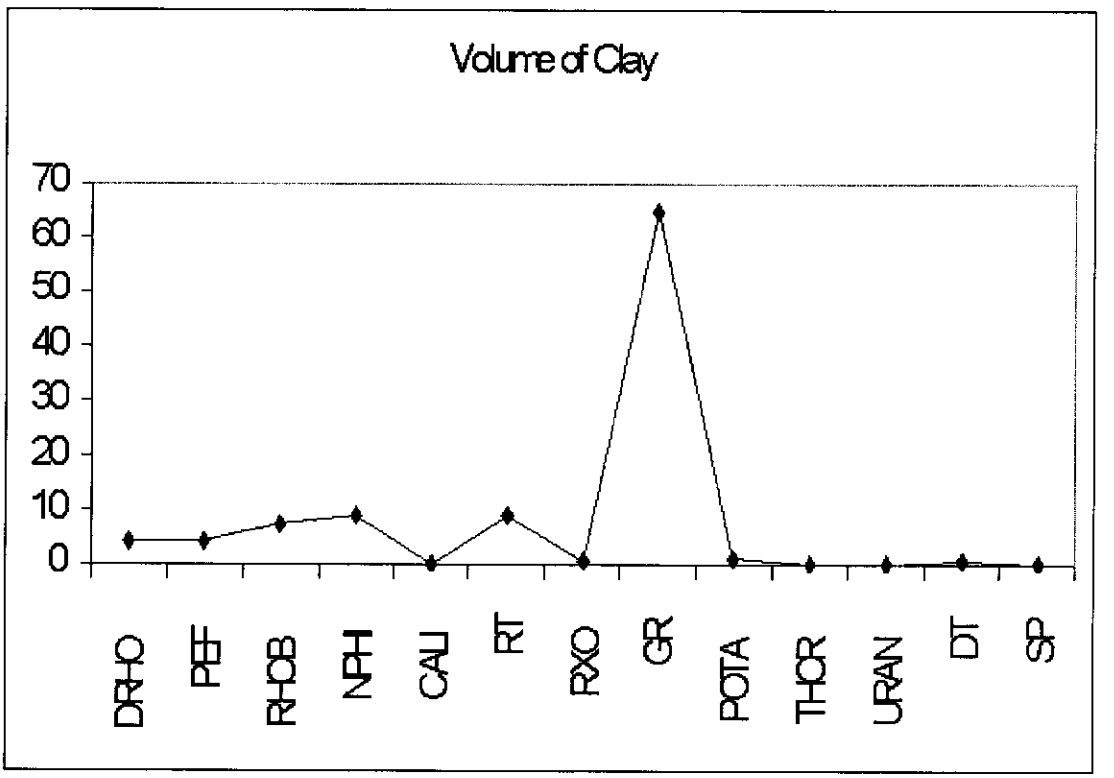


Figure 5.10: Input Contribution Measure of the Test Case.

Based on the most significant input logs, two interpretation systems were set up and the accuracy of these models was compared. The first interpretation system only made use of a BPNN. The second was a proposed compact generalised neural-fuzzy system. To compare the effects of the number of fuzzy memberships on the performance of the second system, 5 and 7 membership functions were used. Table 5.10 shows the system configurations under each system and their required training times. The training time is based on a Pentium 166MMX PC running software developed using a Borland C++ compiler. As the proposed generalised neural-fuzzy system comprises a BPNN and the self-extraction fuzzy rules algorithm, the training time shown in Table 5.10 is the total of the two-step procedure. It can be observed that the number of rules for the proposed generalised fuzzy interpretation system covers all the possible input membership combinations. For example, a five membership fuzzy system with four input variables will have 54, that is, 625 rules. In this way, the generalisation capability of the neural network will be able to predict data that are not covered by the original training data. The prediction accuracy of all the system is shown in Table 5.11. The error measure used for comparison is the Mean Square Error (MSE).

Table 5.10: System configuration and training time

System	Type	Configuration	No. of rules extracted	Training Time
BPNN	BPNN	4 input nodes, 8 hidden nodes 1 output node	NA	24 min
FUZ5MF	Compact Generalised Neural-Fuzzy System	5 membership functions	625 rules	26min
FUZ7MF	Compact Generalised Neural-Fuzzy System	7 membership functions	2401 rules	30min

Table 5.11: Prediction accuracy of all the system

System	Type	Configuration	MSE
BPNN	BPNN	4 input nodes, 8 hidden nodes 1 output node	0.0020
FUZ5MF	Compact Generalised Neural-Fuzzy System	5 membership functions	0.0026
FUZ7MF	Compact Generalised Neural-Fuzzy System	7 membership functions	0.0023

From Table 5.11, the BPNN system has MSE of 0.0020, the 5-membership generalised neural-fuzzy system of 0.0026 and 7-membership generalised neural-fuzzy system of 0.0023. Although the MSE shows that the BPNN has the best prediction results, the generalised system could be fully understood and modified by a user. From the MSE, the generalised neural-fuzzy system has comparatively good prediction results as compared to those from BPNN, but with human understand fuzzy rules presented. Between the different fuzzy systems, the one with the 7-membership performs best. This is because of the increase in the number of fuzzy rules that are used to define the underlying function.

The output plot of the predicted results is shown in Figure 5.11. This shows the compact generalised neural-fuzzy system's performance is comparable to that of the BPNN. It has the generalisation ability of the BPNN and at the same time provides users with human understandable rules. A small selection of the rules is given in Figure 5.12 and the division of the membership functions is shown in Figure 5.13. Table 5.12 shows a summary of comparison for the features of the two test systems

in the case study as well as the self-generating fuzzy rules system. From this table, it can be seen that this proposed compact generalised neural-fuzzy system has all the advantages of the BPNN model and the Self-generating Fuzzy Rules System. In addition, by using the input contribution measure, the user can identify the significant input variables to build the final system so as to reduce the number of fuzzy rules.

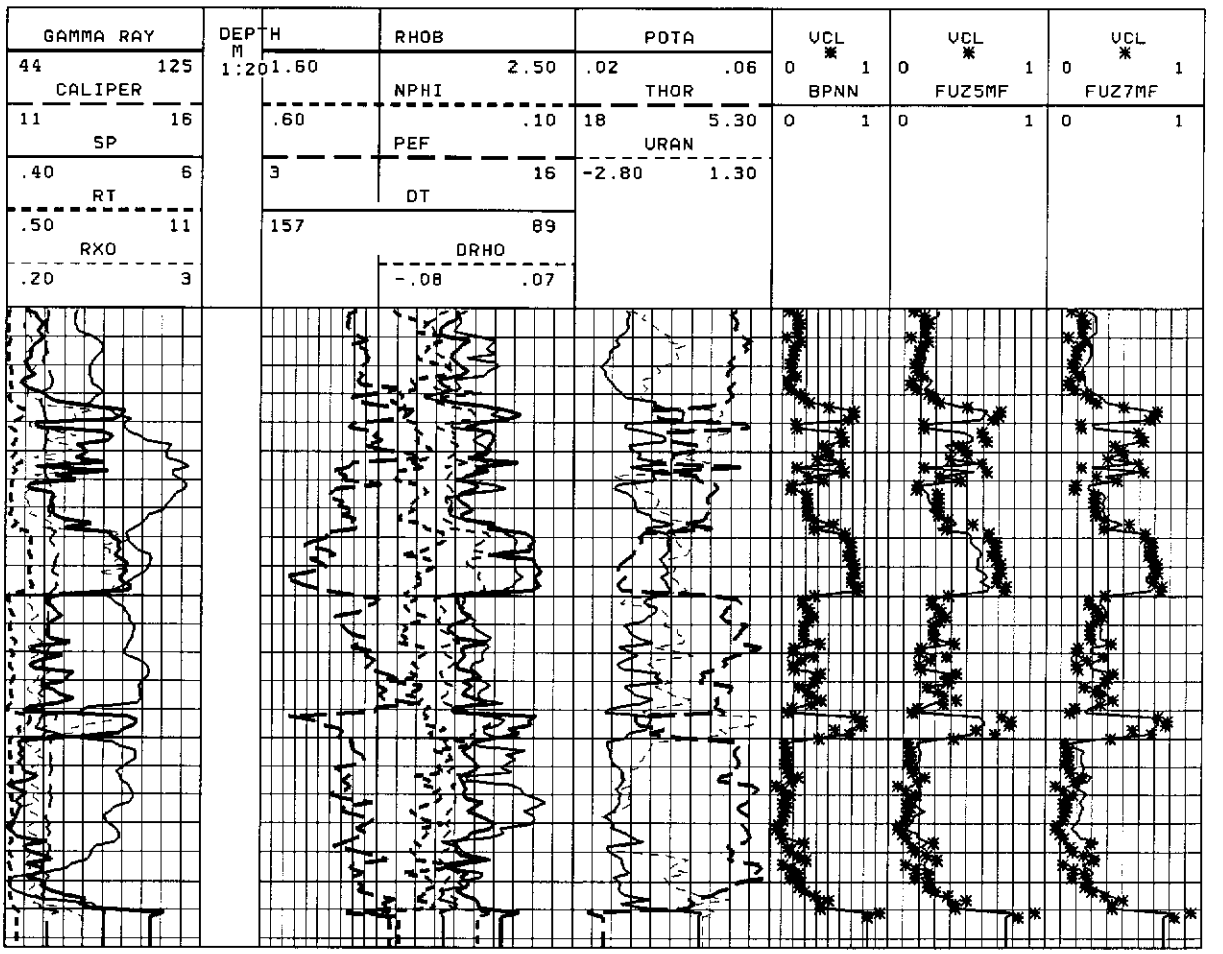


Figure 5.11: Graphical plot of the predicted results and core data.

```
If RHOB= VL and NPHI= VH and RT= VL and GR= VH then VCL= H
If RHOB= VL and NPHI= VH and RT= L and GR= VL then VCL= VL
If RHOB= VL and NPHI= VH and RT= L and GR= L then VCL= L
If RHOB= VL and NPHI= VH and RT= L and GR= M then VCL= M
If RHOB= VL and NPHI= VH and RT= L and GR= H then VCL= H
If RHOB= VL and NPHI= VH and RT= L and GR= VH then VCL= VH
If RHOB= VL and NPHI= VH and RT= M and GR= VL then VCL= VL
If RHOB= VL and NPHI= VH and RT= M and GR= L then VCL= VL
If RHOB= VL and NPHI= VH and RT= M and GR= M then VCL= VL
If RHOB= VL and NPHI= VH and RT= M and GR= H then VCL= M
If RHOB= VL and NPHI= VH and RT= M and GR= VH then VCL= VH
If RHOB= VL and NPHI= VH and RT= H and GR= VL then VCL= VL
If RHOB= VL and NPHI= VH and RT= H and GR= L then VCL= VL
```

Figure 5.12: Examples of the Generalised Fuzzy Rules.

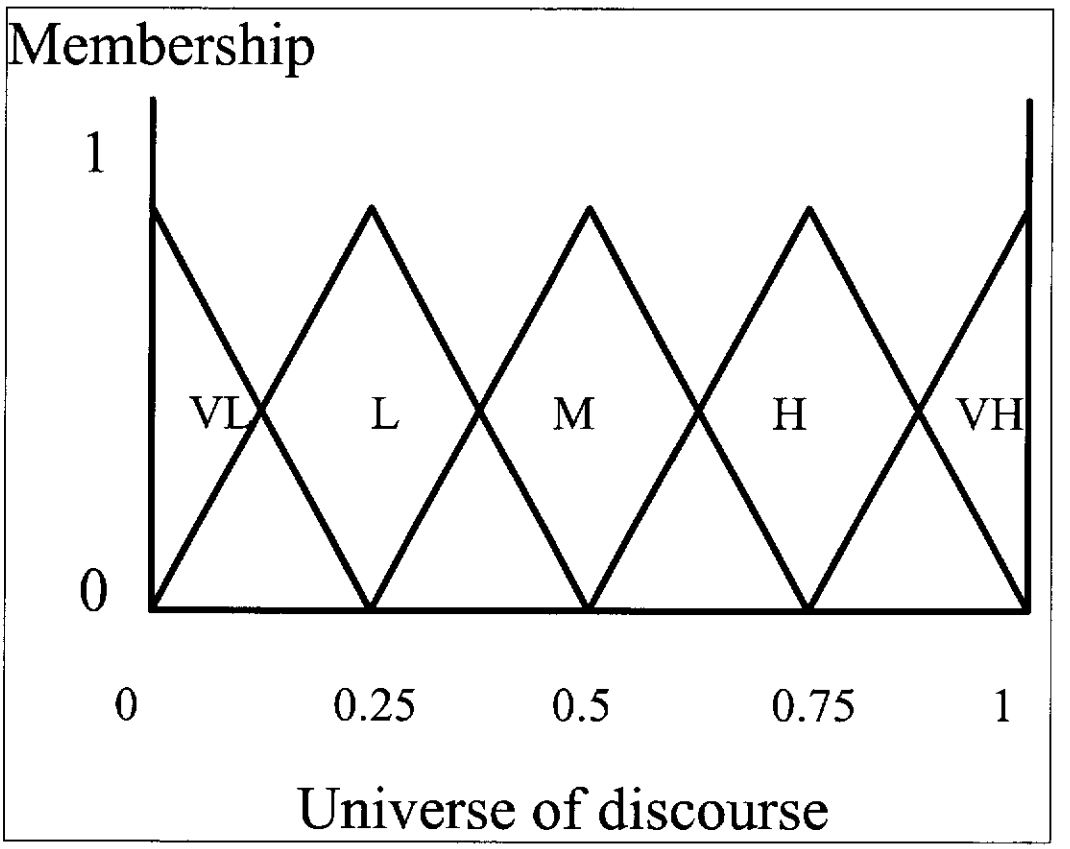


Figure 5.13: Division of the five membership functions.

Table 5.12: Comparison summary

System	Learning ability	Generalisation ability	Noise Rejection	Interpretation of new data	Linguistic Fuzzy rules
BPNN	Yes	Yes	Yes	Yes	No
Self-generating Fuzzy Rules System	No	No	No	No	Yes
Compact Generalised Neural-Fuzzy System	Yes	Yes	Yes	Yes	Yes

5.5 A REDUCED FUZZY RULES BASE SYSTEM

5.5.1 WHY IT IS NECESSARY TO REDUCE THE FUZZY RULE BASE

The compact generalised neural-fuzzy system outlined has addressed the disadvantages of the ANN and FL system. In the compact generalised neural-fuzzy interpretation model, a complete set of fuzzy rules covering all the fuzzy patches in the input-output state space is derived. This set of fuzzy rules describes the generalised underlying function of the available training data. The prediction accuracy of this fuzzy model, however, is dependent on the number of fuzzy memberships. In general, the prediction accuracy will normally improve as the number of membership increases. However, the fuzzy rules will grow exponentially with the increasing number of membership functions. Further, the fuzzy rules will also increase exponentially with an increase in the number of inputs. The formula used to obtain the total number of fuzzy rules of the generalised neural-fuzzy interpretation model can be expressed as:

$$\text{Number of fuzzy rules} = M^I$$

where M = number of memberships

I = number of input parameters

In the case of a data set consists of four input data and seven membership functions, the total number of fuzzy rules will be 2401. With this number of fuzzy rules, it becomes apparent that it is impractical to examine or manipulate the model manually. In addition, the prediction process will take a comparatively long time.

As the purpose of generating fuzzy rules in obtaining the generalised underlying function from the training data is to allow for user interaction, an approach to reducing the number of fuzzy rules is necessary. In this section, a reduced fuzzy rule-base approach to the development of a generalised neural-fuzzy interpretation model is proposed. A case study based on the well log data analysis used in petroleum industry illustrates the proposed method. The results show that the number of fuzzy rules for the testing well can be reduced by up to 95%. It is also shown that the prediction accuracy is preserved as compared to the compact generalised neural-fuzzy interpretation model.

5.5.2 REDUCED FUZZY RULES BASE APPROACH

For most cases, the data set under investigation will not utilise all the fuzzy rules in the generalised interpretation model. As the input data are normally obtained from specific sample within the population, the characteristics of the sample will normally cover part of the generalised underlying function. Based on this property, the fuzzy rule-base of the interpretation model can be reduced by examining the number of times that a fuzzy rule in the generalised fuzzy system is fired. This will lead to the development of a unique and compact rule-base for each sample as described below.

Step 1. A compact generalised neural-fuzzy interpretation model with the complete generalised fuzzy rule-base is set up. The rule-base is named as the Generalised Fuzzy Library.

Step 2. Input data from a new data set are applied to the model after the normalisation and fuzzification processes.

Step 3. The fuzzy rules fired in the inference process are extracted. This set of rules is the Reduced Fuzzy Rule Base for the particular sample under investigation.

Step 4. The number of times that a rule is fired will indicate the importance of the rule within the rule-base.

Step 5. For different samples within the population, a compact model can be established for each case based on the reduced fuzzy rule-base approach.

As the number of rules in the compact model is much smaller, the inference process will take a much shorter time. In addition, an examination of the reduced rule-base by a data analyst is now feasible.

5.5.3 REDUCED FUZZY RULES BASED IN PRACTICE

A problem of well log data analysis is used to illustrate the approach. It is based on typical data from two boreholes. A total of 289 core data in the first borehole was used as the training data. A total of 140 core data from the second borehole was used for testing. The second borehole was used as a "blind test" to benchmark the performance of the proposed approach. The core or log data from the second well was not been used in the training process in any way. For these two boreholes, data from four well logging instruments were used. The input logs used were bulk density (RHOB), neutron (NPHI), uninvaded resistivity (RT), and gamma ray (GR). The petrophysical property that is of interest in this case is volume of clay (VCL). The presence of clay has an important effect on the permeability that in turn governs the ease of extraction of fluid. In addition, it also affects the log readings. It is therefore one of the important properties in well log interpretation. All data was normalised between 0 and 1.

Two tests were carried out, the first based on the generalised neural-fuzzy well log interpretation model as described in previous section. The second test was based on the Reduced Fuzzy Rule Base method described. The training well data was used to train and extract the fuzzy rules that best described the generalised underlying function. After the fuzzy rules were extracted, they were used to form the Generalised Fuzzy Library for the boreholes around that region. In this case, the Generalised Fuzzy Library with 7 fuzzy membership functions consists of 2401 fuzzy rules. In the first test, the Generalised Fuzzy Library with all 2401 fuzzy rules was used to infer results for the testing well. In the second test, only the fired fuzzy

rules from test 1 were used to infer results for the testing well. These fired fuzzy rules then formed the Reduced Fuzzy Rule Base for the testing well for examination by the user.

For the first test, a generalised neural-fuzzy well log interpretation model was used. A total of 2401 fuzzy rules were generated using 7 membership functions. The division of the memberships is evenly distributed using triangular functions as shown in Figure 5.14. These 2401 fuzzy rule also formed the generalised fuzzy library for boreholes around the region. To determine the performance of the system, the predicted results for the second well using the generalised fuzzy library were compared to the actual core data. The errors were measured in terms of the normalised Mean Square Error (MSE).

In the case of the second test, log data from the second well were applied to the generalised fuzzy library. The number of times a fuzzy rule fired was recorded. Those fuzzy rules fired during the process were then extracted to form the Reduced Fuzzy Rule Base for the second well. Prediction results are inferred only from the reduced fuzzy rule base. The MSE of the second approach was also calculated.

Table 5.13 gives a summary comparing the results obtained from the two tests. For the reduced fuzzy rule-base approach, the number of fuzzy rules used to infer the predictions for the testing well was 124. This is greatly reduced from the original Generalised Fuzzy Library as used in Test 1. With the set of reduced fuzzy rule base and the information about how many times that the rules were fired, the log analysts may focus their attention to those rules that are deemed to be important. In terms of

accuracy, as the rules in the second approach are identical to those in the generalised fuzzy library, the prediction accuracy is not affected at all. In terms of execution time for the inference process, the reduced fuzzy rule-base approach is also much shorter. The processing time is based on a Pentium 166-MMX Personal Computer where the program employed was developed in the C++ programming language.

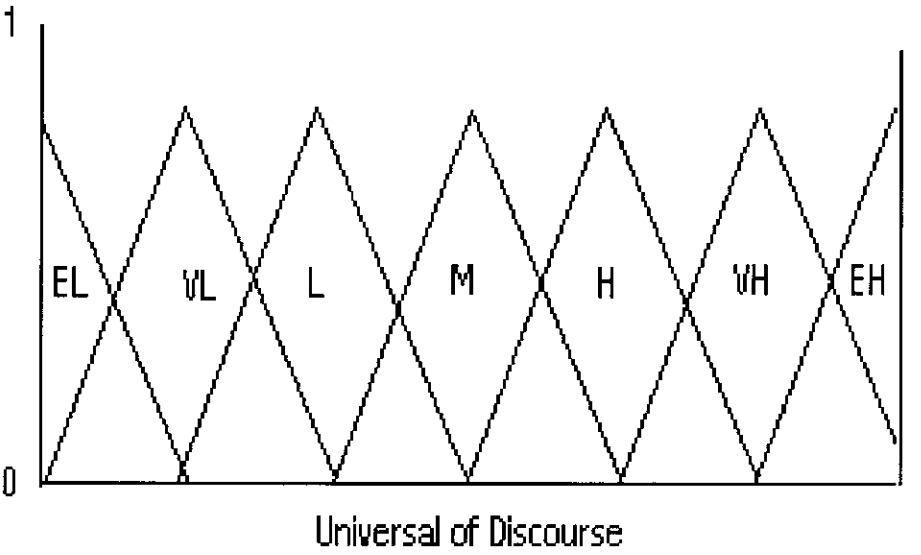


Figure 5.14: Distribution of Membership Functions

Table 5.13. Summary of results from the two approaches.

Approaches	No. of rules	"Blind" Test MSE	Processing Time
Generalised Neural Fuzzy Model	2401	0.00237	4 minutes
Reduced Fuzzy Rule Base	124	0.00237	10 sec

As the reduced fuzzy rules are used to infer results for the testing well, the log analysts can manipulate this set of rules based on their experience and knowledge of the borehole. In turn this will modify the prediction characteristics of the testing well. Figures 5.15 and 5.16 show the crossplots of the results obtained from the

generalised neural-fuzzy model and from the reduced fuzzy rule-base respectively. It can be observed that there is no difference in the predictions obtained from the two systems.

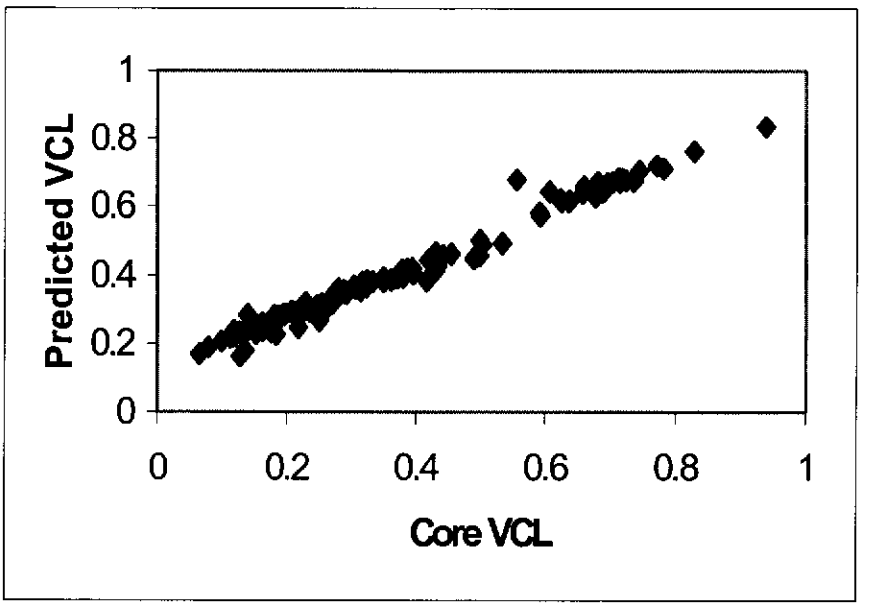


Figure 5.15: Crossplot of the predicted results from Generalised Neural-Fuzzy Model vs Core VCL

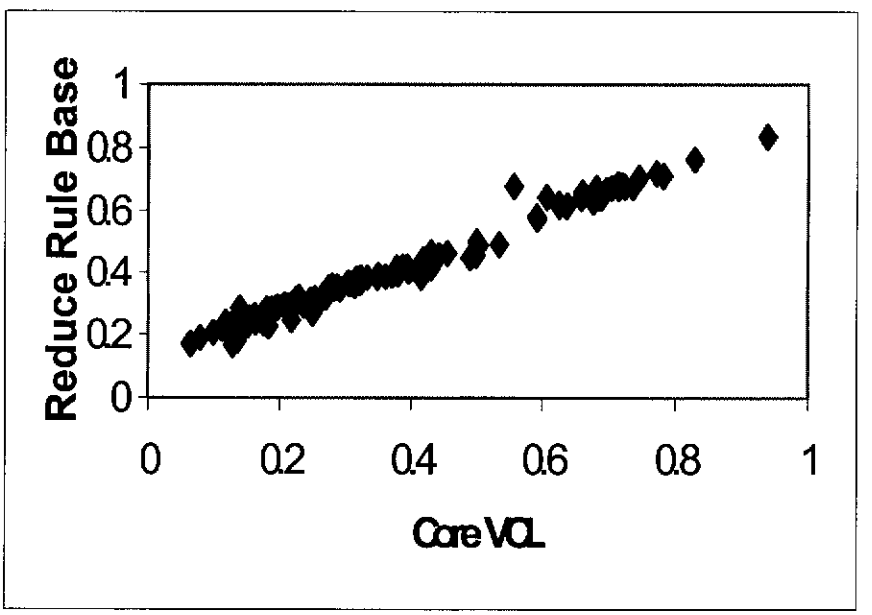


Figure 5.16: Crossplot of the predicted results from the Reduce Fuzzy Rule-Base vs Core VCL

5.6 A COMPLETE DATA ANALYSIS PACKAGE

5.6.1 THE OVERALL APPROACH

Before using the overall data analysis approach outlined, the analyst has to first decide on using a single ANN or Modular Neural Network by examining the size of the data. Normally, for a large volume of available training data, it can be assumed that the underlying function is heterogeneous. In this case, a Modular Neural Network should be used. Figure 5.17 shows the flow diagram of how this data analysis model can be built, and Figure 5.18 shows how the constructed model can be used to predict data.

After determining the type of ANN to be used, the SOM data splitting technique is used to divide the available data into training and validation sets. By using these sets together with the early-stopping validation technique, an analyst can ensure the generalisation capability of the ANN. In cases where the training data are not evenly distributed, that is, with bias, interactive reinforcement learning can be used to control the training process. This will ensure that the ANN can recognise the minorities that are significant in the training data, and effectively allow the analyst to control how the ANN should learn.

If the available input parameters are large, the straightforward input contribution measure can be used to identify the significant input parameters required to predict the particular output variable. This effectively reduces the number of fuzzy rules extracted by the compact generalised neural-fuzzy system, and may also improve the prediction results in some instances. After the generalised ANN data analysis model

has been built, data are generated for the self-generating fuzzy rules algorithm to extract human understandable rules for building the generalised fuzzy rules base. When the prediction model is used to predict a sample in the population, the reduced fuzzy rules base that best describes the function used in the prediction can then be constructed. With this reduced fuzzy rules base, an analyst can shorten the time to examine the prediction model in order to modify or add-on knowledge and so improve the prediction results for that sample.

Building of the data analysis model.

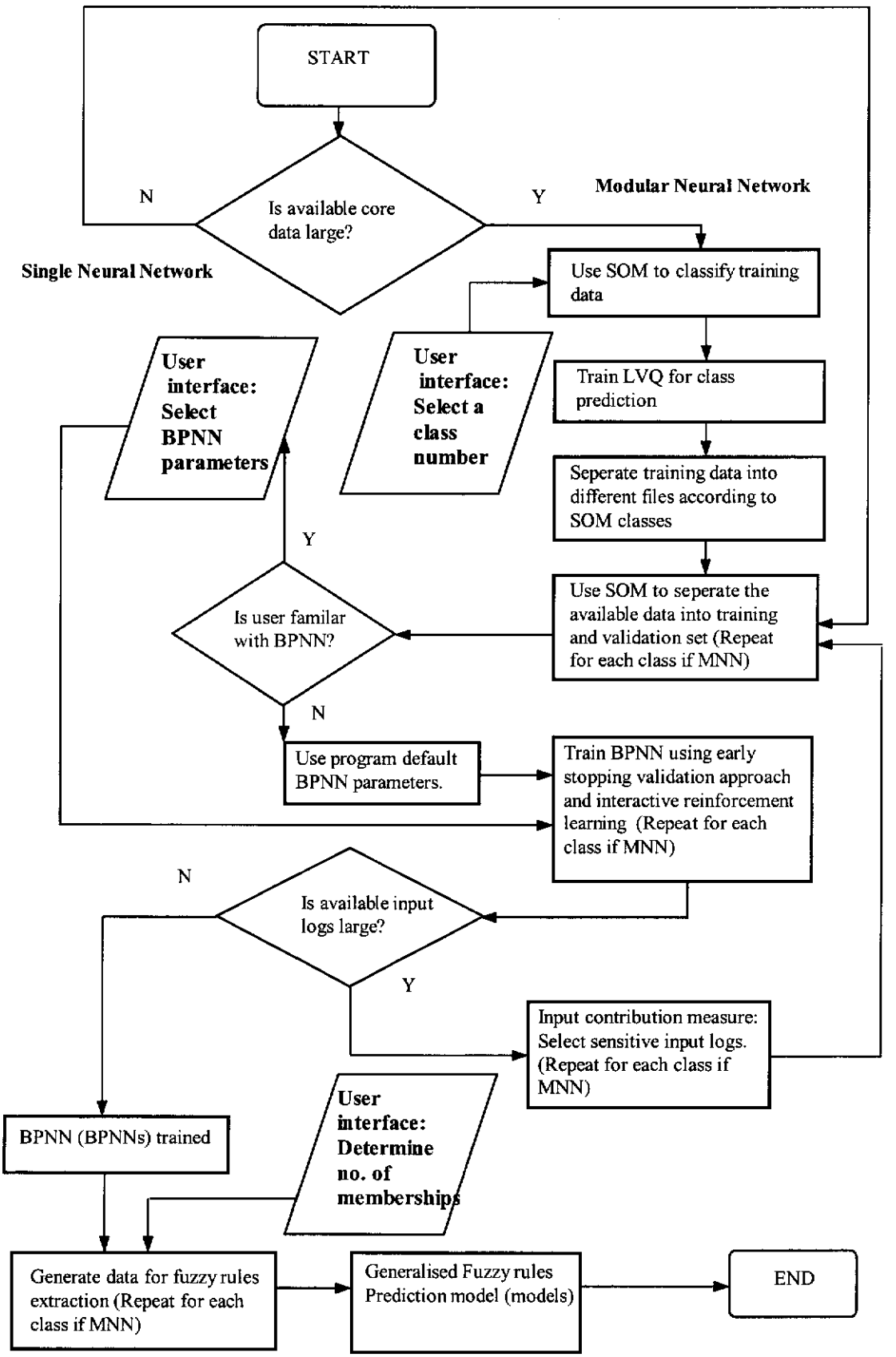


Figure 5.17: Building of the data analysis model

Predicting output from the built interpretation model

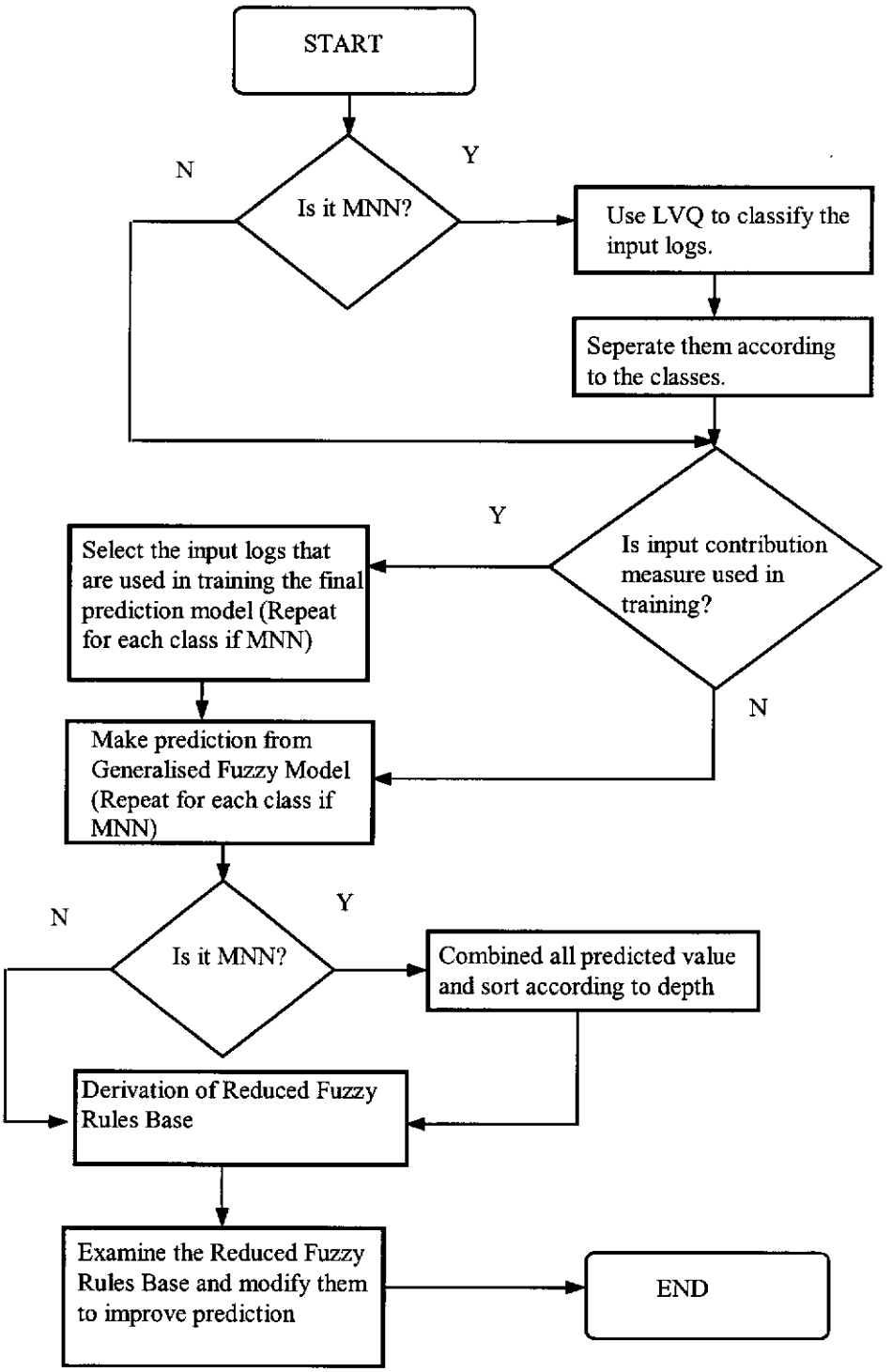


Figure 5.18: Prediction algorithm

5.6.2 A COMPARISON WITH OTHER APPROACHES

The data analysis approach outlined offers the support that data analysts require and it is also more robust in most respects than traditional statistical, Artificial Neural Network, Fuzzy Logic and Neuro-fuzzy approaches. Table 5.14 summarises a comparison between this data analysis approach and traditional approaches based onthe desired characteristics of a data analysis tool. Table 5.14 suggests this data analysis approach is both useful and significant. The traditional statistical approach used in the comparison is a non-parametric multiple regression analysis using predictive learning. The traditional Neuro-Fuzzy approach refers to approaches that incorporate neural network and fuzzy logic as a hybrid system.

Most of the approaches in the comparison table have a self-learning ability except for the FL approach. FL normally requires the analyst to perform intensive study on the available data and derive the relationship - and so fuzzy rules - manually. This is a very tedious and time-consuming task. Hence FL is normally considered to lack a generalisation capability, unless the data contains straightforward underlying functions without noise. Most approaches under comparison, except FL, would normally incorporate some kind of validation technique to ensure their generalisation capability. However, in this comparison, the reliability of their generalisation capability is not to be taken into consideration.

When using a data analysis tool, a novice analyst would like to spend minimum time in gaining familiarity. In this case, ANN and this proposed approach stand up well

above the other. The analyst just needs to know a few initial parameters, and the tool will undertake learning by itself.

Table 5.14: Summary of Comparisons

	Traditional Statistics	ANN	FL	Traditional Neurofuzzy	New approach
Self-learning	Yes	Yes	No	Yes	Yes
Generalisation	Yes	Yes	No	Yes	Yes
Time spend on understanding how to use the data analysis tool	Very long	Short	Long	Moderate	Short
Self-explained	Not quite	No	Yes	No	Yes
Ease of modification and add-on	No	No	Yes	No	Yes
Controlling how the model is built	Yes	No	Yes	No	Yes
Handling of very large amount of data	No	Yes	No	Yes	Yes
Handling of very large amount of input variables	No	Yes	No	No	Yes

After the data analysis model has been constructed, how well the model can be understood by the analyst is considered by the self-explained feature in the comparison table. For this feature, FL and this proposed approach stand out from the others as they both provide near-natural language fuzzy rules. The statistical approach is not considered as self-explained because an analyst needs to have strong statistics background in order to derive and understand the model. With the availability of the human understandable fuzzy rules in FL and this approach, it is easy for an analyst to modify or add-on to the prediction model. The analyst could also control how the analysis model should predict. However, the analyst would also

like to control how the prediction model is formulated, ANN and traditional Neuro-Fuzzy data analysis approaches normally do not allow any such interaction.

When there is a very large volume of available training data, statistical approaches and FL will normally have some difficulty in handling them as these techniques required assumptions to be made and the detailed analysis of the data. A large available training data volume is very tedious for an analysts to fully process. As the complexity of the prediction model is a function of the input variables, statistical approaches, FL and even Neuro-Fuzzy techniques could have difficulty in deriving a prediction model. In cases where there are a large number of input variables, the learning of the underlying function could be handled well with the ANN and the approach examined here.

CHAPTER 6: A NEW APPROACH TO DATA ANALYSIS

6.1 INTRODUCTION

A data analysis approach has been outlined that combines the best of ANN and FL approaches together with new techniques. This research has achieved the principal objective of producing a desirable data analysis model providing accurate and reliable results that is capable of self-learning and which is self-explained. It also meets the objective of being an automatic data analysis approach, as minimum users' training and understanding of the data analysis model are required.

This new data analysis approach copes with problems where the primary needs are the following:

1. Robustness in the data analysis model

2. An ability to cope with very high data volumes.

3. A need to include external knowledge of some form into the analysis process.

It is questionable if data analysis of the character currently desired in many practical applications is possible by a single method. A more sophisticated approach based on the integration of a number of methods is becoming essential and that is one of the prime characteristics of the data analysis system put forward. By far the most novel feature, however, is that it allows a varying degree of human intervention in the

analysis ranging from none whatsoever by an inexperienced analyst to a very strong interactive involvement if this is seen as producing worthwhile outcomes.

This approach is suitable for applications that require quantitative, non-parametric, non-linear, inferential analysis using predictive learning. It stands up well against other data analysis approaches such as conventional statistics, ANN analysis, Fuzzy Logic analysis, and conventional Neuro-Fuzzy logic analysis.

As shown in chapter 5, this approach embodies all the advantages of other common approaches. It has the self-learning and generalisation capabilities of conventional statistical, ANN and Neuro-Fuzzy data analysis methods. Further, it resembles ANN in the sense that the effort needed to employ the system effectively is minimal, but at the same time it offers the advantages of fuzzy logic systems in offering self-explanation features. That allows easy modification of the knowledge base utilised for analysis, and the inclusion of prior knowledge if this is important to the problem.

This data analysis system copes well with all three of the issues outlined. It has been tested with real world problems drawn from the resource industry and shown to offer significant benefits. This research has also initiated the use of ANN and FL in the field of hydrocyclone control data analysis in mineral processing. At the same time, it has improved the analysis process in well log data analysis in petroleum exploration.

6.2 ROBUSTNESS OF THE NEW APPROACH

In any data analysis problem, the essential objective is to extract information where this may be interpreted as meaningful data that aids human understanding of some problem. It is always possible to assume that the sample under analysis would normally provide enough information to the analyst to infer the rest of the population of interest. However, in most practical cases, as indicated by the two applications presented in this research, many factors could distort the sample in establishing the data analysis model. If these factors are not handled well in establishing the data analysis model, the objective of the data analysis may not be met.

The problem, though, is that data in most practical problems is subject to noise and possibly nonlinear distortion, and there is no simple remedy to this at source. In this circumstance, a data analysis model is required that can successfully reject the noise - or at least minimise its impact - and overcome nonlinearity.

Noise can be due to human errors, incorrect mapping between input and output variables, or experimental errors. The main advantage of neural networks is the ability to reject noise. However, as has been shown in this research, if the training of the neural networks is not handled well, a poor outcome may result.

Although validation is the most common way to ensure generalisation capability of the data analysis, a problem has been discovered in this research. However, SOM data splitting better ensures that the training data set covers the whole sample space and that the validation set can give a better indication of the generalisation function

Besides eliminating the noise in establishing the data analysis model, the distribution of the available data is also an important factor in the success of the data analysis. Depending on the nature of the problem, the data collected may not be evenly distributed. If some of the significant data collected only represents a small portion of the total available sample, they could be treated as noise when the validation technique is used. This research has also recognised the fact that sometimes it is not possible or feasible to obtain more training data to reinforce the minorities. The Interactive Reinforcement Learning has shown to be successful in overcoming this problem.

6.3 DEALING WITH HIGH DATA VOLUMES

The need to cope with large data volumes is self-evident in a modem context. The increasing use of automated measurement systems means there are no longer limits in many cases on data gathering.

This data analysis system copes well with high data volumes through a simple process of modularisation. With the advancement of technology, some applications could result with a large pool of data that could be used for establishing the data analysis model. Under this kind of situation, it would be very tedious and time consuming for any data analysis technique to fully realise the underlying functions. One assumption has to be made under this kind of situation. It is assumed that the underlying functions are very complex.

In an inference system, clustering can break down the underlying function into different regions before establishing the data analysis model. However, most analysts would like the clustering to be done as easily as possible, so that the overall analysis time would not be increased but at the same time accuracy is ensured. This research has recognised this need, and the SOM and LVQ techniques used in the Modular Neural Network (MNN) have been shown to be very successful in performing a break down on the large data set.

It is reasonable to expect that the number of input variables for a typical problem will increase in the future. If all the input variables are used, this can result in a very complex data analysis model. In most cases, the complexity of this model is a function of the number of the input variables. In cases where the input variable has little implication on the results, it can have a negative effect. It would affect the accuracy of the model. The Input Contribution Measure discussed has handled this issue well by identifying those significant input variables required for the prediction of the results.

6.4 HUMAN INVOLVMENT IN THIS DATA ANALYSIS SYSTEM

The final need is a little unusual and a feature of this method. In many problems, while the specific numerical outcomes of the analysis are, of course, not known, the form of the solution is from either past experience or theoretical considerations. In that case, an analyst may wish to influence the development of the solution by including external information, or even to make the data analysis process interactive.

The method presented here is automatic, as it requires minimum decision and assumption making by the analyst when building the data analysis model. At the same time, it requires little knowledge on the analyst's part concerning the methods employed, and the time taken to learn how to use this data analysis tool is minimal. For any non-experienced user, the tool can automatically select the initial parameters it requires. Further, the tool includes techniques that cope with the generalisation issue in learning.

If, though, the user wishes to strongly interact with the analysis, then this tool provides a number of features to allow it. It is a self-explained data analysis system in the sense that it offers reasons for its analysis through a listing of human understandable fuzzy rules that permit an analyst to understand the model. Further, if desired the analyst can influence the further processing of the data by either modifying or omitting any of these rules, or by including additional rules.

Understanding how the model performs an analysis of the problem is important in many instances. It allows an analyst to generate deductions from the analysis, justify the feasibility of the analysis, and verify the accuracy of the predictions. This adaptive feature is important. When the model is applied in different situations, naturally different requirements will arise.

6.5 FURTHER ADVANCEMENT

Although this data analysis approach has made use of statistical theory in realising the factors that affect the generalisation capability of the Artificial Neural Network

(ANN), statistics is not incorporated in any way to enhance the data analysis model. Statistical theory has existed for many years, and is now a rich and sophisticated body of knowledge. It would be useful to incorporate some of that with the data analysis approach developed here.

This might be done when analysing the available training and prediction data. Statistical methods can be used to examine whether the predicted data sample is similar to the training data, or the significance of the training sample in inferring the population under study. This can greatly assist the analyst to understand why a prediction is not as good as expected. In addition, some of the confidence tests used in statistics may also be used to provide some indication on the built model. This research has already combined the advantages of using the ANN and Fuzzy Logic (FL). It would be fruitful if more advanced statistical techniques could also be incorporated.

This research highlights the need to solve the problem of the "curse of dimensionality" in the fuzzy rules. As FL makes use of fuzzy patches to cover the fuzzy function approximation, the smaller the patches, the better the prediction. However, this leads to rule explosion. The number of fuzzy rules involved is directly related to the number of input variables and the number of memberships derived in the fuzzy systems. The fuzzy rules would grow exponentially with the increase of the input variables and the membership functions. Moreover, as the number of fuzzy rules increase, the time taken to infer results may also increase. This is normally caused by the inefficiency of the search algorithm in finding the appropriate fuzzy rules to fire. Some kind of optimising technique, compression technique or

identification algorithm for highlighting the significant rules may need to be studied in order to solve this problem.

As data warehousing is an emerging technology, the incorporation of a data mining algorithm would be an added advantage to the data analysis model. The main objective would be to allow knowledge re-use. Analysis done in one situation could act as a guideline, experience or even underground rules for any new analysis. This could allow the data analysis model to have self-adaptive and self-adjusting features. This is important, as most of the analysis done for a problem in a given environment would normally be applicable to any new situations in a similar environment.

REFERENCES:

Abe, S., and Lan, M.S. (1995) "Fuzzy Rules Extraction Directly from Numerical Data for Function Approximation", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25 (1), pp.119- 129.

Abu-Mostafa, Y. (1990) "Learning from Hints in Neural Networks", *Journal of Complexity*, 6, pp.192-198.

Asquith, G.B. and Gibson, C.R. (1982) *Basic Well Log Analysis for Geologists*, The American Association of Petroleum Geologists.

Baldwin, J.L., Bateman, A.R.M., and Wheatley, C.L. (1990) "Application of Neural Network to the Problem of Mineral Identification from Log Wells", *The Log Analyst*, vol. 3, pp. 279-293.

Baum, E.B., and Haussler, D. (1989) "What Size Net Gives Valid Generalisation?" *Neural Computation*, 1, pp. 151-160.

Bradley, D. (1965) *The Hydrocyclone*, Pergamon Press Ltd.

Bruce, C.H., and Robert, G.C. (1994) *Neural Nets: Applications in Geography*, Kluwer Academic Publishers.

Cheng, B., and Titterington, D.M. (1994) "Neural Networks: A Review from a Statistical Perspective", *Statistical Science*, 9, pp. 2-54.

Cherkassky, V., Friedman, J.H., and Wechsler, H. (1994) *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*, Springer-Verlag.

Coates, G.R., and Dumanoir, J.L. (1974) "A New Approach to Improved Log-derived Permeability", *The Log Analyst*, vol. 15, no. 1, pp. 17-31.

Condert, L., Frappa, M., and Arias, R. (1994) "A Statistical Method for Litho-facies Identification", *Journal of Applied Geophysics*, 32, pp. 257-267.

Crain, E.R. (1986) *The Log Analysis Handbook Volume 1: Quantitative Log Analysis Methods*, Penn Well Publishing Company.

Doventon, J.H. (1986) *Log Analysis of Subsurface Geology - Concepts and Computer Methods*, John Wiley & Sons, U.S.A.

Friedman, J.H. (1994) "An Overview of Predictive Learning and Function Approximation", in *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*, Springer-Verlag, pp. 1-61.

Garson, G.D. (1991) "Interpreting Neural Network Connection Weights", *AI Expert*, pp. 47-51.

German, S., Beinenstock, E., and Doursat, R. (1992) "Neural Networks and the Bias/Variance dilemma", *Neural Computation*, 4, pp. 1-58.

Gupta, A., and Eren, H. (1990) "Mathematical modelling and on-line control of Hydrocyclones", *Proceedings Aus. IMM*, 295 (2), pp. 31-41.

Hardle, W. (1990) *Applied Non-parametric Regression*, Oxford University Press.

Jang, R.J. (1993) "ANFIS: Adaptive-Network-Based Fuzzy Inference System", *IEEE Transactions on System, Man, and Cybernetics*, vol. 23 (3), pp. 665-684.

Kapadia, S.P., and Menzie, U., (1985) Determination of Permeability Variation Factor V from Log Analysis, *SPE-14402: Society of Petroleum Engineers*.

Kartalopulos, S.V. (1996) *Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications*, IEEE Press.

Kearns, M.J. (1994) *An Introduction to Computational Learning Theory*, MIT Press.

Kelsall, D.F. (1952) "A Study of the Motion of Solid Particles in a Hydrocyclone", *Transactions of Institution Chemical Engineering*, 30, pp. 87-104.

Kohonen, T. (1989) *Self-Organization and Associative Memory*, 3rd Edition, Springer-Verlag, Berlin.

Kohonen, T. (1990) "The Self-Organising Map", *Proceedings of the IEEE*, Vol. 78, No. 9, September, pp. 1464-1480.

Kohonen, T., Kangas, J., Laaksonen J. and Torkkola K. (1992) "LVQ_PAK: A program package for the correct application of Learning Vector Quantization algorithms", *Proceedings of the International Joint Conf. on Neural Networks*, June, pp. I-725-730.

Kohonen, T. (1995) *Self-Organising Map*, Springer-Verlag.

Kosko, B. (1997) *Fuzzy Engineering*, Prentice-Hall.

Kovach, W.L. (1993) *Multivariate Statistics Package User Manual.*

Lawrence, S., Giles, C.L., and Tsoi, A.C. (1996) "What Size Neural Network Gives Optimal Generalisation? Convergence Properties of Backpropagation", *Technical Report UMIACS-TR-96-22 & CS-TR-3617*, Institute for Advanced Computer Studies, University of Maryland.

Lin, Y., and George, A.C. (1995) "A New Approach to Fuzzy-Neural System Modeling", *IEEE Transactions on Fuzzy Systems*, vol 3 (2), pp. 190-197.

Lynch, A.J., and Rao, T.C. (1975) "Modelling and scaling up of Hydrocyclone classifiers", *Proceedings Eleventh International Mineral Processing Congress*, pp. 245-270.

Lynch, A.J. (1977) *Mineral Crushing and Grinding Circuits*, Elsevier Scientific Publications.

MacLachlan, G.J. (1992) *Discriminant Analysis and Statistical Pattern Recognition*, Wiley Publications.

Marett, G., and Kimminau, S. (1990) "Logs, Charts, and Computers: The History of Log Interpretation of Modeling", *The Log Analyst*, November/December, pp. 335-353.

Mendenhall, W., and Sincich, T. (1992) *Statistics for Engineering and the Sciences*, 3rd Edition, Dellen Publishing Company.

Mizrahi, J., and Cohen, E. (1966) "Studies of some factors influencing the action of Hydrocyclone", *Inst. Min. Met.*, pp.C318-330.

Moody, J. (1992) "The Effective Number of Parameters: an Analysis of Generalization and Regularization in Nonlinear Learning Systems", *Advances in Neural Information Processing Systems*, pp. 847-854.

Nauck, D. (1995) "Beyond Neuro-Fuzzy: Perspectives and Directions", *Proceedings of the Third European Congress on Intelligent Techniques and Soft Computing*, pp. 1159-1164.

Nelson, M.C., and Illingworth, W.T. (1991) *A Practical Guide to Neural Nets*, Addison-Wesley.

Osborne, D.A. (1992) "Neural Networks Provide More Accurate Reservoir Permeability", *Oil and Gas Journal*, 28, pp. 80-83.

Phipps, M.C., and Quine, M.P. (1998) *A Primer of Statistics: Data Analysis, Probability, Inference*, 3rd Edition,Prentice Hall.

Plitt, L.R. (1976) "A Mathematical Model for Hydrocyclone Classifier", *C.I.M. Bulletin*, 69(776), pp. 114-122.

Plutowski, M., Sakata, S., and White, H. (1994) "Cross-validation Estimates IMSE", *Advances in Neural Information Processing Systems 6*, pp. 391-398.

Prechelt, L. (1994) "PROBEN1 – A Set of Neural Network Benchmark Problems and Benchmarking Rules", *Technical Report 21/94*, Universitat Karlsruhe.

Rider, M. (1996) *The Geological Interpretation of Well Logs*, Second Edition, Whittles Publishing.

Ripley, B.D. (1993) "Statistical Aspect of Neural Networks", In O.E. Barndorff-Nielsen, J.L. Jensen and W.S. Kendall (Eds) *Network and Chaos: Statistical and ProBabilistics Aspects*, pp. 409-456.

Rogers, S.J., Fang, J.H., Karr, C.L. and Stanley, D.A. (1992) "Determination of Lithology from Well Logs Using a Neural Network", *The AAPG Bulletin*, Vol. 76 No. 5, pp. 731-739.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986) "Learning Internal Representation by Error Propagation" in *Parallel Distributed Processing*, vol. 1, Cambridge MA: MIT Press, pp. 318-362.

Sarle, W.S. (1994) "Neural Networks and Statistical Models", *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, pp. 1538-1550.

Sarle, W.S. (1995) "Stopped Training and Other Remedies for Overfitting", *Proceeding of the 27th Symposium on the Interface of Computing Science and Statistics*, pp. 352-360.

Smith, M. (1993) *Neural Networks for Statistical Modelling*, Van Nostrand Reinhold.

Solla, S. (1993) "The Emergence of Generalisation Ability in Learning", *Advances in Neural Information Processing Systems*.

Stone, M. (1974) "Cross-validatory Choice and Assessment of Statistical Predictions", *Journal of the Royal Statistical Society*, B, 36, pp. 111-133.

Vapnik, V.N. (1995) *The Nature of Statistical Learning Theory*, Springer-Verlag.

Wadsworth, H.M. (1990) *Handbook of Statistical Methods for Engineers and Scientists,* McGraw-Hill Inc., pp. 12.21-12.25.

Wang, P. (1991) *Fuzzy Logic Theory and Applications*, National Chio Tung University, Taiwan.

Wang, C., Venkatesh, S.S., and Judd, J.S. (1994) "Optimal Stopping and Effective Machine Complexity in Learning", *Advances in Neural Information Processing Systems*, 6, pp. 303-310.

Weigend, A., Rumelhart, D., and Huberman, B. (1991) "Generalisation by Weight Elimination with Application to Forecasting", *Advances in Neural Information Processing Systems*.

Weiss, S.M., and Kulikowski, C.A. (1991) *Computer Systems That Learn*, Morgan Kaufmann.

Wendt, W.A., Sakurai, S., and Nelson, P.H. (1986) "Permeability Prediction from Well Log using Multiple Regression", in Lake, L.W., and Caroll, H.B., *Reservoir Characterization*, Academic Press, pp. 181-221.

White, H. (1989) "Learning in Artificial Neural Networks: A Statistical Perpective", *Neural Computation*, 1, pp.425-464.

Wiener, J.M., Rogers, J.A., Rogers, J.R., and Moll, R.F., (1991) "Predicting Carbonate Permeabilities from Wireline Logs Using a Back-propagation Neural Network", *Society of Exploration Geophysicists, 1991 Technical Program*, vol. 1, pp 285-289.

Williams, T. (1994) "Special Report: Bringing Fuzzy Logic & Neural Computing Together", *Computer Design*, July, pp.69-84.

Wong, P.M., Jian, F.X. and Taggart, I.J. (1995a) "A Critical Comparison of Neural Networks and Discriminant Analysis in Lithofacies, Porosity and Permeability Predictions", *Journal of Petroleum Geology*, vol. 18(2), April, pp. 191-206.

Wong, P.M., Taggart, I.J. and Gedeon, T.D. (1995b): "The use of Fuzzy ARTMAP for Lithofacies Classifications: a Comparison Study", *SPWLA 36th Annual Logging Symposium*.

Wong, P.M., Gedeon, T.D., and Taggart, I.J. (1995c) "An Improved Technique in Porosity Prediction: A Neural Network Approach", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33 no. 4, pp. 971-980.

Yao, C.Y., and Holditch, S.A. (1993) "Estimating Permeability Profiles using Core and Log Data", *Proceedings of SPE Eastern Regional Conference*, SPE-26921.

Yu, X.H. (1992) "Can Backpropagation Error Surface not have local minima", *IEEE Trans. On Neural Networks*, 3, pp. 1019-1021.

Zadeh, L.A. (1965) "Fuzzy Sets", *Information & Control*, Vol. 8, pp. 338-353.

Zhang, Y.Q., and Kandel, A. (1998) "Compensatory Neurofuzzy Systems with Fast Learning Algorithms", *IEEE Transactions on Neural Networks*, vol. 9 (1), pp.83-105.

APPENDIX

EXAMPLES OF WELL LOG TRAINING DATA

Well 1

DEPT	FPLC	DT5	RHOM	PEFL	GR	HTHO	HFK	CPER
2384.5	25.70983	127.6295	2.05352	3.79039	108.8282	10.55514	0.04647	1200
2385	24.15843	124.1845	2.05552	3.66891	105.5139	10.42928	0.0468	1080
2386.5	28.74162	135.8735	2.06598	3.75522	92.44421	11.99003	0.04799	760
2388	28.30051	163.6027	2.07553	3.66213	106.1338	11.58234	0.04668	878
2388.5	29.00179	165.1741	2.07275	3.61293	103.4823	12.94101	0.04773	1000
2389.5	29.97439	164.2113	2.07659	3.70124	101.0237	12.88491	0.04856	680
2390.5	31.18223	164.5089	2.07312	3.46471	101.2554	12.36763	0.04783	750
2391	32.46099	163.7203	2.07693	3.2266	100.5139	12.54343	0.04847	683
2391.5	32.83678	162.811	2.10556	3.34587	100.0795	12.67908	0.04949	670
2393	36.39331	161.5774	2.20925	3.83055	98.39657	11.92749	0.05102	730
2394.5	26.80068	161.1354	2.36919	4.3879	100.0538	9.06182	0.04223	213
2396	44.56467	159.1518	2.39352	4.642	101.5655	10.60322	0.04178	99
2396.5	42.26454	160.7753	2.23499	3.77477	104.1641	10.64734	0.04454	57
2397.5	35.987	163.6072	2.12767	3.46546	104.9578	12.63567	0.04578	3.75
2400	38.57444	148.2441	2.12298	3.32466	103.3136	11.02678	0.0442	479
2400.5	36.23676	140.7351	2.11538	3.36057	105.9651	11.31663	0.04379	456
2402	40.50702	119.7322	2.10875	3.77632	102.9772	11.40214	0.04955	519
2404	36.47886	104.4688	2.11688	3.90013	83.56909	10.96408	0.04996	659
2404.5	38.28829	106.5685	2.1217	3.74942	89.66756	10.89684	0.04969	719
2405.5	38.02789	123.9211	2.11592	3.38349	97.26858	11.24379	0.05254	490
2406.5	35.32311	129.7024	2.10913	3.40037	101.554	10.39445	0.05465	841
2407.5	39.81006	130.0476	2.08435	3.76026	100.5967	10.35458	0.05238	978
2409	40.38274	127.5699	2.08504	3.80364	99.29298	10.64094	0.05308	436
2410.5	37.21211	127.9568	2.09191	3.46264	108.5056	11.0534	0.05164	1090
2411	38.55773	128.0461	2.09243	3.4495	106.0615	10.80573	0.04913	1530
2412.5	36.46567	127.0833	2.09712	3.57044	104.1902	12.01374	0.04769	1530
2414	38.18531	128.0432	2.09475	3.42954	103.8804	11.64647	0.04613	738
2415.5	37.30674	127.2396	2.08527	3.28715	101.2613	11.54598	0.04669	920
2416.5	37.42654	127.0476	2.08474	3.37564	103.0261	11.95632	0.0472	890
2417	36.11405	126.8557	2.08882	3.38028	105.0533	11.94015	0.04742	1300
2419	35.21386	126.5447	2.1156	3.80586	108.7242	13.12063	0.0462	1360
2420.5	42.21754	126.1592	2.12594	3.81187	100.8335	14.0877	0.04614	1030
2422	38.52407	126.5595	2.10742	3.74661	103.3832	14.03065	0.04352	821
2424	43.13388	127.3274	2.1332	3.93999	104.6429	13.95048	0.04698	1720
2425.5	39.11763	127.0491	2.09957	3.65254	102.9692	13.84829	0.04357	836
2427	37.55252	126.5119	2.09067	3.62158	107.9417	13.66085	0.04495	553
2428.5	37.94286	126.2232	2.09999	3.50885	104.4351	16.33503	0.04438	1070
2429	37.10844	126.5089	2.09353	3.38108	104.319	15.84915	0.04328	1190
2430	37.29977	126.4822	2.10044	3.41243	105.2829	15.35676	0.04314	19
2432.5	16.07891	126.7961	2.17954	3.78141	111.0953	18.84766	0.04745	362
2433.5	23.26469	127.4926	2.16776	3.72426	109.023	17.61089	0.04584	131
2434	20.59321	127.6771	2.18052	3.96681	108.2106	17.80061	0.04503	29.9
2435.5	13.60252	126.4673	2.14337	3.50857	111.3867	18.74508	0.04381	133
2438.5	25.23774	117.3348	2.3025	5.43066	122.355	17.16517	0.03584	343

Well 2

DEPT	NPHI	DT	RHOB	PEF	GR	THOR	POTA	KH
2344.5	31.8234	129.8	2.3032	4.8477	121.625	14.7967	0.0581	148.3895
2345	30.9595	134.5	2.2944	4.0898	120.875	14.9935	0.0595	129.9638
2345.5	29.9372	140.1	2.1929	3.4531	119.8125	14.0592	0.0575	184.3427
2346	29.673	147.8	2.094	3.0762	116.75	13.2393	0.0558	241.3442
2346.5	31.5987	160.7	2.0551	3.1152	117	12.9576	0.0552	179.7563
2347	36.1035	157.9	2.0611	3.1035	114.9375	13.4062	0.0566	141.6146
2347.5	39.4973	155.3	2.0717	3.0918	109.4375	13.4707	0.0567	143.2423
2348	41.5862	143.9	2.0752	3.0176	109.4375	13.2802	0.0563	185.1544
2348.5	38.0752	138.8	2.0726	3.0898	113.9375	13.5213	0.0569	209.5713
2349	40.4182	133.7	2.0704	3.1816	123.3125	13.5497	0.0569	164.147
2349.5	37.3875	134.1	2.075	3.2754	130	13.5921	0.0572	156.217
2350	38.5163	133.5	2.0743	3.3008	129.125	13.4982	0.0573	160.2001
2350.5	35.6464	132.9	2.0701	3.2852	129.375	12.8718	0.0566	232.0691
2351	36.6567	133.2	2.0616	3.3027	123	12.7664	0.0566	272.8171
2351.5	36.3114	132.6	2.0563	3.2656	117.0625	12.6287	0.0562	340.7158
2352	37.783	132.4	2.0506	3.2578	111	12.4183	0.0556	394.3295
2352.5	39.1894	133.2	2.0527	3.1113	112.3125	12.4585	0.0558	346.0764
2353	40.3632	133.5	2.0539	3.0781	117.8125	12.4324	0.0559	286.3412
2353.5	40.4579	133.5	2.058	2.9277	123.1875	13.5863	0.0597	177.8476
2354	38.1276	134.2	2.0601	3.0098	117.5	12.6666	0.0579	303.268
2354.5	37.0641	134.8	2.0664	3.0293	117.25	12.4347	0.0576	335.6624
2355	34.9576	135	2.0701	3.1758	109.125	12.3256	0.057	451.5985
2355.5	37.4572	134.8	2.0787	3.2285	116.875	12.6231	0.0575	309.755
2356	39.8439	134.2	2.0877	3.2812	110.25	12.6033	0.0568	327.3916
2356.5	41.6194	133.9	2.0949	3.2559	117.0625	12.5192	0.0565	256.6865
2357	40.3533	133.6	2.0946	3.082	104.25	12.9509	0.0574	334.4174
2357.5	39.0723	133.4	2.0901	2.9902	106.6875	12.9598	0.0571	330.2338
2358	38.1824	133.6	2.0889	2.9531	102.375	13.1858	0.0571	353.5749
2358.5	40.3746	133.8	2.0921	3.0508	105.6875	12.9344	0.0565	314.3037
2359	38.9244	133.9	2.1008	3.1445	108.25	12.8908	0.0562	314.6046
2359.5	38.1942	134.1	2.1035	3.2109	118.0625	13.0961	0.0565	234.6194
2360	36.0135	134.2	2.092	3.1914	120.5625	13.2788	0.0567	232.3286
2360.5	36.4662	135.1	2.0783	3.1562	117.6875	13.5071	0.0569	225.9565
2361	37.9217	135.3	2.0645	3.1152	113.625	13.6338	0.0572	229.2514
2361.5	39.4365	136.4	2.0671	3.2031	112.375	13.4333	0.0571	229.8401
2362	39.6361	136.5	2.0711	3.3555	113.6875	13.5045	0.0574	216.4228
2362.5	38.1035	135.2	2.0812	3.4629	108.125	13.2841	0.0573	298.9472
2363	36.9327	135.8	2.0839	3.5039	110.25	13.6396	0.0583	266.9182
2363.5	37.5817	135	2.0789	3.2715	110.6875	13.5745	0.0588	268.937
2364	37.9183	135.7	2.0722	3.1211	112.25	14.0053	0.0601	221.2356
2364.5	39.7439	136.6	2.069	3.0762	111.375	13.9333	0.0597	205.5683
2365	39.3411	136.5	2.0787	3.2441	116.0625	13.9531	0.0598	183.0535
2365.5	40.5484	136.3	2.085	3.4668	116.5625	14.2713	0.0611	157.3013
2366	37.6175	136.5	2.1091	3.5234	119.75	14.4254	0.0617	154.2131
2366.5	37.1688	136.3	2.1214	3.4629	125.75	14.474	0.0622	130.0112
2367	37.1566	136.2	2.1337	3.4863	125.8125	14.3637	0.0623	133.9189
2367.5	39.1456	136.1	2.141	3.5469	131	13.7977	0.0604	121.8549
2368	41.0485	159.4	2.1349	3.7363	120.625	13.5723	0.06	83.4441
2368.5	41.1575	136.7	2.1538	4.1953	122.5625	13.2886	0.0592	165.2291
2369	40.4513	133	2.2406	5.5039	109.5625	13.1077	0.0587	275.939
2369.5	38.1864	132	2.3948	6.2578	106.1875	13.1672	0.0588	287.5779

2370	35.9084	133.3	2.5012	5.8945	101.375	13.2179	0.0589	291.7811
2370.5	37.8119	95.5	2.4789	5.5859	103.875	14.0598	0.0606	479.1415
2371	38.5985	83.9	2.4126	4.9766	111.375	14.4362	0.0615	477.5863
2371.5	42.5596	88.2	2.3441	4.4375	109.625	14.4478	0.061	402.4848
2372	41.4803	89.5	2.262	3.8223	114.9375	14.5896	0.0604	362.0255
2372.5	43.8892	119.4	2.1905	3.4297	115.9375	14.2685	0.0578	164.8001
2373	42.5951	136.2	2.149	3.1621	126.25	14.2483	0.0564	82.4498
2373.5	41.0224	136.5	2.1188	3.0273	130.375	14.0538	0.0548	85.286
2374	38.2809	137.3	2.0993	2.9883	125.8125	14.577	0.0551	93.2231
2374.5	36.0392	137.1	2.0865	2.9746	123.4375	15.1792	0.0553	91.7697
2375	37.8364	134.4	2.0793	2.8398	122.125	15.5992	0.0554	79.7254
2375.5	38.2468	127.3	2.0726	2.8379	121.875	15.3706	0.0539	101.7385
2376	37.8639	135.7	2.0921	2.8594	124.8125	15.7914	0.0542	62.9952
2376.5	34.7852	133.4	2.1226	3.1016	125	15.7354	0.0537	76.572
2377	35.2534	129.7	2.167	3.332	127.8125	15.9454	0.0537	66.8242
2377.5	36.6042	127.8	2.2	3.5605	130.5	16.6607	0.0545	45.6993
2378	38.7658	127.7	2.2134	3.9551	129.5	16.3493	0.0535	45.4092
2378.5	37.2991	125.2	2.2489	4.5234	133.75	15.8697	0.0509	50.9167
2379	37.0922	117	2.2898	4.2539	125.125	15.5519	0.0495	82.8072
2379.5	38.3091	110.8	2.2693	3.709	123.9375	15.6906	0.0496	91.4432
2380	39.844	109.8	2.1751	3.1328	121.1875	15.7273	0.0493	103.0666
2380.5	40.4826	114.2	2.1004	2.8887	127.6875	16.7389	0.0535	62.0835
2381	37.7307	123	2.0848	2.7949	127.4375	17.205	0.0543	49.5236
2381.5	38.3671	128.2	2.0983	2.8652	129.125	17.0985	0.0539	40.0746
2382	36.091	132	2.1282	2.9219	127	17.2799	0.0539	38.1647
2382.5	36.228	129.1	2.1425	2.9609	131.75	17.357	0.0534	34.4473
2383	35.8427	128.6	2.1632	2.9707	136.375	17.7112	0.0535	27.6345
2383.5	37.9237	128.7	2.1884	2.9453	144.875	17.4852	0.0525	21.3496
2384	40.9253	117.9	2.2071	2.9922	154.125	17.2565	0.052	20.8401
2384.5	42.559	117.7	2.239	3.2324	154.625	16.9613	0.0514	19.9181
2385	42.1896	128.5	2.3165	4.4688	157.5	16.6752	0.0506	14.4369
2385.5	39.0795	125.3	2.4601	4.1641	146.625	16.3605	0.0494	19.7248
2386	35.4295	111.5	2.4924	3.6758	143.25	16.6723	0.05	31.5006
2386.5	35.8359	111.2	2.3746	3.2051	136.625	16.9997	0.0509	38.8858
2387	37.9492	110.2	2.2871	3.4746	131.25	16.8165	0.0507	49.6037
2387.5	38.3502	110.5	2.287	3.6055	132.25	16.0365	0.0493	61.3364
2388	37.7484	110.5	2.308	3.5176	126.625	16.322	0.0501	64.7953
2388.5	36.8089	109.8	2.3001	3.2969	126.5625	16.1347	0.0497	74.5967
2389	37.1415	110.1	2.2703	3.168	122.375	16.1866	0.0508	85.0178
2389.5	36.9499	121.7	2.2265	3.1602	123.125	16.3186	0.052	63.4442
2390	35.8361	125.3	2.1911	3.1094	131.5	16.6207	0.0529	47.4561
2390.5	36.0514	127.6	2.174	3.2891	130.25	16.8114	0.0538	44.8594
2391	34.7635	129.1	2.1651	3.3574	135.875	16.457	0.0534	46.0827
2391.5	35.6133	129.2	2.1567	3.3262	129.875	16.3875	0.0537	53.4675
2392	38.6631	128.7	2.1501	3.2051	129.125	15.958	0.0528	54.4819
2392.5	39.3681	128.7	2.153	3.168	126.25	15.8028	0.0528	59.2239
2393	38.258	127.6	2.1508	3.2168	127.9375	15.8988	0.0533	61.0675
2393.5	34.3268	129.4	2.1534	3.3379	134.75	16.258	0.0543	54.2646
2394	33.1839	129.1	2.1632	3.4258	138.125	16.0317	0.0542	57.1696
2394.5	34.6204	128.4	2.1651	3.5488	136.625	16.0229	0.0547	57.2805
2395	35.7829	128.2	2.1675	3.5	130.75	16.0138	0.0547	62.9669
2395.5	38.1588	127.5	2.1564	3.4043	124.5	16.2351	0.0556	63.4055
2396	37.397	127.3	2.1544	3.2891	122.8125	15.9434	0.0555	78.0392
2396.5	38.3017	127.7	2.1615	3.2441	122.875	16.3101	0.0569	64.9985

2397	37.3928	128	2.1784	3.2676	124.1875	15.8946	0.056	73.149
2397.5	36.36	126.3	2.1824	3.3457	124.0625	16.0802	0.0567	77.2988
2398	35.3877	126.3	2.1736	3.4199	123.875	16.0348	0.057	85.7268
2398.5	36.0098	128.1	2.1702	3.3691	118.875	16.0211	0.0572	91.5075
2399	35.4624	127.4	2.1706	3.377	124.5	15.7123	0.0567	91.5501
2399.5	34.5148	127.4	2.1709	3.2773	122.875	15.9777	0.0573	91.8303
2400	36.0318	126	2.1747	3.4316	124.0625	16.1845	0.0573	78.3304
2400.5	35.9591	127.6	2.1893	3.3887	123.5	16.7509	0.0585	61.5335
2401	39.4253	126.2	2.1991	3.4844	134.25	17.3247	0.0592	32.5007
2401.5	37.2033	127	2.2005	3.3711	141.625	17.3881	0.0583	28.8779
2402	37.9991	127.3	2.204	3.2871	151.75	18.8479	0.0586	14.3415
2402.5	35.2151	127.7	2.2077	3.2324	150.375	18.6478	0.0573	17.1025
2403	38.4653	127.7	2.2133	3.2285	151.875	19.4696	0.0582	11.5898
2403.5	39.4159	127.7	2.227	3.2383	151.75	19.6913	0.0579	10.4145
2404	38.9959	129.6	2.2259	3.2715	147	19.3655	0.0571	11.7119
2404.5	36.4104	127.5	2.2265	3.2832	147.125	19.6342	0.0576	12.733
2405	35.3369	128.3	2.2167	3.4902	138.625	19.5575	0.0571	15.5844
2405.5	35.9395	127.5	2.2238	3.4961	143.625	19.8465	0.0579	13.1031
2406	36.3987	127.6	2.2299	3.6562	142.5	19.5978	0.0578	13.7946
2406.5	37.248	126.3	2.2501	3.793	137.125	14.8377	0.0569	75.1507
2407	35.7224	123.9	2.2695	4.0039	131.625	14.696	0.0569	105.4594
2407.5	34.9273	122.5	2.2747	3.9395	130.125	14.6991	0.0569	118.796
2408	32.8386	122.7	2.2457	3.8496	133.25	14.1987	0.0557	148.1887
2408.5	34.3837	124	2.2156	3.7461	128.75	14.3333	0.0561	148.9396
2409	36.3038	126.7	2.2026	3.7598	127.5625	14.2385	0.0558	136.0167
2409.5	41.2884	129	2.1977	3.8633	122.0625	14.5995	0.0567	102.7043
2410	44.215	129.8	2.203	3.9355	127.8125	14.3503	0.0563	78.8154
2410.5	43.9503	133.8	2.1953	3.7129	125.3125	14.539	0.0566	72.011
2411	40.5401	128.3	2.1856	3.3828	121.9375	14.2032	0.0556	124.394
2411.5	37.3802	127.7	2.1648	3.3281	115.5	13.9007	0.0551	203.2546
2412	39.2187	126.4	2.1625	3.4844	112.5	13.0508	0.0537	276.0769
2412.5	39.6937	127.2	2.1841	3.7773	116.0625	13.0528	0.054	239.1516
2413	39.4101	124.7	2.2244	4.0234	112.9375	13.3429	0.0546	247.1272
2413.5	35.2031	117.3	2.3044	4.9297	112.1875	12.4402	0.0518	429.8177
2414	32.5794	110.7	2.4143	5.8555	109.1875	11.8251	0.0494	599.1857
2414.5	30.7616	112.8	2.487	4.8672	106.625	11.539	0.0487	606.6481
2415	29.9292	117.4	2.4148	4.0898	99.5	11.7326	0.0502	666.5497
2415.5	34.0705	94.1	2.271	3.4961	98.25	12.0655	0.0513	887.7913
2416	36.1846	100.5	2.1667	3.3301	104.75	12.2866	0.0523	732.5217
2416.5	36.5313	101	2.1151	3.3086	108.3125	12.4547	0.0534	696.8019
2417	35.8147	116.2	2.101	3.3477	109.5	12.3397	0.0538	566.2405
2417.5	38.1541	132.5	2.0968	3.4941	107.375	12.4798	0.0545	384.9583
2418	41.9538	135.2	2.1057	3.4336	106.125	12.0219	0.0535	353.5988
2418.5	43.473	127.3	2.1164	3.4512	104.9375	10.4621	0.051	591.7349
2419	43.5595	126.3	2.1443	3.4277	101.3125	11.2131	0.0538	546.373
2419.5	41.3113	125.9	2.1553	3.6074	100.625	11.0674	0.0536	620.3539
2420	40.4361	133	2.1644	3.6465	100.0625	10.9815	0.0533	573.7007
2420.5	41.0761	132.7	2.1682	3.666	105.75	11.622	0.0547	429.2542
2421	41.3347	133.2	2.1859	3.832	108.375	11.3618	0.0538	414.2103
2421.5	42.3305	132.7	2.2026	4.1133	108.5625	11.3524	0.0538	400.3065
2422	39.5093	133.5	2.2106	4.3398	110.0625	11.0348	0.0529	458.8552
2422.5	40.158	133.6	2.2082	4.2695	112.125	10.7214	0.0525	463.5211
2423	39.6581	132.8	2.2014	4.1914	113.5625	10.2473	0.0519	526.4439
2423.5	42.5191	132.8	2.1986	4.0898	108.3125	10.1021	0.052	551.3867

2424	42.4739	133.2	2.1876	3.9648	105.4375	10.1917	0.0529	580.2407
2424.5	43.0898	133.3	2.1726	3.8398	106.3125	10.2315	0.0535	564.4222
2425	43.8927	134.6	2.1699	3.8184	111.875	10.5408	0.0548	448.2883
2425.5	43.1694	133.6	2.1705	3.8945	107.0625	10.5448	0.0553	526.9612
2426	41.4061	132.3	2.1665	3.9316	108.875	9.275	0.0536	733.968
2426.5	39.2516	131.9	2.1553	3.7852	101.25	9.1371	0.0536	898.8781
2427	39.2757	130.1	2.1489	3.6523	107.125	8.8467	0.0531	887.0999
2427.5	38.629	127.9	2.1397	3.5605	106.625	8.8987	0.0529	923.6324
2428	36.5717	126.6	2.1352	3.5762	107.9375	9.5264	0.0547	892.7578

Well 3

DEPT	NPHI	DTCO	RHOM	PEFL	GR	HTHO	HFK	KH
2352.5	44.5902	140.9583	2.0085	4.0285	109.3483	16.593	0.0518	681.5688
2353	44.9013	135.8036	2.0946	4.3138	107.834	15.0881	0.048	568.1135
2353.5	43.8096	131.3958	2.1203	3.6993	109.8934	15.1027	0.049	199.4702
2354	42.9073	127.8199	2.0788	4.1216	111.2591	15.0853	0.05	517.2463
2354.5	41.9425	127.6905	1.938	4.6066	110.9676	14.5518	0.051	1281.529
2355	40.609	130.1771	1.9433	5.0771	110.2952	13.8583	0.0492	1419.765
2355.5	40.7153	135.0551	2.0618	3.9947	108.5664	13.5041	0.051	677.7079
2356	41.6305	137.3854	2.1652	4.2941	107.212	13.8114	0.0482	391.9763
2356.5	44.2427	140.0521	2.1583	4.2824	105.377	13.6633	0.0472	423.5577
2357	45.3989	137.8155	2.1432	4.4718	103.7542	15.0956	0.0464	500.32
2357.5	46.6951	129.6146	2.0924	4.0119	106.2473	16.9567	0.0476	338.5563
2358	49.3649	123.8125	2.0804	4.1168	108.9027	18.1951	0.0487	330.5359
2358.5	52.4663	121.4911	2.0893	4.143	110.8061	19.9523	0.0491	212.2548
2359	53.4635	121.1414	2.0361	5.1663	114.8706	22.0399	0.0501	664.4201
2359.5	50.6244	122.5744	1.6283	4.8998	117.0753	22.4531	0.049	1564.89
2360	44.9915	124.5982	1.6741	5.1905	124.7668	22.4531	0.049	1505.54
2360.5	40.4871	127.0565	1.9834	4.2849	128.2969	23.0125	0.0466	192.2384
2361	39.1984	127.9226	2.1783	3.0354	132.3562	22.3649	0.0452	1.8873
2361.5	38.7682	128.6205	2.3463	4.2298	130.1015	22.457	0.0434	1.7349
2362	39.2798	126.0908	2.3375	3.7507	126.4885	21.1553	0.0422	1.5524
2362.5	39.0846	125.2991	2.3971	5.6297	126.0981	21.6003	0.042	9.1243
2363	39.218	125.1265	2.3613	4.3246	126.4189	21.1652	0.0426	2.301
2363.5	38.8386	126.5417	2.1957	3.5116	128.3548	20.2025	0.0432	4.3305
2364	38.0844	127.6964	2.2244	3.3468	127.2538	19.743	0.0435	2.932
2364.5	37.6301	128.9211	2.1471	4.6105	127.7726	19.2813	0.0437	78.8234
2365	37.9982	130.0327	2.1637	3.8163	126.4331	18.9065	0.0452	15.6449
2365.5	36.4657	130.0833	2.1494	4.1351	129.3125	19.1194	0.0458	30.7051
2366	35.9109	128.8676	2.1077	4.2679	133.1169	19.5858	0.0456	53.0581
2366.5	35.8074	124.5357	2.1668	3.7192	134.8958	19.1727	0.0448	7.5261
2367	36.1734	122.3988	2.1309	3.38	137.2665	20.1336	0.0452	5.1244
2367.5	36.6924	122.8884	2.1097	3.6067	134.0307	21.2508	0.0468	10.1582
2368	35.2279	125.3571	2.1159	3.6976	134.1708	20.9204	0.0478	12.3227
2368.5	36.8008	127.4479	2.1158	3.6082	131.206	20.0982	0.0477	14.9765
2369	36.653	127.4628	2.1055	3.952	130.5597	20.0828	0.0481	37.989
2369.5	37.1922	127.1741	2.1141	3.4065	131.6097	19.1304	0.0474	13.214
2370	35.1992	126.8795	2.1052	3.8418	133.3555	18.5676	0.0475	37.0975
2370.5	35.8452	126.6414	2.0567	4.6047	135.0455	18.9345	0.0472	236.3419
2371	37.0627	126.2485	2.1174	3.8838	133.7957	18.7403	0.047	30.6311
2371.5	38.0345	126.1696	2.1434	3.9787	133.335	17.6474	0.0474	33.9063
2372	37.5913	126.2292	2.1449	4.0806	134.3954	18.0375	0.0465	33.2195

2372.5	36.9307	126.4985	2.1197	4.0621	134.455	17.9953	0.0479	50.1344
2373	36.8249	127.0417	2.1268	3.7729	132.901	17.8903	0.0482	29.2542
2373.5	35.7505	127.4494	2.1392	3.8698	130.8365	17.9668	0.046	31.225
2374	35.2813	127.2812	2.1386	3.2763	127.5534	18.0528	0.0465	12.5902
2374.5	36.3764	127.067	2.152	3.4832	126.862	18.2005	0.0472	15.6375
2375.5	36.966	126.9062	2.114	3.9467	128.6474	20.4488	0.0478	35.738
2376	36.4588	126.9688	2.0993	4.0268	131.507	20.4366	0.0473	42.0953
2376.5	36.3057	127.6518	2.0944	4.5753	132.7412	20.107	0.0475	127.1129
2377	36.0297	127.7976	2.1141	4.1394	136.5224	20.4743	0.0494	30.6706
2377.5	35.0565	127.8869	2.1236	4.1609	136.7215	21.1331	0.0515	25.1108
2378	35.3022	128.0045	2.136	3.7212	136.6582	21.0528	0.0499	8.2252
2378.5	37.5717	125.8304	2.1688	3.9627	136.8573	20.8977	0.0488	7.6969
2379	38.8296	125.939	2.1597	3.4869	135.5704	21.2833	0.0488	4.0773
2379.5	39.4013	125.9568	2.1849	4.1678	137.9059	22.2314	0.0473	5.8575
2380	37.2918	125.8482	2.1535	3.6596	135.706	23.1108	0.0478	3.9518
2380.5	37.5629	125.619	2.1845	3.4399	137.9108	22.5199	0.048	2.2219
2381	38.1345	125.5372	2.1983	3.4917	135.6589	20.6606	0.0473	2.7538
2381.5	38.1805	127.1384	2.2055	3.4739	136.6439	19.9666	0.0485	2.7997
2382	37.1154	129.1027	2.1949	3.7401	133.9569	19.4402	0.0487	5.8352
2382.5	37.8978	130.6696	2.2098	3.3216	134.395	19.2883	0.0494	2.8679
2383	37.5944	132.9152	2.2011	4.8763	128.9408	18.044	0.0487	89.4297
2383.5	39.8697	133.1845	2.1445	3.4483	122.274	16.8511	0.0485	27.6298
2384	40.0412	134.4137	2.1423	5.0079	112.256	15.2001	0.047	665.5573
2384.5	40.7884	134.5104	2.0523	4.3459	104.3735	13.4824	0.0461	949.5594
2385	40.0454	132.811	2.0872	5.3604	100.8014	12.7265	0.0478	1336.341
2385.5	39.3715	133.0923	1.9333	4.0983	98.4443	11.2986	0.0477	1420.138
2386	39.4172	133.0476	2.0757	3.9196	98.0303	11.7525	0.0501	916.8935
2386.5	39.7305	132.5536	2.0332	3.6422	96.9164	11.8311	0.0503	980.637
2387	40.5153	131.2976	2.039	4.4131	98.6085	12.2118	0.0505	1225.734
2387.5	41.6535	126.1622	2.0915	4.479	99.8455	12.7273	0.0504	1043.71
2388	40.4707	124.9464	2.0037	4.3114	101.8044	13.5516	0.0495	1178.08
2388.5	40.1156	124.3378	2.0795	4.1055	100.7623	13.1221	0.048	842.0729
2389	38.4668	123.4673	2.0661	4.4432	103.4815	13.667	0.0484	982.1821
2389.5	38.6455	123.4628	2.0382	3.8561	104.7324	13.7274	0.0494	784.9274
2390	38.1878	124.2946	2.0877	3.8319	106.237	14.4877	0.0504	496.7915
2390.5	38.6469	121.3036	2.1247	3.8448	107.1421	13.5067	0.0485	372.0146
2391	39.229	120.878	2.1359	4.929	107.4072	14.0687	0.0471	829.8019
2391.5	37.6647	120.3125	2.1279	3.6024	108.767	13.7552	0.0467	228.602
2392	38.362	119.5685	2.1337	4.2811	108.4816	13.9285	0.0472	510.3304
2392.5	36.6876	115.2335	2.1314	4.4973	106.9709	13.9417	0.047	679.4718
2393	36.0226	116.2111	2.1237	4.2096	108.6547	13.1401	0.0457	561.5479
2393.5	35.6786	117.1888	2.107	4.078	105.494	13.1401	0.0457	623.3601
2394	37.8506	122.4893	2.115	4.0753	104.2595	12.9664	0.046	622.0894
2394.5	38.5609	125.8413	2.1187	4.0413	98.1696	14.2947	0.0463	635.068
2395	36.8493	129.1934	2.1402	4.1941	96.5898	13.9935	0.0463	681.7765
2395.5	34.6814	131.241	2.1748	4.2625	93.2482	13.7241	0.0449	651.6025
2396	36.4118	133.0695	2.1777	4.6298	96.4124	12.8771	0.0442	812.3757
2396.5	37.8305	135.1552	2.2201	5.0117	99.0841	14.3111	0.0474	706.4843
2397	39.2235	136.5855	2.1141	3.7325	102.295	13.9934	0.0496	445.9944
2397.5	38.9109	135.8538	2.1139	4.0787	101.4125	14.7076	0.0518	615.8798
2398	39.1969	134.9412	2.0887	3.838	100.0193	13.8427	0.0491	654.9655
2398.5	38.9831	134.0286	2.0917	4.0532	100.693	13.8017	0.049	747.4307
2399	40.2778	133.116	2.0884	4.0495	103.1741	13.1636	0.0495	754.1999
2399.5	41.4475	132.2034	2.0845	4.4104	102.5066	12.9972	0.0506	977.388

2400	40.708	131.8396	2.1051	4.2034	99.0829	11.9281	0.0491	923.7972
2400.5	39.605	131.7371	2.1037	3.9333	95.4935	11.6266	0.0491	888.4826
2401	38.2864	131.6346	2.0902	4.3145	99.9474	11.4195	0.0482	1050.927
2401.5	37.7462	131.532	2.1053	4.1166	103.942	10.8993	0.0483	840.0012
2402	37.9593	131.4295	2.116	3.8225	107.4273	10.638	0.0501	601.0807
2402.5	38.0438	131.327	2.1332	4.4814	106.5215	10.5974	0.0502	890.735
2403	37.7735	131.2244	2.1594	4.6236	107.3281	9.4607	0.0491	886.2816
2403.5	36.9761	131.1219	2.1763	5.3868	106.1103	9.3112	0.0484	1191.299
2404	35.9444	131.0194	2.1573	4.8208	104.0364	9.0858	0.0482	1079.293
2404.5	37.168	130.9168	2.1782	4.8125	105.0153	8.8691	0.0455	964.0909
2405	37.75	130.8143	2.1825	4.6662	103.6763	8.6361	0.0448	913.4073
2405.5	38.8	130.7118	2.1655	4.3838	104.4929	8.5463	0.046	841.7522
2406	39.4672	130.6093	2.1834	4.7226	102.9086	8.9568	0.0474	952.0706
2406.5	40.7271	130.5067	2.1473	4.0735	103.0712	9.724	0.0511	744.0908
2407	41.8419	130.4042	2.1415	3.9568	102.6323	10.177	0.0515	699
2407.5	40.9241	130.3017	2.1497	4.2603	103.6499	10.2444	0.0504	795.1604
2408	38.6342	130.1991	2.1327	4.8756	103.9478	9.9582	0.0485	1150.181
2408.5	38.5177	130.0966	2.1204	4.2646	101.7163	10.5548	0.049	944.98
2409	39.7308	129.9941	2.1027	3.7125	98.4338	9.9347	0.0494	839.2264
2409.5	40.354	129.8915	2.1006	3.8251	96.6535	9.8315	0.0497	947.3512
2410	40.0703	129.789	2.0865	3.8188	99.1831	10.19	0.049	923.3358
2410.5	38.7878	129.6865	2.059	3.8875	101.2877	10.1057	0.0471	1013.89
2411	38.0128	129.5839	2.0868	3.2784	103.3255	10.0785	0.0456	528.7665

EXAMPLES OF HYDROCYCLONE TRAINING DATA

Run	Qi	Qo	Qu	Qo/Qu	So/Su	D50S	D50cS	D50F	D50cF
77	422	222	200	1.11	1.181	29	49	29	50
78	388	204	184	1.11	1.089	29	48.5	28.5	47
79	352	179	173	1.03	1.06	27	43.5	26.5	41.5
80	318	153	165	0.93	0.925	21	41	21	41
81	285	123	162	0.76	0.679		39.5	9	39.4
82	247	56	191	0.29	0.306		46.5		47
71	421	261	160	1.63	1.501	17	21	17.8	21.2
72	389	235	154	1.53	1.486	17.4	21.8	18.5	21.5
73	352	209	143	1.46	1.351	17.3	21.7	17.5	22
74	320	182	138	1.32	1.156	16.5	21	17	21.7
75	286	151	135	1.12	0.991	14	21	15.5	21
76	247	99	148	0.67	0.685	8	23	3	21
65	422	278	144	1.93	1.665	13.9	16.8	14.6	17
66	387	251	136	1.85	1.787	14	16.6	14.1	17
67	353	223	130	1.71	1.504	12	16.8	14.1	17
68	318	197	121	1.63	1.252	11.5	17	14.4	17.7
69	285	159	126	1.26	1.126	11.3	17.3	1.5	17.5
70	247	119	128	0.93	0.82	4.5	17.5	8	18
83	421	278	143	1.94	2.11	9	13.5	8.9	12.5
84	388	258	130	1.99	1.804	11	14	11	15.5
85	388	258	130	1.99	1.886	9	14	9	14.5
86	353	230	123	1.87	1.872	9.4	15.5	10	15
87	319	197	122	1.62	1.559	9.5	15.5	9.5	15.5
88	285	166	119	1.4	1.265	7.5	16	9.5	16.5
89	248	129	119	1.08	0.924		17	3	17.5
90	493	242	221	1.23	1.24	28	40	29	40
91	421	225	196	1.15	1.298	28	40.5	27.5	40
92	388	205	183	1.12	1.065	27.5	39.5	27	39
93	353	180	173	1.04	1.004	24.5	37	25.5	38
94	318	159	159	1	0.844	19.5	37.5	22.5	39
95	284	125	159	0.79	0.74	9.5	34	10	34
96	246	51	195	0.26	0.358	0	46		45
97	388	199	189	1.06	1.108	24.5	38	23.5	37.5
98	388	200	188	1.07	1.004	24	36.5	25	37
99	389	203	186	1.09	1.13	25	35	26	36
100	388	208	181	1.15	1.165	26.4	35.8	24.2	36.5
101	389	208	181	1.15	1.148	24.5	36.2	25	35.5
102	388	209	179	1.17	1.21	24.5	33	24	32
103	388	239	149	1.6	1.575	13.5	18	14	18
104	388	237	151	1.57	1.653	13	18	13	18.5
105	388	240	148	1.62	1.739	13.5	18	14	16
106	389	243	146	1.66	1.777	13.5	17.8	14	18
107	389	243	146	1.66	1.768	14	17.2	13.5	17.5

108	388	250	138	1.81	1.639	11	16	11.5	16
109	490	339	151	2.25	2.147	52	67	51.5	67
110	387	259	128	2.02	1.919	49.5	66	49	66.5
111	318	213	105	2.03	2.006	53.5	68	54	70
1112	284	187	97	1.92	1.842	54	71	56	70.5
113	248	157	91	1.73	1.599	52	70	57	70
114	388	269	119	2.26	2.402	58	73	57	72.5
115	388	271	117	2.32	2.346	55	68	56	68.5
116	388	268	120	2.23	2.264	56	68	54	64.5
117	388	266	122	2.18	2.169	55	67	55	67
118	429	338	91	3.71	3.779	20	20.5	19.5	21.5
119	388	308	80	3.83	3.732	21	22	21.5	23
120	317	247	70	3.55	3.423	21	22	21.5	22
121	287	226	61	3.72	3.475	20	21.5	21	22
122	247	196	51	3.81	3.14	21	22	21.5	23.5
123	389	303	86	3.54	3.914	22	23	21.5	22.5
124	391	309	82	3.79	3.843	20	21	20	21
125	389	308	81	3.79	3.686	20.5	22	20	21
126	385	321	64	5	3.976	19.5	20.5	20	21
127	475	337	138	2.44	2.306	38.5	43.5	39	44
128	390	260	130	2.01	2.254	37.5	42.5	36.5	41.5
129	322	210	112	1.87	1.873	38.5	45	40	45
130	287	187	100	1.87	1.767	41	49	41.5	49
131	250	161	89	1.81	1.847	41.5	48	42	49.2
132	382	275	107	2.57	2.648	43	48	42.5	47.8
133	483	362	121	3	3.277	66	75.5	64	75
134	391	281	110	2.56	2.707	64.5	76	62	75
135	317	227	90	2.53	2.725	52	58.5	50	58
136	283	201	82	2.45	2.655	54.5	60	51	59
137	245	170	75	2.27	2.392	50	58	50	57
138	435	377	58	6.5	8.5	15	15.5	15	15.5
139	386	334	52	6.4	7.8	17.5	18	17	17
140	318	273	45	6.1	6.3	17	17.5	16.5	16.5
141	290	248	42	5.9	6.1	18	18	19	19
142	249	210	39	5.4	4.7	19	19	19	19
143	386	339	47	7.2	9.2	19	19	17.5	17.5
144	420	373	47	7.9	9.9	20	20	19.5	19.5
145	389	346	43	8	8.5	20	20	20	20
146	320	291	29	10	7.8	21	21	23.5	23.5
147	285	256	29	8.83	7.808	22	22	23	23
148	246	219	27	8.11	7.348	24	25	24	25
149	387	230	157	1.47	1.452	15	19	15	19
150	387	241	147	1.64	1.756	22	26	21	25

151	387	253	134	1.89	1.832	31	34	34	34
152	249	155	64	1.65	1.454	30	35.5	30.8	36
153	387	247	140	1.76	1.72	7	13.5	7.5	13.5
154	391	284	107	2.65	2.73	11	12.5	11	14
155	390	329	61	5.39	5.573	16	17	16	17
156	246	200	46	4.35	3.91	16	18	18	19
157	578	300	278	1.08	1.378	21	26.5	19	25.5
158	518	267	251	1.06	1.355	22	27	19.5	26
159	456	220	236	0.93	1.329	21.5	26.5	18	25
160	580	348	232	1.5	1.956	15	17.5	14	16.5
161	519	302	217	1.39	1.809	15	17	13	16
162	460	259	201	1.29	1.648	15	18	14	19
163	583	375	208	1.79	2.227	8	11	7.5	10.5
164	582	375	207	1.8	2.191	7	11	6.5	10
165	582	376	206	1.82	2.195	7.5	11.5	6	11
166	583	377	206	1.82	2.148	7	10	6	10.5
167	583	377	206	1.82	2.209	6	10	6	9.5
168	582	379	203	1.86	2.182	7	11	6.2	10.5
169	517	326	191	1.71	1.93	7.5	11.5	6	11
170	462	285	177	1.6	1.951	9.5	12.5	7	12
171	579	377	202	1.86	2.234	6	9.8	5	9.5
172	519	331	188	1.77	2.174	6.5	11	5	10
173	463	289	174	1.66	1.092	6	11	4	11
174	577	304	273	1.11	1.21	22.5	29.9	20.1	29.5
175	580	320	260	1.23	1.391	27	32.5	26	32.5
176	581	206	375	1.41	1.594	35	44.5	33	45
177	367	171	196	0.87	1.195	21.5	28.5	17	27
178	367	194	173	1.12	1.275	27.2	33.5	23.5	32.5
179	370	219	151	1.45	1.688	37.5	46	34.5	45.5
180	267	114	153	0.74	0.871	19	26.5	16	26
181	270	137	133	1.04	1.27	26.2	34.5	23	33.5
182	269	163	106	1.52	1.738	40	49.5	38	48
183	578	354	224	1.58	1.992	14	17	14	16
184	583	374	209	1.79	2.135	16	19	16	18
185	580	362	218	1.66	1.819	22	30.5	26.6	30
186	269	129	140	0.92	1.156	12.5	19	8.5	18.5
187	266	151	115	1.31	1.612	19	23	17.5	22
188	269	167	102	1.63	1.833	26	30	25	29
189	582	378	204	1.85	1.898	11	13	10	13
190	579	405	174	2.33	2.926	13	14.5	13	14

191	578	424	154	2.75	3.238	18	19.5	18	18.5
192	369	206	163	1.27	1.834	12.5	16	1	14
193	368	228	140	1.64	2.301	16	18	14	15.5
194	369	263	106	2.51	2.83	19	21.5	20	22
195	269	140	129	1.08	1.191	9	17	10	17
196	267	163	104	1.57	1.803	15.5	18	20	20
197	268	185	83	2.24	2.637	22	24.2	20.5	23
198	581	386	195	1.98	2.352	9	11	6	10
199	580	420	160	2.62	2.958	9	10.8	7	10.4
200	578	477	101	4.72	5.372	12	13	12	12.5
210	269	140	129	1.08	1.022	6	15.5	7	16
202	270	170	100	1.7	2.019	14	16.8	12	16.2
203	269	203	66	3.1	3.756	17.5	19.8	17	19