

---

Alpana Bhatt

**Reservoir properties from well logs  
using neural networks**

---

A dissertation for the partial fulfilment  
of requirements for the degree of  
Doktor Ingeniør at the

Department of Petroleum Engineering and  
Applied Geophysics  
Norwegian University of Science and Technology

November 2002



# Summary

In this dissertation we have developed multiple networks systems (multi nets or committee machines, CM) for predicting reservoir properties e.g. porosity, permeability, partial fluid saturation and for identifying lithofacies from wireline and measurement while drilling (MWD) logs. The method is much more robust and accurate than a single network and the multiple linear regression method. The basic unit of a committee machine is a multilayer perceptron network (MLP) whose optimum architecture and size of training dataset has been discovered by using synthetic data for each application. The committee machine generated for each property has been successfully applied on real data for predicting the reservoir properties and analysing the lithofacies in the Oseberg field, one of the major fields of the North Sea. The advantage of this technique is that it can be used in real time and thus can facilitate in making crucial decisions on the reservoir while drilling. The trained networks have been successfully used in bulk conversion of wireline and MWD logs to reservoir properties. All the programming has been done using MATLAB programming language and different functions from the neural network toolbox.

For porosity prediction we have made a study initially with a single neural network and then by the CM approach. We have demonstrated the benefits of committee neural networks where predictions are redundantly combined. Optimal design of the neural network modules and the size of the training set have been determined by numerical experiments with synthetic data. With three inputs i.e. sonic, density and resistivity, the optimal number of hidden neurons for the porosity neural network has been determined to be in the range 6-10, with a sufficient number of training patterns of about 150. The network is sensitive to the fluid properties. The unconstrained optimal linear combination of Hashem (1997), with zero intercept term based on least squares, is the most suitable ensemble approach for the porosity CM and the accuracy is mainly limited by the accuracy of the training patterns and the accuracy of the log data themselves. In application to real data the maximum standard error of the difference between prediction and helium core porosity data is 0.04. The benefit of neural networks compared with the multiple linear regression (MLR) technique has been briefly touched upon by showing that MLR fails to reproduce the minor non-linearity

embedded in the common log-to-porosity transforms, whereas the neural network reproduces the same data with high accuracy.

In permeability prediction by CM we have demonstrated the benefits of modularity by decomposing the permeability range into a number of sub-ranges to increase resolution. We have used synthetic data for evaluating the optimal architecture of the component neural networks. With the four inputs; i.e. sonic, density, gamma and neutron porosity, we find that optimal number of hidden units of the permeability neural network is confined to the range 8-12 where the variance and bias are at their minima. In general, the errors steadily decrease with the number of training facts. A practical lower limit has been set to 300, or twice the size of the training set required for the porosity network due to the increased complexity of the background relationships with the log readings.

Since we use a logarithmic permeability scale rather than a linear scale, the success of optimal linear combination (OLC) in the porosity CM is not repeated when it is applied to the permeability CM. In fact noise amplification takes place. Simple ensemble averaging is shown to be the preferred method of combining the outputs. A different training strategy must be applied i.e. the validation approach, requiring the training to stop when the level of minimum variance has been reached. Provided that precautions are taken, the permeability CM is more capable of handling the non-linearity and noise than MLR and a single neural network. The benefit of range splitting, using the modularity imbedded in the CM approach, has been demonstrated by resolving details in the combination of logs that otherwise would be invisible. In application to real data a minimum standard deviation error of the difference between prediction and Klinkenberg corrected air permeability data is around 0.3 in logarithmic units (of mD), mainly due to limitations in the techniques of measurement.

We have developed and tested a modular artificial neural network system for predicting the fluids water, oil and gas, and their partial saturations directly from well logs, without explicit knowledge of the fluid and rock properties normally required by conventional methods. For inputs to the networks we have used the density, sonic, resistivity and neutron porosity logs. From synthetic logs based on a realistic petrophysical model we have determined by numerical experiments the optimal architecture, and network training procedure for partial fluid saturation.

The output of three saturations from a single MLP (4-10-3) reveals the same accuracy as those of three individual MLPs with one output (4-4-1). The latter has the advantage of simplicity in terms of number of neurons, which implies fewer training patterns and faster training. Moreover, simplicity in the MLP improves modularity when used for building blocks in the multi-net system. For the optimal 4-4-1 MLP the number of training patterns should be in excess of 100 to ensure negligible errors in the case of data with moderate noise. A

committee neural network for each fluid type is the preferred solution, with each network consisting of a number of individually trained 4-4-1 MLPs connected in parallel and redundantly combined using optimal linear combination compared with a single MLP realisation. The OLC approach implies an overall error reduction by an order of magnitude.

Based on these findings we have made a modular neural network (MNN) system consisting of three CMs; one for each fluid type, where each CM contains nine MLPs connected in parallel, and with outputs that are combined using the OLC approach. Using training patterns from CPI logs we have demonstrated its application to real data from North Sea reservoirs containing the full range of fluid types and partial saturation. The saturation predictions from the fluid CMs are further combined in a MNN with the laboratory measured relative permeability curves for both the oil-water and gas-oil fluid systems to generate relative permeability logs.

The accuracy in prediction saturation essentially depends on the accuracy of the training patterns, which are from the computer processed interpretation (CPI) logs, and the accuracy of the individual log measurements. The idea of using neural networks for fluid saturation is thus not to eliminate the careful petrophysical evaluation behind the CPI log, but to transfer into the neural network for future application the effort and expertise already imbedded in the petrophysical database. Comparison of  $S_w$  values of the neural network with those of CPI logs, in wells that are unknown to the network, indicates a standard error of less than 0.03.

The problem of identification of lithofacies from well logs is a pattern recognition problem. The CM architecture is based on combining back propagation artificial neural networks (BPANN) with a recurrent BPANN (R-BPANN) adopted from time series analysis. The recurrent BPANN exploits the property of facies i.e. it consists of several sequential points along the well bore thus effectively removing ambiguous or spurious classification. The multiclass classification problem has been reduced to a two-class classification problem by using the modular neural network system. Ensembles of neural networks are trained on disjoint sets of patterns using a soft overtraining approach for ensuring diversity and improving the generalisation ability of the stack.

We have used synthetic logs from a realistic model with a very small layer contrast and moderate noise level and we found an excellent classification performance only slightly less than 100% hit rates. By introduction of fine layering in the model we have shown that the performance is only slightly reduced, demonstrating excellent performance of the RBPANN for layer enhancement, also in the case of thin layers. Classification from real data is more challenging since the facies in the present study were initially defined by visual inspection of cores, and thus not fully compatible with the readings of the logging

tools, which detect different physical properties and have coarser spatial sampling. Application to the four facies of the Ness Formation reveals an average hit-rate well above 90% in wells unknown to the network. Compared with similar published classification studies our results reveal slightly to significantly better performance.

The CM approach for porosity, permeability and water saturation is developed and tested on MWD data also. We trained CM architecture for porosity, permeability and water saturation using MWD data. Since cores are normally not collected in horizontal well the patterns for MWD networks are predictions made by wireline networks. The application of this technique is to predict reservoir properties while drilling.

# Acknowledgements

I sincerely wish to thank Dr. Hans B. Helle for his very valuable involvement and advice throughout this work. I am also very thankful to Prof. Bjorn Ursin for his confidence in me and for the support, guidance and crucial insights he provided during this research work.

I thank Norsk Hydro and Oseberg licence group for providing technical data, financial support and office facilities.

I immensely benefited from the technical discussions that I had with Brian Farrely and Dr. Jos Okkerman on various occasions. I thank them for helping me in improving my approach to many issues in this work.

I am grateful to Dr. Mary M. Poulton, Dr. Alv Aanestad and Professor Tor Arne Johansen for reviewing this work. I am thankful for their critical comments and very useful suggestions for improving the quality of this thesis.

I thank all employees and staff of IPT, NTNU for making my stay in the department very comfortable and satisfying. I especially want to thank my colleagues at IPT, Nam Hoai Pham and Klaus Lykke Dagø, for their academic feedback, help and pleasure of sharing some challenges together.

My gratitude to Mr. J.S.Thakur and his noble friend without whose benevolent intervention this work could not have been undertaken.

It is my pleasure to thank my parents for their unstinted support and understanding. I thank my son Padmanabh whose charm provided the much-needed comfort and invigorated my mind throughout this scientific pursuit. Last but not the least I am grateful to my husband for all his patience in encouraging me to do this study.

Trondheim, November 2002

Alpana Bhatt



## List of symbols

$a_{21}$ :	partial tortuosity for fluid flowing through the sand matrix
$a_{23}$ :	partial tortuosity for fluid flowing through the clay matrix
$b$ :	bias
$B$ :	geometrical factor
$C$ :	clay content
$\mathbf{d}$ :	desired output vector
$D$ :	grain size
$D_s$ :	effective particle size of sand
$D_c$ :	effective particle size of clay
$\bar{D}$ :	effective grain size
$\mathbf{e}$ :	error output vector
$e$ :	error between the desired and the actual output
$E$ :	expectation operator
$f$ :	volume fraction
$F$ :	target function
FFI:	free fluid index
$H_c$ :	hydrogen index of clay
$H_g$ :	hydrogen index of gas
$H_o$ :	hydrogen index of oil
$H_s$ :	hydrogen index of sand
$H_w$ :	hydrogen index of water
$h(z)$ :	output from BPANN at depth $z$ for lithofacies prediction
$\tilde{h}(z)$ :	output from RBPANN at depth $z$ for lithofacies prediction
$\bar{\tilde{h}}_k(z)$ :	output from CM for $k^{th}$ lithofacies prediction

$\bar{h}_k(z)$ : output from RBPANN for  $k^{th}$  lithofacies prediction  
 $k$ : permeability  
 $k_r$ : relative permeability  
 $L(z)$ : output from the combiner  
 $m$ : cementation factor  
 $M$ : particle mass  
 $n$ : saturation exponent  
 $r(\mathbf{x})$ : real response from a CM  
 $R_c$ : resistivity of clay  
 $R_t$ : true resistivity of a formation  
 $R_w$ : resistivity of formation water  
 $S_g$ : gas saturation  
 $S_o$ : oil saturation  
 $S_w$ : water saturation  
 $S_{wi}$ : irreducible water saturation  
 $t$ : time  
 $T$ : tortuosity  
 $u$ : input to adder  
 $\mathbf{w}$ : weight vector  
 $w_{kj}^i$ : weights on the connection between  $j^{th}$  input unit and  $k^{th}$  output unit corresponding to the  $i^{th}$  layer.  
 $\mathbf{W}$ : weight matrix  
 $\mathbf{x}$ : input vector  
 $\mathbf{y}$ : output vector  
 $\tilde{y}$ : combined output from  $K$  experts in a CM  
 $z$ : depth

$\alpha$ :	weights from OLC approach on each expert in CM
$\delta$ :	approximation error between the desired response and the output from CM
$\varepsilon$ :	cost function
$\varphi(\cdot)$ :	activation function
$\phi$ :	porosity
$\phi_c$ :	porosity in clay
$\phi_N$ :	neutron porosity log
$\phi_p$ :	percolation porosity
$\phi_s$ :	porosity in sand
$\gamma$ :	gamma ray log
$\gamma_{\min}$ :	minimum value of gamma ray log
$\gamma_{\max}$ :	maximum value of gamma ray log
$\eta$ :	regression parameters
$v$ :	intermediate output
$\rho$ :	bulk density
$\rho_f$ :	effective fluid density
$\rho_o$ :	density of oil
$\rho_w$ :	density of water
$\rho_g$ :	density of gas
$\rho_m$ :	matrix density
$\sigma$ :	standard deviation
$\Delta$ :	difference
$\Delta t$ :	bulk transit time
$\Delta t_c$ :	transit time of clay
$\Delta t_f$ :	effective fluid transit time
$\Delta t_g$ :	transit time of gas

$\Delta t_m$  : matrix transit time

$\Delta t_o$  : transit time of oil

$\Delta t_w$  : transit time of water

## List of acronyms

BPANN:	back-propagation artificial neural network
CM:	committee machine
CPI :	computer processed interpretation
LMBP:	Levenberg Marquardt back-propagation
LMS:	least mean square
mD :	milli Darcy
MLFF:	multi layer feed forward
MLP:	multi layer perceptron
MLR:	multiple linear regression
MNLR:	multiple nonlinear regression
MNN:	modular neural network
MWD:	measurement while drilling
OLC:	optimal linear combination
RBPANN:	recurrent back-propagation artificial neural network



# Contents

<b>Summary</b> .....	<b>iii</b>
<b>Acknowledgements</b> .....	<b>vii</b>
<b>List of symbols</b> .....	<b>ix</b>
<b>List of acronyms</b> .....	<b>xiii</b>
<b>1. Introduction</b> .....	<b>1</b>
<b>2. Neural networks</b> .....	<b>7</b>
2.1 Introduction.....	7
2.2 Neuron model.....	8
2.3 Network architectures .....	11
2.3.1 Learning .....	11
2.3.2 Perceptron architecture .....	12
2.3.3 Adaline architecture.....	13
2.3.4 Multilayer perceptron .....	14
2.4 Network tasks .....	17
2.4.1 Pattern recognition .....	17
2.4.2 Function approximation.....	18
2.5 Bias variance dilemma .....	19
2.6 Advantages and disadvantages of a MLP network .....	21
2.7 Multiple networks system .....	21
2.7.1 Ensemble combination .....	22
2.7.2 Modular neural network.....	25
2.8 Multiple linear regression .....	27
2.8.1 Analogy of MLR with neural network.....	28
<b>3. Porosity prediction</b> .....	<b>31</b>
3.1 Introduction.....	31
3.2 Synthetic data .....	31
3.3 Optimal design of networks.....	34
3.4 Training strategy .....	38
3.5 Comparison of alternative techniques .....	39
3.6 Real data .....	42
3.7 Conclusions .....	55
<b>4. Permeability prediction</b> .....	<b>57</b>
4.1 Introduction.....	57
4.2 Errors in data.....	60
4.2.1 Measurement conditions.....	60
4.2.2 Resolution and spatial sampling .....	63

4.2.3 Anisotropy .....	63
4.3 Synthetic data .....	64
4.4 Optimal design of networks .....	68
4.5 Error analysis of the committee networks .....	73
4.6 Real data.....	75
4.7 Conclusions.....	85
<b>5. Fluid saturation prediction.....</b>	<b>87</b>
5.1 Introduction .....	87
5.2 Water saturation measurement on cores .....	88
5.2.1 Retort method .....	88
5.2.2 Dean- Stark extraction method.....	88
5.3 Log responses to pore fluids.....	89
5.4 Synthetic data .....	92
5.5 Optimal design of networks .....	93
5.6 CM architecture .....	100
5.7 Real data.....	102
5.8 Conclusions.....	109
<b>6. Lithofacies prediction.....</b>	<b>111</b>
6.1 Introduction.....	111
6.2 Multiclass classification using MNN and stacked generalisation .....	113
6.3 Recurrent networks for enhanced layer detection.....	115
6.4 Optimal design and training of networks for classification ...	118
6.4.1 Back propagation neural net for classification.....	122
6.4.2 Recurrent back propagation networks for classification ..	123
6.4.3 Ensemble network and stacked generalisation.....	125
6.5 Classification of logfacies in a multi-layer model with thin layers .....	127
6.6 Real data.....	130
6.7 Conclusions.....	136
<b>7. Applications using MWD logs.....</b>	<b>137</b>
7.1 Introduction .....	137
7.2 Sources of error in the data .....	138
7.3 Porosity prediction .....	139
7.4 Permeability prediction.....	141
7.5 Water saturation prediction .....	143
7.6 Conclusions.....	145
<b>8. Conclusions and future work .....</b>	<b>147</b>
<b>References .....</b>	<b>151</b>

# Chapter 1

## Introduction

Many forms of heterogeneity in rock properties are present in clastic petroleum reservoirs. Understanding the form and spatial distribution of these heterogeneities is important in petroleum reservoir evaluation. Porosity, permeability and fluid saturation are the key variables for characterising a reservoir in order to estimate the volume of hydrocarbons and their flow patterns to optimise production of a field. These are different for different rocks. Lithofacies are defined as a lateral mappable subdivision of a designated stratigraphic unit, distinguished from adjacent subdivisions on the basis of lithology, including all mineralogical and petrographical characters and those paleontological characters that influence the appearance, composition or texture of the rock. Thus identification of lithofacies is an important task in knowing the heterogeneity of the reservoir.

Porosity is described as the ratio of the aggregate volume of interstices in a rock to its total volume whereas the permeability is defined as the capacity of a rock or sediment for fluid transmission, and is a measure of the relative ease of fluid flow under pressure gradients. Knowledge of permeability in formation rocks is crucial to oil production rate estimation and to reservoir flow simulations for enhanced oil recovery. Reliable predictions of porosity and permeability are also crucial for evaluating hydrocarbon accumulations in a basin-scale fluid migration analysis and to map potential pressure seals to reduce drilling hazards.

There is no well log which can directly determine porosity and hence it is measured on cores in the laboratory. This is an expensive exercise and hence is not a routine operation in all drilled wells. Several relationships have been offered which can relate porosity to wireline readings such as the sonic transit time and density logs. However, the conversion from density and transit time to equivalent porosity values is not trivial (Helle et al., 2001). The common conversion formulas contain terms and factors that depend on the individual location and lithology e.g. clay content, pore fluid type, grain density and grain transit time for the conversion from density and sonic log, respectively, that in general are unknowns and thus remain to be determined from rock sample analysis.

Permeability is also recognised as a complex function of several interrelated factors such as lithology, pore fluid composition and porosity. It can be measured directly from pressure test in wells and on cores in the laboratory. Although these are important methods neither of them is good enough to allow them to be widely used owing to technical and financial reasons. The routine procedure in the oil industry has been to estimate it from well logs. The permeability estimates from well logs often rely upon porosity e.g. through the Kozeny-Carman equation or Wyllie and Rose model (Wyllie and Rose, 1950) which contains adjustable factors such as the Kozeny constant that varies within the range 5-100 depending on the reservoir rock and grain geometry (Rose and Bruce, 1949). Nelson (1994) gives a detailed review of these problems.

Helle et al. (2001) demonstrated that instead of a single log a group of logs should be used to compute a petrophysical property provided there is a relationship between the property and the logs. Similar opinions have been emphasised in Wendt et al. (1986). They reported a study of permeability prediction by using multiple linear regression techniques. The study demonstrated that the correlation coefficient between the predicted and the actual permeability increases as other logs and log-derived parameters than porosity are included in the prediction for permeability.

Finding the distribution and composition of subsurface fluids is another main objective in hydrocarbon exploration, field development and production. Since direct sampling of underground fluids and determination of fluid saturation in the laboratory is an expensive and time-consuming procedure, indirect determination from log measurements is the common approach. The common practice in the industry is to determine water saturation from empirical formulas using resistivity, gamma ray logs and porosity estimates. The hydrocarbon saturation is then calculated from water saturation as both water and hydrocarbon form the composite pore fluid. Apart from the convenience and financial benefits, methods based on log measurements imply the technical advantage of providing a continuous record and sampling of a larger rock volume. Fluid evaluation from log data and accurate conversion of the logs to fluid saturation values thus also constitute important tasks in reservoir characterisation.

The standard practice in industry for calculating water saturation is by using different saturation models. But these models should be tuned to the area of work which requires the estimation of parameters in the laboratory. Thus it would be better to obtain water saturation from logs using neural networks without explicitly depending on the auxiliary parameters.

Lithofacies defines a body of rock on the basis of its distinctive lithological properties, including composition, grain texture, bedding characteristics, sedimentary structures and biological features. The common practice in oil industry is to manually examine the various facies identified on

cores from well logs with the aid of graphical techniques such as cross plotting. This method is labour intensive and becomes cumbersome when the number of logs to be analysed simultaneously increases. Thus there are several advantages in making this method computerised while retaining the expert reasoning of an experienced geologist.

Neural networks have been applied in a wide variety of fields to solve problems such as classification, feature extraction, diagnosis, function approximation and optimisation (e.g. Lawrence, 1994; Haykin, 1999). Although it seems clear that neural networks should not be used where an effective conventional solution exists, there are many tasks for which neural computing can offer a unique solution, in particular those where the data is noisy, where explicit knowledge of the task is not available or when unknown non-linearity between input and output may exist. Artificial neural networks are most likely to be superior to other methods under the following conditions (Masters, 1993):

- i) The data on which a conclusion is based is “fuzzy”, or is subject to possibly large errors. In this case the robust behaviour of neural networks is important.
- ii) The patterns important to the required decision are subtle or deeply hidden. One of the principal advantages of a neural network is its ability to discover patterns in data, which are so obscure as to be imperceptible to the human brain or standard statistical methods.
- iii) The data exhibits significant unpredictable non-linearity. Neural nets are marvellously adaptable to any non-linearity.
- iv) The data is chaotic (in the mathematical sense). Such behaviour is devastating to most other techniques, but neural networks are generally robust with input of this type.

In many respects the above list summarises the features of conventional earth science data, and are the main reasons for the increasing popularity of ANN in geoscience and petroleum engineering (Mohaghegh, 2000; Nikravesht et al., 2001).

Neural networks for quantitative analysis of reservoir properties from well logs have been demonstrated in several practical applications (e.g. Huang et al., 1996; Huang and Williamson, 1997; Zhang et al., 2000; Helle et al., 2001), where the artificial neural network approach is shown to be a simple and accurate alternative for converting well logs to common reservoir properties such as porosity and permeability. Single multilayer perceptrons (MLP) consisting of an input layer, a hidden layer and an output layer, trained by a back-propagation algorithm (e.g. Levenberg-Marquardt, see Hagan et al., 1996) have been the

conventional work horse for most practical applications over the last decade. Back-propagation learning techniques have been shown to be universal approximators (e.g. White, 1992), implying that they will approximate any static function provided sufficiently representative input-output sample pairs of the function are given. However, the concomitant disadvantage of their ability to generalise beyond the set of examples on which they were trained, is that they are likely to make errors.

If we accept that neural networks are unlikely to generalise perfectly to all possible test cases, we have good reason for exploring ways of improving the performance of neural networks. A single MLP, when repeatedly trained on the same patterns, will reach different minima of the objective function each time and hence give a different set of neuron weights. A common approach therefore is to train many networks, and then select the one that yields the best generalisation performance. However, since the solution is not unique for noisy data, as in most geophysical inversion problems, a single network may not be capable of fully representing the problem at hand. Selecting the single best network is likely to result in loss of information since, while one network reproduces the main patterns, the others may provide the details lost by the first. The aim should thus be to exploit, rather than lose, the information contained in a set of imperfect generalisers. This is the underlying motivation for the committee neural network approach, or committee machine, where a number of individually trained networks are combined, in one way or another, to improve accuracy and increase robustness. An important observation (Naftaly et al., 1997) is that a committee can reduce the variance of the prediction while keeping the bias constant, whereas Hashem, (1997) proposed unconstrained optimal linear combination to eliminate the bias.

Helle et al. (2001) demonstrated the prediction of porosity and permeability using single neural network. While using the committee machine approach Bhatt and Helle (2001a) have demonstrated improved porosity and permeability predictions from well logs using ensemble combination of neural networks rather than selecting the single best by trial and error. Helle and Bhatt (2001) successfully applied the ensemble approach to predict partial fluid saturation. Bhatt and Helle (2001b) successfully applied a committee machine using a combination of back propagation neural network and recurrent neural network for the identification of lithofacies.

In this dissertation we have devised a technique using neural networks for predicting porosity, permeability, fluid saturation and identifying lithofacies from log data. The technique utilises the prevailing unknown relationship in data between well logs and the reservoir properties. It utilises the ability of neural network to discover patterns in the data important for the required decision, which may be imperceptible to human brain or standard statistical methods. The method is better than the commonly practised technique in industry because it

does not require a deep geological knowledge of the area and is much faster to use than the standard statistical methods. It is more robust and accurate than the standard multiple linear regression technique. This technique makes the identification of lithofacies much simpler in comparison to manual identification specially when the number of logs to be analysed increases. Thus the idea of this dissertation is not to eliminate the interpretation from an experienced petrophysicist but to make the task simpler and faster for future work.

Thus in this study we have explored the capabilities of back propagation neural networks on synthetic and real data. The study shows the applications of neural networks in predicting reservoir properties with a short introduction to neural networks and the modular neural networks in Chapter 2. In all the applications synthetic data has been utilised for determining the optimal architecture of the network. In Chapter 3 we compare the prediction of porosity using a single neural network with that of the committee machine approach. The optimal architecture of the network was then applied to real data. We concluded that the optimal linear combination (OLC) approach (Hashem, 1997) is the best for porosity prediction. In Chapter 4 we discuss the different factors affecting the permeability and the reasons for the difference between the predicted and the core permeability. We demonstrate a comparison of the prediction of permeability using a single neural network with that of the committee machine approach. In Chapter 5 a committee machine has been devised to predict partial fluid saturation, which was further utilised for generating relative permeability logs. In Chapter 6 we devised a committee machine using back propagation and a recurrent back propagation neural network for predicting lithofacies within the Ness formation of the Oseberg field. The chapter discusses the method of identifying three lithofacies using synthetic data when the contrast between the lithofacies is only 1.25% of the contrast between sand and shale. BPANN leaves some overlapping in lithofacies, which are further, reduced by using RBPANN by utilising the past and the future predicted lithofacies and the past logs. The misclassification in predictions reduced from 6.74% to 2.89% by the application of RBPANN. The same technique is then applied to real data for identification of lithofacies within the Ness formation. In Chapter 7 we applied the committee machine architecture for predicting porosity, permeability and fluid saturation on measurement while drilling data. The individual networks for measurement while drilling were trained on patterns generated from wireline networks in the absence of core data. The main aim behind this approach is to determine the reservoir properties while drilling. Chapter 8 is the final conclusion of the thesis and contains suggestions for future work.



# Chapter 2

## Neural networks

### 2.1 Introduction

A neural network can be described as a massively parallel-distributed processor made up of simple processing units called neurons, which has a natural tendency for storing experiential knowledge and making it available for use. It resembles the human brain in the following respects: (i) knowledge is acquired by the network from its environment through a learning process (ii) interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

The human brain is a very complex, non-linear and parallel computer system, which is capable of thinking, remembering and problem solving. It has the capability to organise its structural constituents known as neurons so as to perform certain computations e.g. pattern recognition, perception etc. much faster than a computer. There have been many attempts to emulate brain functions with computer models and although there have been some spectacular achievements coming from these efforts, all of the models developed to date pale into oblivion when compared with the complex functioning of the human brain.

The fundamental cellular unit of the brain's nervous system is a neuron. It is a simple processing element that receives and combines signals from other neurons through input paths called dendrites. If the combined input signal is strong enough then the neuron fires, producing an output signal along the axon that connects to the dendrites of many other neurons. Figure 2.1 is a sketch of a biological neuron showing its various components. Each signal coming into a neuron along a dendrite passes through a synapse or a synaptic junction. This junction is an infinitesimal gap in the dendrite that is filled with neuro-transmitter fluid that either accelerates or retards the flow of electrical charges. The fundamental actions of the neuron are chemical in nature, and this fluid produces electrical signals that go to the nucleus or soma of the neuron. The adjustment of the impedance or conductance of the synaptic gap is a critically important process. These adjustments lead to memory and learning. As the synaptic strengths of the neurons are adjusted the brain learns and stores information.

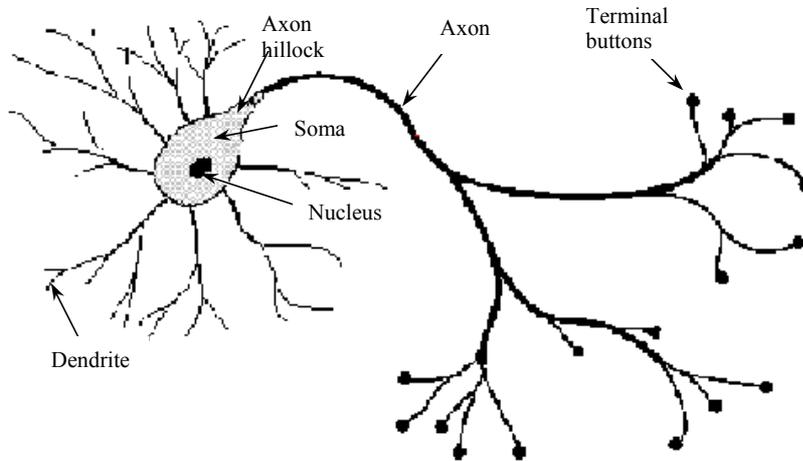


Figure 2.1: Schematic of a biological neuron

## 2.2 Neuron Model

A biological neuron is a fundamental unit of the brain's nervous system. Similarly, an artificial neuron is a fundamental unit to the operation of the neural network. The block diagram of Figure 2.2 shows the model of a neuron. Its basic elements are:

- i) A set of synapses or connecting links, each of which is characterised by a weight of its own. A signal  $x_j$  at the input of synapse  $j$  connected to neuron  $k$  is multiplied by the synaptic weight  $w_{kj}$ .
- ii) An adder for summing the input signals, weighted by the respective synapses of the neuron.
- iii) An activation function for limiting the amplitude of the output of a neuron. It limits the permissible amplitude range of the output signal to some finite value.

The neuronal model also includes an externally applied bias,  $b_k$  which has the effect of increasing or lowering the net input of the activation function, depending on whether it is positive or negative respectively.

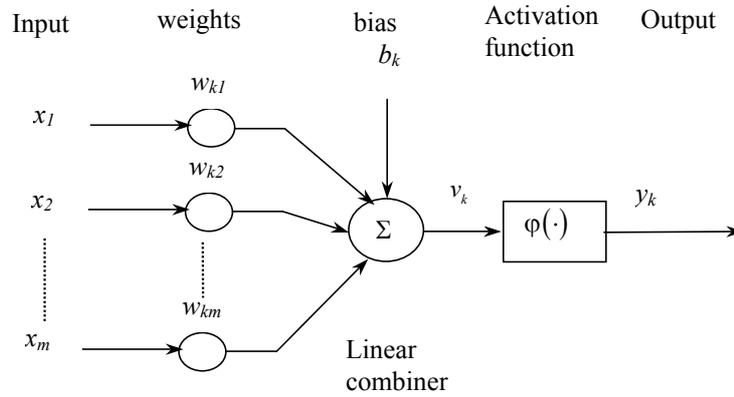


Figure 2.2: Model of a neuron

Mathematically the function of the neuron  $k$  can be expressed by

$$y_k = \varphi(u_k + b_k) \quad (2.1)$$

where

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (2.2)$$

$x_j$  is the input signal from an  $m$  dimensional input,  $w_{kj}$  is the synaptic weights of neuron  $k$ ,  $u_k$  is the linear combiner output due to the input signals,  $b_k$  is the bias,  $\varphi(\cdot)$  is the activation function, and  $y_k$  is the output signal of the neuron. The relation between the linear combiner output  $u_k$  and the activation potential  $v_k$  is

$$v_k = u_k + b_k \quad (2.3)$$

The activation function  $\varphi(v)$  defines the output of a neuron in terms of the induced local field  $v$ . The three most common activation functions are:

i) Hardlimit function

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \quad (2.4)$$

The output of neuron  $k$  employing such a function is expressed by

$$y_k = \begin{cases} 1 & \text{if } v_k \geq 0 \\ 0 & \text{if } v_k < 0 \end{cases} \quad (2.5)$$

A symmetrical hard limit function is described as

$$\varphi(v) = \begin{cases} +1 & \text{if } v \geq 0 \\ -1 & \text{if } v < 0 \end{cases} \quad (2.6)$$

which is most commonly used in pattern recognition problems.

ii) Linear function

$$\varphi(v) = v \quad (2.7)$$

The output of a linear transfer function is equal to its input. This activation function is used in pattern recognition and in function approximation problems.

iii) Sigmoid function

This is the most common form of activation function used in the construction of multilayer networks that are trained using back-propagation algorithm because it has the feature of being non-decreasing and differentiable and its range is  $0 \leq \varphi(v) \leq 1$ . An example of the sigmoid function is

$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad (2.8)$$

where  $a$  is the slope parameter of the sigmoid function. The slope at the origin equals  $a/4$ . In the limit as the slope parameter approaches infinity the sigmoid function becomes a threshold function. A sigmoid function has a continuous range of values from 0 to 1 (Figure 2.3). Sometimes it is desirable to have the activation function range from -1 to +1 in which case the activation function assumes an antisymmetric form with respect to the origin. Then  $\varphi(v)$  can be given by hyperbolic tangent function defined by

$$\varphi(v) = \tanh(v) \quad (2.9)$$

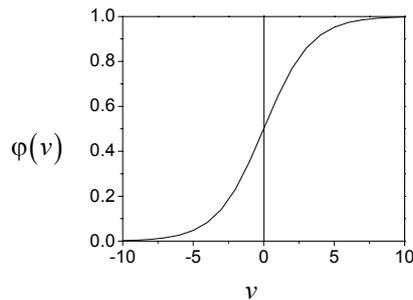


Figure 2.3: Sigmoid function

## 2.3 Network architectures

Warren McCulloch and Walter Pitts (McCulloch and Pitts, 1943) introduced one of the first models of an artificial neuron. The main feature of this model is that the weighted sum of input signals is compared with a threshold to determine the neuron output. When the sum is greater than or equal to the threshold, the output is 1. When the sum is less than the threshold the output is 0. They showed that this architecture could in principle compute any arithmetic or logical function. But as there was no training algorithm available the parameters of their networks had to be designed manually, unlike biological networks.

In late 1950s Frank Rosenblatt (Rosenblatt, 1958) and several other researchers developed a class of neural networks called perceptrons. The neurons in these networks were similar to those of McCulloch and Pitts but the key contribution of Rosenblatt was the introduction of a learning rule for training perceptron networks to solve pattern recognition problems. He proved that his learning rule will always converge to the correct network weights, provided such weights exist. The learning was simple and automatic. The perceptron was even able to learn when initialised with random values for its weights and biases.

### 2.3.1 Learning

A learning rule or training algorithm, is the procedure for modifying the weights and biases of a network. There are many types of learning algorithms that can be arranged into three main classes:

(i) Supervised learning: The learning rule is provided with a set of examples of proper network behaviour with inputs and outputs. As the inputs are applied to the network, the network outputs are compared to the targets. The learning rule

is then used to adjust the weights and biases of the network in order to move the network outputs closer to the targets.

(ii) Reinforcement learning: This is similar to supervised learning except that instead of being provided with a correct output for each network input the algorithm is only given a grade. The grade is a measure of the network performance over some sequence of inputs.

(iii) Unsupervised learning: In this type of learning the weights and biases are adjusted in response to network inputs only. There are no target outputs available. Most of these algorithms perform some kind of clustering operation. They learn to categorise the input patterns into a finite number of classes.

### 2.3.2 Perceptron architecture

A single neuron perceptron architecture is very similar to that shown in Figure 2.2. It has a hardlimit transfer function. The output of this network is given by

$$y = \text{hardlim}(\mathbf{w}^T \mathbf{x} + b) \quad (2.10)$$

where  $\mathbf{w}$  is the weight vector and  $\mathbf{x}$  is the input vector. For each neuron there will be one decision boundary for classifying the inputs given by

$$\mathbf{w}^T \mathbf{x} + b = 0 \quad (2.11)$$

A single perceptron can classify the input vectors into two categories only since its output can be either 0 or 1. This is the limitation of the perceptron architecture. A multiple perceptron architecture can classify inputs into many categories. The network weights are adjusted by supervised learning algorithm. Let  $D$  be a training set consisting of  $n$  inputs and outputs such that if  $d$  is the desired output and  $y$  is the computed output then the error  $e$  for  $k^{\text{th}}$  iteration is given by

$$e(k) = d(k) - y(k) \quad (2.12)$$

and the adjustments to weights and bias is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + e(k) \mathbf{x}(k) \quad (2.13)$$

$$b(k+1) = b(k) + e(k) \quad (2.14)$$

The perceptron learning rule is simple and powerful. It always converges to weights that accomplish the desired classification. However the limitation of a single perceptron network is that it can only classify the input vectors that can be separated by a linear boundary since its output can be either 0 or 1.

### 2.3.3 ADALINE architecture

Bernard Widrow and Hoff (1960) introduced the ADALINE (ADaptive LInear NEuron) network which is very similar to perceptron. The transfer function of ADALINE is linear, unlike the perceptron (Figure 2.2) and the output of the network is given by

$$y = \text{purelin}(\mathbf{w}^T \mathbf{x} + b) \quad (2.15)$$

where the decision boundary is same as given by equation (2.11). The learning algorithm for ADALINE is known as the Delta rule or least mean square (LMS) algorithm. It is also an example of supervised training. The algorithm will adjust the weights and biases of the ADALINE architecture by minimising the mean square error between the targeted output and the computed output, defined by

$$E[e^2] = E[(d - y)^2] = E[(d - \mathbf{w}^T \mathbf{x})^2] \quad (2.16)$$

The LMS algorithm finds the minimum point of the mean square error using the steepest descent algorithm (Hagan et al.1996). The adjustments to weights and bias using LMS algorithm for  $k^{\text{th}}$  iteration is given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\eta e(k) \mathbf{x}(k) \quad (2.17)$$

$$b(k+1) = b(k) + 2\eta e(k) \quad (2.18)$$

where  $\eta$  is a constant known as learning factor.

Both the ADALINE and the perceptron have the same limitation that they could classify only linearly separable problems. The LMS algorithm is however more powerful than the perceptron learning rule. While the perceptron rule is guaranteed to converge to a solution that correctly categorises the training patterns, the resulting network can be sensitive to noise since patterns often lie

close to the decision boundaries. The LMS algorithm minimises the mean square error and therefore tries to move the decision boundaries as far from the training

patterns as possible. The LMS algorithm has many more applications than the perceptron learning rule, particularly within the area of digital signal processing.

The LMS algorithm is optimal for a single neuron because the mean squared error surface for a single neuron has only one minimum point and constant curvature, providing a unique solution. For a multilayer perceptron network, on the other hand, the error surface may have many local minima and the curvature can vary widely in different regions of the parameter space.

Thus the LMS algorithm fails to produce a unique solution in multilayer networks. It does not specify how to adjust the weights for interior layer units during learning. In multilayer networks this is known as the credit assignment problem. There is no clear way in which to assign credit or blame to the internal layer unit weights for the reduction of output unit errors. The credit assignment problem was solved using the error back-propagation (BP) method (Hagan et al., 1996) which is a generalisation of the LMS algorithm.

### 2.3.4 Multilayer perceptron

The multilayer perceptron (MLP) networks are currently the most widely used neural networks. These networks can do the classification for patterns having nonlinearly separable boundaries since the network consists of many neurons. It consists of an input layer one or more internal layers of hidden neurons and an output layer. They are also called multilayer feed forward networks (MLFF). A 3-layer MLP network is shown in Figure 2.4. The hidden layers are also called internal layers as they receive internal inputs i.e. inputs from other processing units and produce internal outputs i.e. outputs to other processing units. Consequently they are hidden from the outside world. The network is provided with a training set of patterns having inputs and outputs. Real valued  $m$ -dimensional input feature vectors  $\mathbf{x}$  are presented to each of the first hidden layer units through weight vector  $\mathbf{w}$ . Hidden layer unit  $k$  receives input  $j$  through the synaptic weight,  $w_{kj}$ ,  $k = 1, 2, \dots, h$ , and  $j = 1, 2, \dots, m$ . Unit  $k$  computes a function of the input signal  $\mathbf{x}$  and the weights  $w_{kj}$  and passes its output forward to all of the units in the next successive layer. Like the first hidden layer, the units of the second hidden layer are fully connected to the previous layer through the synaptic weights. These units also compute a function of their inputs and their synaptic weights and pass their output on to the next layer. The output of one layer becomes the input to the following layer. Then at the output unit error is calculated between the target value and the computed value of the pattern.

This process is repeated until the final computation is produced by the output unit. The learning algorithm for this type of network is called the back-propagation (BP) algorithm which was published in the mid 1980s for multilayer

perceptrons. This architecture of the network is the basic unit in this study. Hornik et al. (1989) suggested that if a sufficient number of hidden units are available then an MLP with one hidden layer having a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer can approximate any function to any degree of accuracy.

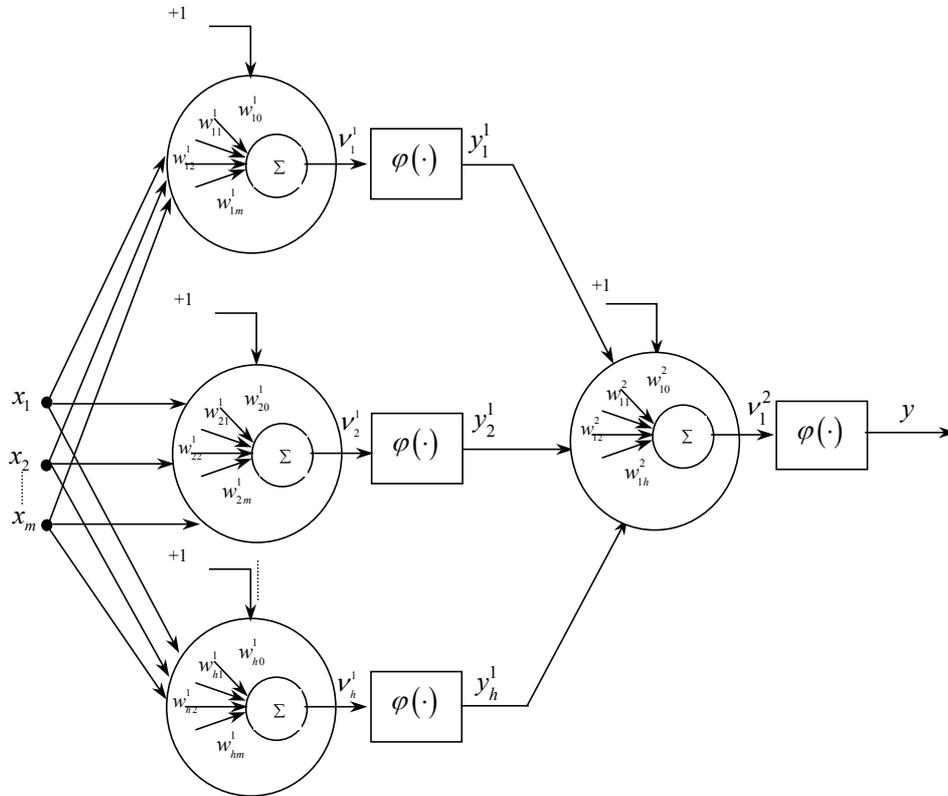


Figure 2.4: A multi layer perceptron network (superscript denotes the number of layer)

The BP algorithm is the generalisation of the LMS algorithm using the mean square error as the cost function. The mean square error is minimised iteratively using the steepest descent algorithm. The difference between the BP algorithm and the LMS algorithm lies in the way in which the gradient of the error is calculated. The BP algorithm uses the chain rule in order to compute the derivatives of the squared error with respect to the weights and biases in the hidden layers. It is called back-propagation because the derivatives of the squared error are computed first at the last layer of the network and are then propagated backwards through the network using the chain rule to compute the

derivatives in the hidden layer. In BP learning the backward linkages are used only for the learning phase, whereas the forward connections are used for both the learning and the operational phases. The process is repeated a number of times for each pattern in the training set until the total output squared error converges to a minimum or until some limit is reached in the number of training iterations.

One of the major problems with BP algorithm has been the long training times due to the steepest descent method, as it is a simple but slow minimisation method. The learning rate is sensitive to the weight changes. The smaller is the learning rate the smaller will be the changes to the synaptic weights from one iteration to the next, and the smoother will be the trajectory in the weight space. On the other hand, if the learning rate is chosen too large in order to speed up the rate of learning, the resulting large changes in the synaptic weights make the network unstable. In order to speed up the convergence of BP algorithm, along with improved stability, a momentum term is added to the weight update of the BP algorithm. A momentum term is simple to implement and this significantly increases the speed of convergence. The inclusion of momentum term represents a minor modification to the weight update. The inclusion of momentum may also have the benefit of preventing the learning process from terminating in shallow local minima on the error surface.

The second method of accelerating BP algorithm is by using Levenberg-Marquardt BP (LMBP) algorithm (Hagan et al., 1996). It is based on Newton's optimisation method (Hagan et al., 1996) and differs from the usual BP algorithm in the manner in which the resulting derivatives are used to update the weights. The main drawback of the algorithm is the need for large memory and storage space of the free parameters in the computers. If the network has more than a few thousand parameters the algorithm becomes impractical on current machines. In this study we have designed the network architecture such that the number of free parameters is smaller than the number of training patterns. We have used the modular architecture of the network i.e. subdividing the task of one network into several networks. The largest network size we have used in this study is for the permeability network consisting of 4 input neurons, 10 hidden neurons and 1 output neuron, implying 65 free parameters and therefore the LMBP algorithm was adequate for this study.

While training a MLP by the back-propagation algorithm we compute the synaptic weights of a MLP by loading as many of the training examples as possible into the network. The hope is that the neural network so designed will generalise. A network is said to generalise well when the input – output mapping computed by the network is correct for test data which is unknown to the network. A neural network that is designed to generalise well will produce a correct input-output mapping even when the input is slightly different from the examples used to train the network. When however a neural network has too

many hidden neurons, the network may end up memorising the training data. It may do so by finding a feature that is present in the training data (noise) but not true of the underlying function that is to be modelled. This phenomenon is referred to as overfitting. Overfitting is the result of more hidden neurons than is actually necessary, with the result that undesired contributions in the input space due to noise are stored in synaptic weights. However, if the number of hidden neurons is less than the optimum number than the network is unable to learn the correct input output mapping. Hence it is important to determine the optimum number of hidden neurons for a given problem.

Generalisation is influenced by three factors; i) the size of the training set ii) the architecture of the neural network and iii) the complexity of the problem at hand. We cannot control the latter. In the context of the first two for a good generalisation to occur we may vary the size of the training set by keeping the architecture of the network fixed or vice versa. This problem can be resolved in terms of the Vapnik-Chervonenkis (VC) dimension which is a measure of the capacity or expressive power of the family of classification functions realised by a network. It can be defined as the maximum number of training examples for which a function can correctly classify all the patterns in a test dataset (Patterson, 1996). The bounds specified by the VC dimension can be simply stated as following; if one wants an accuracy level of at least 90% then one should use ten times as many training examples as there are weights in the network. The same has been suggested by Poulton (2001). Baum and Haussler (1989) also stated the same that if we want to have an accuracy of  $(1-e)$  the number of training samples should be  $\frac{W}{e}$ . In this study, keeping the above facts in mind we have used synthetic data to find the optimal number of training samples as well as the optimal number of hidden neurons.

## 2.4 Network tasks

Neural networks can be used to perform the two basic tasks of pattern recognition and function approximation.

### 2.4.1 Pattern recognition

Pattern recognition is formally defined as the process whereby a received pattern or signal is assigned to one of a prescribed number of classes. A neural network performs pattern recognition by first undergoing a training session, during which the network is repeatedly presented a set of input patterns along with the category to which each particular pattern belongs. Later a new pattern is presented to the network that has not been seen before but which belongs to the same population of patterns used to train the network. The network is able to identify the class of that particular pattern because of the information it has

extracted from the training data. The pattern recognition performed by a neural network is statistical in nature with patterns being represented by points in a multidimensional decision space. The decision space is divided into regions, each one of which is associated with a class. The decision boundaries are determined by the training process.

This task can be performed by using either an unsupervised network or a supervised network. For an unsupervised network the system is split into two parts; i) an unsupervised network for feature extraction and ii) a supervised network for classification. Such a method follows the traditional approach to statistical pattern recognition. Feature extraction is the process of transforming the input patterns into an intermediate smaller dimensional space thus simplifying the task of classification. Then classification is described as the process by which the intermediate patterns are mapped into one of the classes in an  $r$ -dimensional space where  $r$  is the number of classes to be distinguished.

For a supervised network the system is designed as a multilayer perceptron network using a supervised learning algorithm. The computational units in the hidden layer of the network perform the task of feature extraction and classification both based on the information it has extracted from the training data. In this study for classification of lithofacies we use a supervised network for feature extraction and classification and a recurrent back propagation network for removing the ambiguities due to the overlapping within the classes.

## 2.4.2 Function approximation

Neural networks have the capability of doing a function approximation between inputs and outputs to a desired degree of accuracy. Consider an input - output mapping described by the functional relationship

$$d = f(\mathbf{x}) \quad (2.19)$$

The function  $f(\mathbf{x})$  is assumed unknown. We are given a set of examples to map the function.

$$D = \{(\mathbf{x}_i, d_i)\}_{i=1}^m \quad (2.20)$$

We have to design a neural network that approximates the unknown  $f(\mathbf{x})$  such that the function  $F(\mathbf{x})$  describes the input - output mapping actually realised by the network close enough to  $f(\mathbf{x})$  for all inputs such that

$$|F(\mathbf{x}) - f(\mathbf{x})| < \varepsilon \quad \text{for all } \mathbf{x} \quad (2.21)$$

where  $\varepsilon$  is a small positive number. Provided that the size  $m$  of the training set is large enough and the network is equipped with an adequate number of free parameters then the approximation error  $\varepsilon$  can be made small enough. This function approximation problem can be best solved by using a neural network with supervised learning. This ability of neural network is exploited in approximating an unknown relationship between the inputs and outputs. In this study we have made a function approximation between the input logs and the reservoir properties such as porosity, permeability and water saturation.

## 2.5 Bias variance dilemma

A neural network can be trained to construct a function  $F(\mathbf{x})$ , based on a training set  $D$  as given by equation (2.20) for the purpose of approximating  $d$  for previously unseen observations of  $\mathbf{x}$ . As discussed in Geman et al.,(1992) the dependence of the predictor  $F$  on the training data can be given by  $F(\mathbf{x};D)$ . Then the mean squared error of  $F$  as a predictor of  $d$  may be written as

$$E_D \left[ \left( F(\mathbf{x};D) - E[d | \mathbf{x}] \right)^2 \right] \quad (2.22)$$

where  $E_D$  is the expectation operator with respect to the training set  $D$  and  $E[d|\mathbf{x}]$  is the target function. The error decomposition gives the bias and variance as

$$E_D \left[ \left( F(\mathbf{x};D) - E[d | \mathbf{x}] \right)^2 \right] = \quad (2.23)$$

$$\underbrace{(E_D [F(\mathbf{x};D)] - E[d | \mathbf{x}])^2}_{\text{bias term}} + \underbrace{E_D \left[ \left( F(\mathbf{x};D) - E_D [F(\mathbf{x};D)] \right)^2 \right]}_{\text{variance term}}$$

The bias and variance of a predictor can be estimated when the predictor is trained on different sets of data sampled randomly from the entire set. The bias of a network is intuitively characterised as a measure of its ability to generalise correctly to a test set. The variance of a network is characterised as a measure of the extent to which the output of a network is sensitive to the data on which it was trained. There is a trade-off between bias and variance in terms of training neural networks; the best generalization requires a compromise between the conflicting requirements of small variance and small bias. It is a trade-off because attempts to decrease the bias are likely to result in higher variance, whilst efforts to decrease the variance usually result in increased bias.

This leads to a discussion about two types of training procedures i) overtraining and ii) cross-validation approach. In overtraining the network is trained until the desired limit for the error or the number of training epochs is reached. The neural networks trained by this method when tested gives a smaller bias but a larger variance on the test set. In the cross-validation approach the training dataset is partitioned into two disjoint subsets, estimation subset and the validation subset. The estimation subset is used for training. The training session is stopped periodically and the network is tested on the validation subset and the validation error is measured. The network is allowed to train as long as the mean square error on the validation subset is decreasing. The training is stopped when the validation error starts to increase. Thus the testing results can be used to improve the training. After learning stops the weights are frozen and the network performance can be validated using new data. The purpose of the validation process is to determine if the network is capable of solving the problem it is trained for. This method is known as the early stopping method of training. Figure 2.4 shows the error variation with the two types of training methodologies. The same method has also been suggested by Poulton (2001) in order to know when to stop the training.

In this study we use the overtraining approach for predicting porosity and water saturation, cross validation approach for predicting permeability and a soft overtraining approach for lithofacies identification. These choices are based on tests using synthetic and real data for the problems at hand.

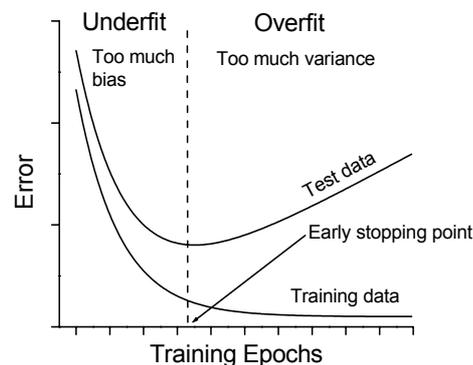


Figure 2.5: Error history during training and testing of a MLFF.

## 2.6 Advantages and disadvantages of a MLP network

An MLP network generates nonlinear relationship between inputs and outputs by the interconnection of nonlinear neurons. The nonlinearity is distributed throughout the network. It does not require any assumption about the underlying data distribution for designing the networks, hence the data statistics do not need to be estimated. Its parallel structure makes it realizable in parallel computers. The network exhibits a great degree of robustness or fault tolerance because of built-in redundancy. Damage to a few nodes or links thus need not impair overall performance significantly. It can form any unbounded decision region in the space spanned by the inputs. Such regions include convex polygons and unbounded convex regions. The network has a strong capability for function approximation. The abilities to learn and generalize are additional qualities. Previous knowledge of the relationship between input and output is not necessary unlike for statistical methods. The MLP has a built-in capability to adapt its synaptic weights to changes in the surrounding environment by adjusting the weights to minimise the error.

For an MLP network the topology is important for the solution of a given problem, i.e. the number of hidden neurons and the size of the training dataset. Because of the distributed nonlinearity and the high connectivity of the network a theoretical analysis of the network response is difficult to undertake. The use of hidden neurons makes the learning process harder to visualise. We identify the optimal architecture of the network by using synthetic data and the theory of VC dimension. Large-scale MLP networks have extremely low training rates when the back-propagation algorithm is used, since the networks are highly nonlinear in the weights and thresholds. We use Levenberg-Marquardt algorithm to overcome this problem. The mean square error surface of a multiple network can have many local minima and a global minimum and the network may get stuck in the local minima instead of converging into the global minimum. As a result the output of a single network may not be satisfactory. In order to avoid this problem we have in this study used multiple networks instead of a single network so that if one of the networks gives larger error it can be eliminated or the contribution of this network can be reduced in the ensemble output. This is called the test and select approach (Sharkey et al. 2000).

## 2.7 Multiple networks system

A multiple network system is a combination of several networks. There are several problems, which cannot be solved by a single network or can be more efficiently solved by multiple networks. Better performance can be achieved by

combining redundant networks instead of a single network. There are two common methods of creating a multiple network system, i) ensemble combination and ii) modular combination. In an ensemble the component networks are redundant in that they each provide a solution to the same task or task component even though this solution might be obtained by different means. By contrast, under a modular approach the task or problem is decomposed into a number of subtasks, and the complete task solution requires the contribution of all of the several modules. Both the ensemble and the modular combinations can exist at either a task or a sub-task level. The ensemble and the modular combinations are not mutually exclusive and an actual multiple network system could consist of a mixture of ensemble and modular combinations at different levels. The architecture of the networks for predicting porosity and water saturation are ensemble combination while the architecture of the networks for predicting permeability and lithofacies are modular and ensemble combination. Haykin (1999) denotes a multiple network system a committee machine (CM).

### 2.7.1 Ensemble combination

The main motivation for combining networks in redundant ensembles is that of improving their generalisation ability or to guard against the failure of individual component networks. The reason for expecting individual networks to sometimes fail or make errors on some inputs is based on the assumption that they have been trained on a limited set of training data, and on the basis of that training data the network is required to estimate the target function. Such estimates will inevitably not be identical to the target function unless the training data set is infinitely large. The mean square error of a predictor is expressed in terms of the bias and the variance as given in equation (2.23).

While generating the ensemble combination the expectation is that the differently trained experts converge to different local minima on the error surface, and the overall performance is improved by combining the outputs in some way. As demonstrated by the error analysis of Naftaly et al.(1997):

- i) The bias of the ensemble averaged-function  $F_1(\mathbf{x})$  pertaining to the committee machine is exactly the same as that of the function  $F(\mathbf{x})$  pertaining to a single neural network.
- ii) The variance of the ensemble-average function  $F_1(\mathbf{x})$  is less than that of the function  $F(\mathbf{x})$ . These theoretical findings point to a training strategy for reducing the overall error produced by the multiple network system due to varying initial conditions (i.e. random initial weights)
- iii) The individual expert should be purposely over-trained to reduce the bias at the cost of the variance.

There is clearly no advantage to be gained from an ensemble that is composed of a set of identical networks, which generalise in the same way. The emphasis here is on the similarity or otherwise of the pattern of generalisation. In principle, a set of networks could vary in terms of their weights, the time they took to converge and even their architecture and yet constitute essentially the same solution, since they result in the same pattern of errors when tested on a test set. The aim then is to find networks, which generalise differently. There are a number of training parameters, which can be manipulated with this goal in mind. These include the following: the initial weights, the training data, the topology of the networks and the training algorithm while keeping the other parameters constant. In this study we generated the ensemble for predicting porosity and water saturation by randomly varying the initial weights from which each network is trained while holding the training data constant. Varying the training data is another popular approach to generate networks for ensemble combination.

The training data can be varied by boosting as described by Schapire (1990) showing that a series of weak learners can be converted to one strong learner as a result of training the members of an ensemble on patterns that have been filtered by previously trained members of the ensemble. The problem with this method is that it requires large amounts of data. Freund and Schapire (1996) proposed the AdaBoost algorithm that largely eliminates this problem. The basis of this algorithm is that the training sets are adaptively resampled such that the weights in the resampling are increased for those cases, which are often misclassified.

Another method of varying the training data is by bootstrapping (Breiman, 1996). Multiple versions of training dataset can be created by picking  $n$  samples at a time from a training dataset of  $N$  samples by replacement. This procedure can be repeated several times to create a number of different, overlapping, data sets. If there is a scarcity of training patterns, as normally in case of real data, then virtual samples can be generated as suggested in Cho et al. (1997). A similar method is to use disjoint or mutually exclusive training sets; i.e. sampling without replacement (Sharkey et al., 1996). There is then no overlap between the data, which is used to train different networks and diversity is thus guaranteed. This method has been used in lithofacies prediction in the present study in generating different networks for a committee of one lithofacies. However, the committees for the different lithofacies have training datasets, which are overlapping.

Once a set of networks has been created, an effective way of combining the several outputs must be found. There are several different methods of combining the networks depending on the task performed by the networks i.e. if it is performing function approximation or pattern recognition. A single output can be created from a set of network outputs via ensemble averaging (Naftaly et

al.,1997) or by means of a weighted average that takes account of the relative accuracies of the networks to be combined (Hashem, 1997).

As suggested by Hashem (1997) the networks could be combined by providing weights to each network. Two approaches were suggested i) unconstrained and ii) constrained. In the unconstrained approach by assigning different weights to the output of the individual network the combined output from the  $K$  experts may be expressed by the weighted sum

$$\tilde{y}(\mathbf{x}, \boldsymbol{\alpha}) = \sum_{k=1}^K \alpha_k y_k(\mathbf{x}) \quad (2.24)$$

where  $y_k(\mathbf{x})$  is output from the individual network,  $\alpha_k, k=1, K$  are the weights and  $\mathbf{x} = \{x_i; i=1..N\}$  is the input vector. The requirement is to evaluate  $\alpha_k$  so that  $\tilde{y}$  provides a least-squares estimate of the desired response  $d(\mathbf{x})$ . Given a set of training data the approximation error is

$$\delta(\mathbf{x}; \boldsymbol{\alpha}) = d(\mathbf{x}) - \tilde{y}(\mathbf{x}; \boldsymbol{\alpha}). \quad (2.25)$$

In order to correct for any bias Hashem et al. (1994) extended the definition of  $\tilde{y}(\mathbf{x}; \boldsymbol{\alpha})$  to include a constant term  $\alpha_0$ . Thus, the modified  $\tilde{y}(\mathbf{x}; \boldsymbol{\alpha})$  is expressed by

$$\tilde{y}(\mathbf{x}, \boldsymbol{\alpha}) = \sum_{k=0}^K \alpha_k y_k(\mathbf{x}) = \boldsymbol{\alpha}^T \mathbf{y}(\mathbf{x}) \quad (2.26)$$

where  $\boldsymbol{\alpha}^T$  and  $\mathbf{y}(\mathbf{x})$  are  $(K+1) \times 1$  vectors and  $y_0(\mathbf{x})=1$ . As demonstrated by Hashem (1997) the unconstrained mean-squared error (MSE) optimal linear combination (OLC) of weights as given by (2.26) is equivalent to regression of  $d(\mathbf{x})$  against  $y_k(\mathbf{x}); k=1, K$  with an intercept term  $\alpha_0$ , i.e.

$$r(\mathbf{x}) = \alpha_0 + \sum_{k=1}^K \alpha_k y_k(\mathbf{x}) + \varepsilon \quad (2.27)$$

where  $\varepsilon$  is a random error with zero mean.

In the constrained approach there is an additional constraint, that is

$$\sum_{k=0}^K \alpha_k = 1. \quad (2.28)$$

The weights can be calculated from the training dataset, or partially on the training and test dataset. This method is called the optimal linear combination (OLC) method which provides an extra fit by means of different weights for each network.

Another approach could be combining the networks nonlinearly in which a nonlinear network learns how to combine the networks with weights that vary over the input space (Wolpert, 1992). While doing the pattern recognition, the networks can be combined by averaging. In the present study while predicting porosity and water saturation we have combined the redundant networks by the OLC method. Each individual network was trained on different initial weights.

## 2.7.2 Modular neural network

Modular decomposition can be undertaken for the purpose of improving performance. A task could be solved with a single monolithic network but better performance is achieved when it is broken down into a number of specialist modules. The divide and conquer approach (Jordan and Jacobs, 1994) that is exemplified by the mixture of experts approach (Haykin, 1999) provides an example of the improved performance that can result from a modular system. In addition to performance improvement it could be necessary to decompose the task into subtasks. Thus the divide and conquer principle can be used to extend the capabilities of a single network. If a single network replaces the combination of experts, this network would have a correspondingly large number of adjustable parameters and hence the risk of overfitting the training dataset increases. The training time for such a large network is likely to be longer than for all the experts trained in parallel. Each subproblem could then be solved with a different neural net architecture or algorithm making it possible to exploit specialist capabilities. Each component in a modular system can take the form of a neural network or a non-neural computing technique. The pre-processing of neural network inputs before training is also viewed as a form of modular decomposition for the purpose of simplifying the problem. A modular approach can also be recombining rather than decomposing as is the case when the input information comes from a number of independent sources or sensors and the potential for modularity is inherent in the task itself. Another reason for adopting a modular approach is that of reducing model complexity, and making the overall system easier to understand, modify and extend. Training times can be reduced as a result of modular decomposition and prior knowledge can be incorporated in terms of suggesting an appropriate decomposition of a task.

The most important factor of a modular system is the way modular components have been created. A task may be decomposed into modules or

alternatively the input to the system might come from a number of independent sources and the task is to combine them to form an overall solution. The decomposition of a problem into modular components may be accomplished automatically, explicitly or by means of class decomposition. Class decomposition involves breaking a problem into subproblems based on the class relationships. The method as proposed by Anand et al. (1995) involves dividing a  $k$ -class classification problem into a  $k$  two-class classification problems, whilst using the same number of training data for each two class classification as the original  $k$ -class problem. We also used the same principle in lithofacies prediction i.e. dividing a multiclass classification problem into a two class classification problem by maintaining an overlapping training dataset for different lithofacies. Each modular component is a neural network trained for a single lithofacies.

An alternative approach is one in which the automatic decomposition of the task is undertaken, characterized by a blind application of a data partitioning technique. Automatic decomposition is more likely to be carried out with a view to improving performance whilst explicit decomposition might either have the aim of improving performance or that of accomplishing a task which either could not be accomplished using a single network or could not be accomplished either as easily or as naturally. Under the divide and conquer approach complex problems are automatically decomposed into a set of simpler problems. Mixture of experts and hierarchical mixtures of experts partition the data into regions and fit the simple surfaces to the data that fall in each region. Expert networks learn to specialize on subtasks and to cooperate by means of a gating network. The regions have soft boundaries which means that data points may lie simultaneously in multiple regions. The mixture of experts model consists of a number of expert networks combined by means of a gating network, which identifies the expert or blend of experts most likely to approximate the desired response.

There are four different modes of combining component networks as viewed by Sharkey (1999); cooperative, competitive, sequential and supervisory. The main difference between cooperative combination and competitive combination is that in cooperative combination it is assumed that all of the elements to be combined will make some contribution to the decision even though this contribution may be weighted in some way; whereas in competitive combination it is assumed that for each input the most appropriate element (task, component or sensor) will be selected. In sequential combination the processing is successive; the computation of one module depending on the output of preceding module. In a supervisory relationship one module, is used to supervise the performance of another module. Ensembles are always involved in cooperative combination.

## 2.8 Multiple linear regression

Like neural networks, regression analysis also deals with the formulation of mathematical models that depict relationships among variables and the use of these modelled relationships for the purpose of prediction and other statistical inferences. In many cases the relationship between the variables is not deterministic but the variables are mutually related or one variable is dependent on a number of influencing or causal variables where the relation is not governed by a precise physical law. A plot of some observed values of the variables

depicts a relation among them. After gaining sufficient knowledge about an empirical relation it may be possible for the investigator to formulate a theory that leads to a mathematical formula and hence to the semi-deterministic case.

Suppose that the response variable  $y$  in an experiment is expected to be influenced by the three causal variables  $x_1, x_2$  and  $x_3$  of vector  $\mathbf{x}$  and that the data relevant to all these variables are recorded with the measurements of  $y$ . By analogy to linear regression model we can write

$$y = \eta_0 + \eta_1 x_{i1} + \eta_2 x_{i2} + \eta_3 x_{i3} + e_i, \quad i = 1, \dots, n \quad (2.29)$$

where  $x_{i1}, x_{i2}, \dots, x_{in}$  are the fixed values of the three independent variables in the  $i^{\text{th}}$  experimental trial and  $y_i$  is the corresponding response. The error components  $e_i$  are assumed to be independent normal variables with mean of zero and variance of  $\sigma^2$ . The parameters  $\eta_0, \eta_1, \eta_2$  and  $\eta_3$  are unknown fixed quantities and are called the regression parameters. The model suggests that the response varies linearly with each of the independent variables when the other two remains fixed. The regression parameters are estimated by the principle of least squares by minimising the sum of squared deviations given by

$$\sum_{i=1}^n (y_i - \eta_0 - \eta_1 x_{i1} - \eta_2 x_{i2} - \eta_3 x_{i3})^2 \quad (2.30)$$

The obtained least squares which are the estimates of  $\eta_0, \eta_1, \eta_2$  and  $\eta_3$  denoted by  $\hat{\eta}_0, \hat{\eta}_1, \hat{\eta}_2$  and  $\hat{\eta}_3$  and can be verified to be the solutions to the following normal equations:

$$\hat{\eta}_1 S_{x_1}^2 + \hat{\eta}_2 S_{x_1 x_2} + \hat{\eta}_3 S_{x_1 x_3} = S_{x_1 y} \quad (2.31)$$

$$\hat{\eta}_1 S_{x_1 x_2} + \hat{\eta}_2 S_{x_2}^2 + \hat{\eta}_3 S_{x_2 x_3} = S_{x_2 y} \quad (2.32)$$

$$\hat{\eta}_1 S_{x_1 x_3} + \hat{\eta}_2 S_{x_2 x_3} + \hat{\eta}_3 S_{x_3}^2 = S_{x_3 y} \quad (2.33)$$

$$\hat{\eta}_0 = \bar{y} - \hat{\eta}_1 \bar{x}_1 - \hat{\eta}_2 \bar{x}_2 - \hat{\eta}_3 \bar{x}_3 \quad (2.34)$$

where  $S_{x_1}^2, S_{x_1 x_2}$ , etc. are the sums of squares and cross products of the variables in the suffix and can be given by

$$S_{x_1}^2 = \sum (x_{i1} - \bar{x}_1)^2 = \sum x_{i1}^2 - n\bar{x}_1^2 \quad (2.35)$$

$$S_{x_1 x_2} = \sum (x_{i1} - \bar{x}_1)(x_{i2} - \bar{x}_2) = \sum x_{i1} x_{i2} - n\bar{x}_1 \bar{x}_2 \quad (2.36)$$

and  $\bar{x}_1, \bar{x}_2, \bar{x}_3$  are the mean values of the variables  $x_1, x_2, x_3$  and are given by

$$\bar{x}_1 = \frac{1}{n} \sum_{i=1}^n x_{i1} \quad (2.37)$$

$$\bar{x}_2 = \frac{1}{n} \sum_{i=1}^n x_{i2} \quad (2.38)$$

$$\bar{x}_3 = \frac{1}{n} \sum_{i=1}^n x_{i3} \quad (2.39)$$

Thus the least squares regression hyperplane is given by

$$\hat{y} = \hat{\eta}_0 + \hat{\eta}_1 x_{i1} + \hat{\eta}_2 x_{i2} + \hat{\eta}_3 x_{i3}, \quad i = 1, \dots, n \quad (2.40)$$

### 2.8.1 Analogy of MLR with neural network

The method of identifying the relationship between the response variable  $y$  and the causal variables  $x_1, x_2$  and  $x_3$  using MLR is analogous to the procedure of identifying relationship between the input and the output variables using neural networks. The parameters  $\eta_1, \eta_2$  and  $\eta_3$  are similar to the weights of the network. The parameter  $\eta_0$  is same as the bias. The regression parameters are estimated by the principle of least squares by minimising the sum of squared deviations similarly the weights and the bias in the neural network are adjusted by minimising the cost function.

The difference between the neural networks and MLR is that the input-output relationship is linear in MLR where as the relationship is nonlinear in neural network. The neural network method unlike the MLR method does not force predicted values to lie near the mean values and thus preserves the natural variability in the data.



# Chapter 3

## Porosity prediction

### 3.1 Introduction

For predicting porosity from well logs we have tested different methods, using a single neural network (Helle et al., 2001) and the committee machine (CM) approach (Bhatt and Helle, 2001a). In the study with single neural network the basic network is a 3-layer MLP network as shown in Figure 2.4. The input layer has 3 neurons corresponding to density, sonic and resistivity logs. We did not use the neutron porosity log because there is quite often problem with tool calibration of neutron log as it is not quantitatively consistent between different service companies. The hidden layer has 7 neurons and the output layer has 1 neuron.

In the committee machine approach we trained 20 networks on the same training dataset with random initial conditions of weights and bias with the goal that different networks will converge differently on the error surface. Out of those 20 networks we selected 9 networks, which gave minimum bias and variance on the validation set. The networks outputs are combined by ensemble averaging (Naftaly et al., 1997) and OLC method (Hashem, 1997). The main aim of using CM is to obtain a better porosity prediction by a combination of networks instead of finding a single network by a trial and error approach. Before applying a new method to real data the common practice is to use synthetic data as it gives us the flexibility to design the best architecture of the neural network by varying the number of patterns or hidden neurons or the noise level.

### 3.2 Synthetic data

The synthetic data for predicting porosity from well logs is defined as follows: The fluid saturation profiles for each fluid type; i.e. water, oil and gas, can be defined in terms of partial fluid saturation of the components which satisfies the following equation

$$S_w + S_o + S_g = 1 \quad (3.1)$$

The density  $\rho_f$  of the composite pore fluid is given by

$$\rho_f = \rho_w S_w + \rho_o S_o + \rho_g S_g \quad (3.2)$$

where  $\rho_w = 1.03$ ,  $\rho_o = 0.75$  and  $\rho_g = 0.25$  g/cm<sup>3</sup>.

A bulk density profile is then given by a linear trend superimposed by a sine-function to cover a realistic range of values (as shown in Figure 3.1). Similar profiles are given for the density  $\rho_m$  and transit time  $\Delta t_m$ , of the matrix grain material and the porosity can then be computed from the density equation

$$\phi = \frac{(\rho_m - \rho)}{(\rho_m - \rho_f)} \quad (3.3)$$

Given the porosity profile and grain properties we compute the sonic log  $\Delta t$  from Wyllie's equation

$$\phi = \frac{(\Delta t_m - \Delta t)}{(\Delta t_m - \Delta t_f)} \quad (3.4)$$

where the average transit time for the composite pore fluid is given by

$$\Delta t_f = \Delta t_w S_w + \Delta t_o S_o + \Delta t_g S_g \quad (3.5)$$

where  $\Delta t_w = 189$ ,  $\Delta t_o = 238$  and  $\Delta t_g = 625$   $\mu$ s/ft.

Assuming clean sandstone the resistivity log  $R_t$  can now be computed from Archie's equation

$$R_t = \frac{FR_w}{S_w^n} \quad (3.6)$$

where the resistivity of water  $R_w = 0.14$  ohmm, the formation factor  $F$  is given by  $F = \frac{a}{\phi^m}$ ,  $a = 0.625$ , and the exponents  $m = n = 2$  using typical values for a North Sea reservoir.

With the selected porosity model we preserve a fairly linear relationship with bulk density (3.3) and transit time (3.4), except for the small non-linear perturbations introduced by the independent variations in fluid density  $\rho_f$ , the matrix density  $\rho_m$  and transit time  $\Delta t_m$  of the rock material. Further non-

linearity between porosity and the logs is introduced through Archie's equation (3.6). The input model is shown in Figure 3.1 and some final logs are shown in Figure 3.2. Four different data sets of total 3000 samples were created with independent random noise of 0, 2.5, 5 and 10 percent added to each log. Subsets were then selected at regular depth intervals for the training patterns (50-150 samples).

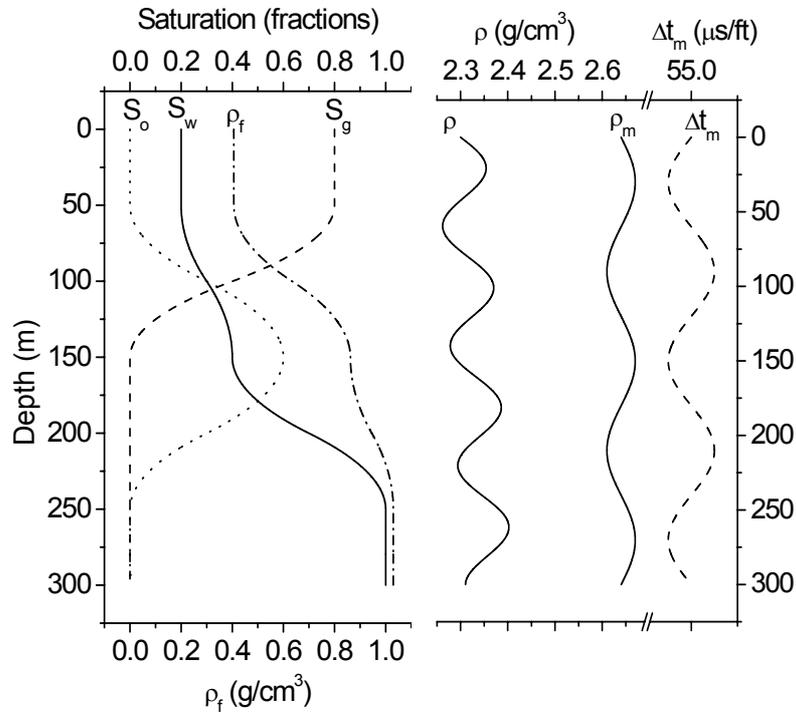


Figure 3.1: Model inputs for the synthetic porosity logs.

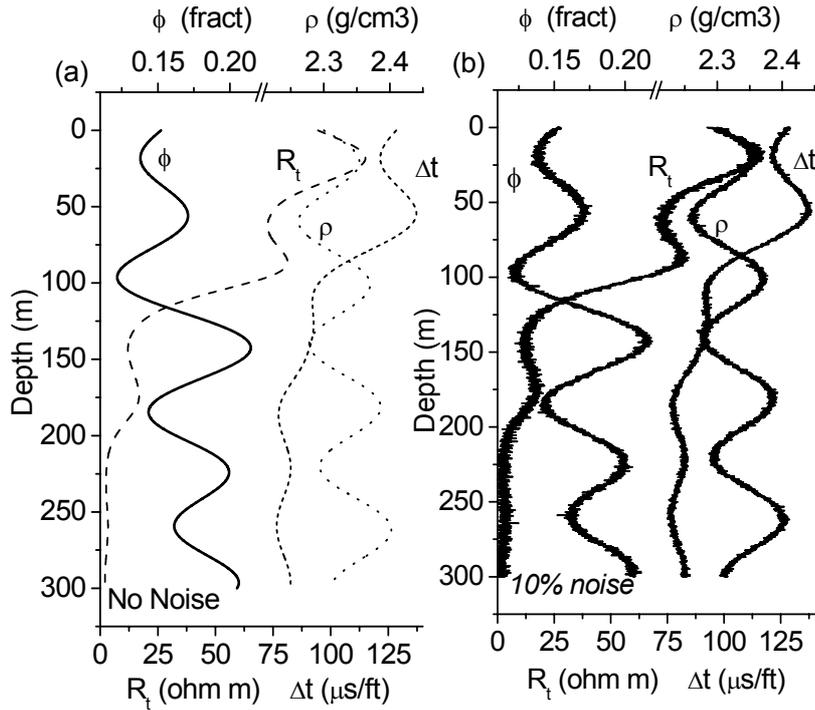


Figure 3.2: Model synthetic logs for porosity network: a) noise free b) with 10% noise

### 3.3 Optimal design of networks

There are two questions in neural network design that have no precise answer because they are application-dependent:

- i) How much data do we need to train the network?
- ii) What is the best number of hidden neurons to use?

In general, the more facts and the fewer hidden neurons, the better. There is, however, a relationship between the number of facts and the number of hidden neurons. Too few facts or too many hidden neurons can cause the network to memorise, implying that it performs well during training, but tests poorly and fails to generalise.

There are no rigorous rules to guide the choice of number of hidden layers and number of neurons in the hidden layers. However, more layers are not better than few, and it is generally known that a network containing few hidden neurons generalises better than one with many neurons (Lawrence, 1994). As discussed by Poulton (2001) the network performance can be considered a quadratic function of the number of hidden neurons so a decrease in number could result in increased performance. With too few hidden neurons the network cannot make adequately complex decision boundaries and with too many it may memorise the dataset. Many researchers have suggested the optimum number of hidden neurons could be the geometric mean of the input and the output neurons if the number of the output neurons is less than the input neurons. For instance, if the relationship between input and output is known to be almost linear we may emulate the linear regression by choosing the number of independent connections; i.e. neuron weights, equal to the number of independent coefficients in the regression equation. Then, a few neurons may be added to the hidden layer in order to account for non-linearity between input and output. On the other hand, the optimal configuration can only be achieved by testing and by learning through experience with the data and problems at hand.

Most neural networks studies reported in geoscience provide only single realisations of the predictions (e.g. Huang and Williamson, 1997; Helle et al., 2001), and the best network among a number of individually trained networks is then selected by trial and error. The latter is a time consuming process does not always guarantee an optimal solution.

Since the optimal network architecture is application-dependent we have to determine its configuration by numerical experiments with the problem and data at hand. So for identifying the best architecture of the network we used synthetic data as discussed before. The topology of the neural network is same as shown in Figure 2.4 with three inputs as density, sonic and resistivity and one output neuron i.e. porosity. The optimal number of hidden units remains to be determined and, as demonstrated in Figure 3.3, we may determine the appropriate size of the hidden layer by doing an error analysis on the outputs using different number of hidden neurons and different number of samples. For this test we use a small CM with only 5 experts and simple averaging for each point to establish more representative results than is possible with a single network. The experts are trained by using 150 facts, selected at regular intervals along the 3000 sample synthetic logs. The experts are purposely over-trained to reduce the bias. We calculate the error by making a difference between the output from each CM with the desired output on all the 3000 samples in the log. A statistical analysis of the error is done by calculating bias and variance. In Figure 3.3 and Figure 3.4 bias is the mean of the error and standard deviation ( $\sigma$ ) is the variance of the error. Thus the standard deviation has the main contribution towards error.

For noise free data both the standard deviation and the bias decay asymptotically to a constant level for a network with more than 10-12 hidden neurons as shown in Figure 3.3a. A consistent feature for all noise levels is the sudden drop in error obtained by increasing the number of hidden units from 4 to 6. Except for noise-free data the standard deviation increases for networks with more than 10 hidden units, while the trend for the bias seems less predictable. In general, however, there is a tendency for the bias to decrease with an increasing number of hidden units, which is consistent with the findings of Parmanto et al. (1996) who has dealt with somewhat simpler problem and network configuration.

Comparing the bias and variance gives insight into the role of noise in over-fitting. For clean data over-fitting is not a problem and we can afford to have a network with higher degree of complexity. More hidden neurons may not improve the performance, but it does not hurt. The problem of over-fitting appears only when the data is noisy. Even with a low noise level of 2.5% the complexity becomes a concern since the standard deviation starts increasing after 10 hidden neurons where the standard deviation is at its minimum for all noise levels tested. The standard deviation surpasses the bias as the leading contributor to the total error as the network complexity grows.

From this experiment it is obvious that the porosity network should have more than 6 and less than 12 hidden units because for all the noise levels standard deviation starts increasing as the number of neurons increases from 10 to 12 hidden neurons. Thus based on the results in Figure 3.3 we have chosen 10 hidden neurons for the porosity network as a fair compromise, implying a total error of less than 0.001 in porosity with reasonably noisy data.

A major assumption in network design is that a sufficient number of well-represented training samples are available. In many real situations, however, training samples are expensive to obtain. Since the problem complexity dictates the size of the network, we may end up with fewer facts than required for a network of a given size. For the present problem, the number of representative core samples is normally the main limitation. The challenge in network design has been the small number of available data, and we can overcome this problem by generating virtual samples as discussed in Cho et al. (1997) and in Wong et al. (2000) as a remedy for insufficient number of training samples.

The benefit of using synthetic data, compared with the real-world data, is thus obvious since it allow us to vary the sampling of the training set as we wish. By varying the number of patterns from 75 to 200, sampled at regular intervals from the complete synthetic logs, for each noise level, we have obtained the results shown in Figure 3.4. Again, the standard deviation and bias (in porosity fractions) are measured relative to the complete set of 3000 porosity samples. As with the number of hidden neurons, also in case of training patterns we find a distinct drop in error above a certain threshold value dependent on noise level.

For noise-free and low-noise (2.5 %) data virtually no improvement can be detected with more than 100 training patterns. For the more noisy data, with 5% and 10 % noise, we apparently still gain accuracy when using more than 150 samples. On the other hand, with 150 training patterns the porosity data is reproduced within a practical error limit of 0.001 in porosity. In general, the latter result is also in agreement with Parmanto et al. (1996).

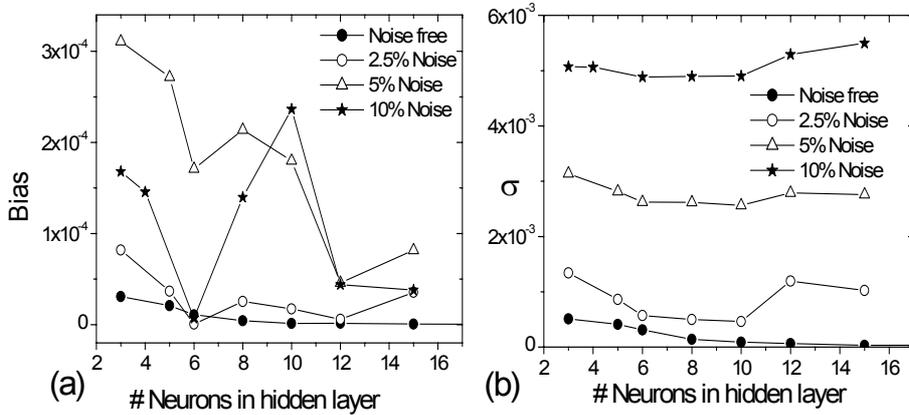


Figure 3.3: Variation of a) bias and b) std. deviation with number of neurons

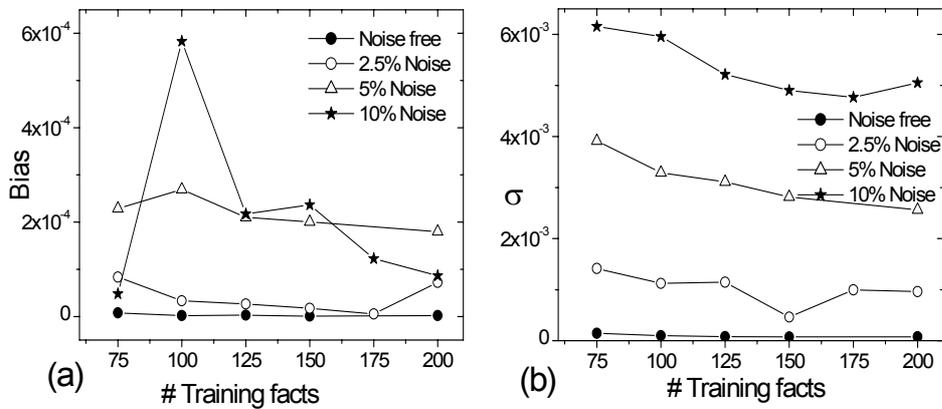


Figure 3.4: Variation of a) bias and b) std. deviation with number of training facts

### 3.4 Training strategy

We compare the performance of the individual trained networks by a cross-validation approach and by over training. In the cross-validation approach the networks were trained on 80% of the data and simultaneously tested on the remaining 20% of the data, in order to avoid over fitting of the data. The training was stopped as soon as the variance on the validation set started to increase, so that the network produced a minimum of bias and variance on the test set (Figure 2.5). If the network is allowed to continue the training beyond this stage it memorises the data from the training set and gives a poor generalisation on the test set. Figure 3.5 shows that the networks trained with the cross-validation approach give smaller error than those trained with the over training approach. However after taking the ensemble average and the OLC the two approaches do not differ significantly because variance is reduced by ensemble averaging and bias is reduced by OLC. In the cross-validation approach the network learns the predictor function by using only 80% of the dataset which may be sometimes insufficient for function approximation. So for a single network the best solution is to perform training by the cross-validation approach, while for a committee machine over training can also be the solution.

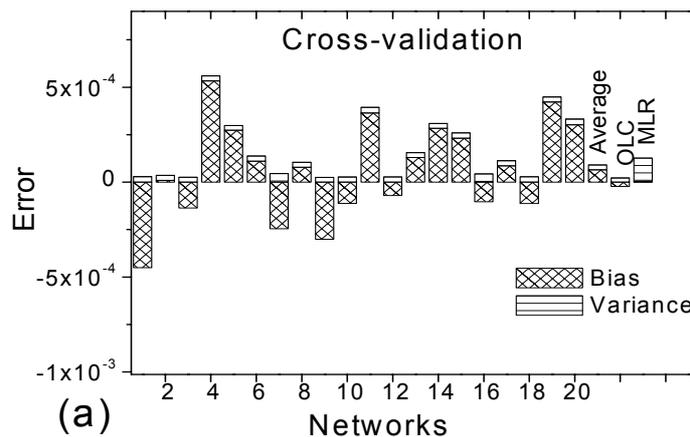


Figure 3.5a): Comparison of the bias and variance of the networks on the test dataset trained with cross-validation

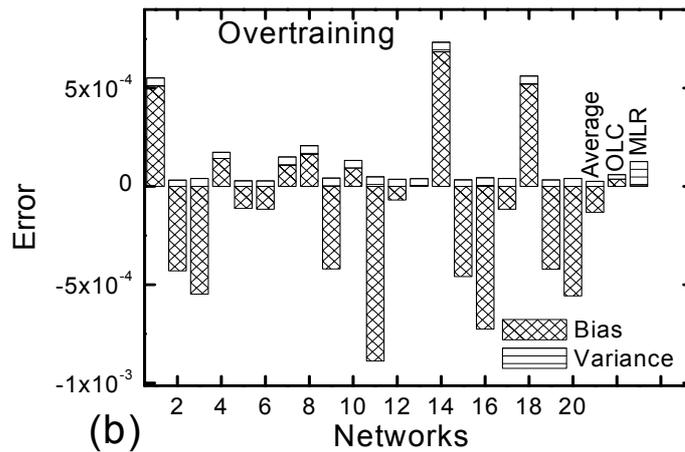


Figure 3.5b): Comparison of the bias and variance of the networks on the test dataset trained with overtraining approach

### 3.5 Comparison of alternative techniques

The resulting committee network for porosity, consisting of nine experts, has been applied to the synthetic data. We compare here the porosity predicted by the OLC approach and the MLR on noise free data, 2.5%, 5% and 10% noisy data on test set. It is quite obvious from Figure 3.6 and Figure 3.7a that the MLR fails to reproduce the model data due to the small non-linearity in the porosity relationship whereas the neural network approach, on the other hand, is capable of handling non-linearity with any desired accuracy. In contrast to MLR models the neural network approach does not force the predicted values to lie near the mean values and thus preserves the actual variability in the data (Rogers et al., 1995) as shown in Figure 3.7b. Even for noisy data the errors of the individual experts are far below any practical level, the ensemble average has significantly reduced the bias and variance also and the OLC further eliminated the bias and reduced the variance as shown in Figure 3.7a. We also obtained similar results for 2.5% and 5% noise level also.

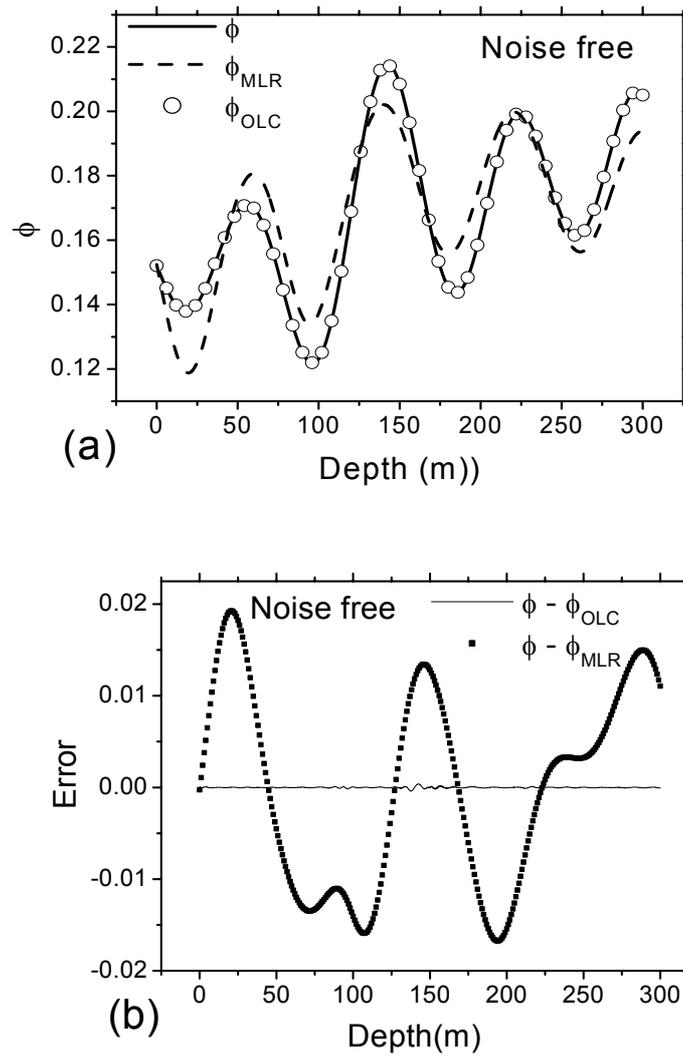


Figure 3.6: Comparison of OLC approach and MLR on noise free data, a) predicted porosity b) error in porosity prediction

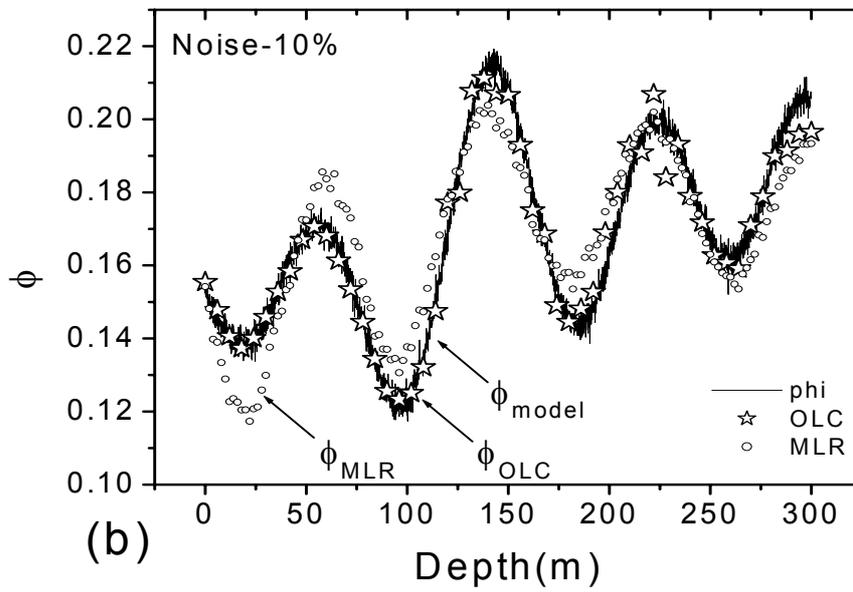
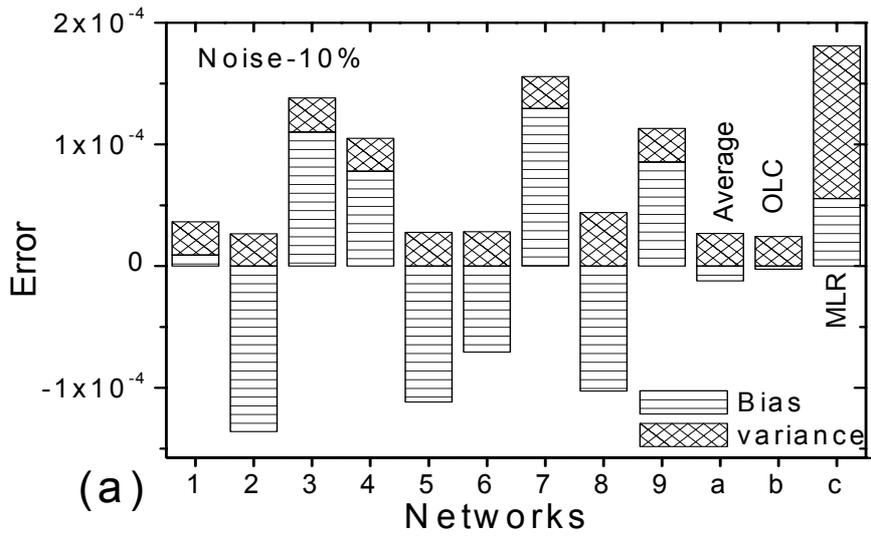


Figure 3.7: Comparison of OLC approach and MLR on data with 10% noise, a)variation of bias and variance, b) predicted porosity

## 3.6 Real data

In our first study (Helle et al., 2001) we used a single neural network for predicting porosity from well logs. The network was trained by the Levenberg-Marquardt back-propagation algorithm. The training facts were dominated by non-reservoir intervals from Tertiary to Jurassic levels. The majority of the porosity values were based on grain density laboratory measurements and bulk density from wireline data (Lucas, 1998). These grain density values have been used as matrix density for calculating porosity from the density porosity transform. The samples of grain density were carefully selected to obtain a range of values appropriate for most sediments in the Viking Graben (Bhatt, 1998) for use in a basin-scale fluid flow analysis. The main advantage of using porosity derived from the density measurements is the fact that these are the best possible estimates of in situ porosity values since the compressibility of the pure grain material is likely to be small compared with that of the matrix. The grain density in the laboratory is thus not very different from in situ values, and hence the porosity estimates are less prone to pressure corrections than those based on core plugs. The network consisted of 3 neurons in the input layer, 7 neurons in the hidden layer and 1 neuron in the output layer. This network has been tested on different wells in various fields. It revealed excellent over-all characteristics when applied to the entire geological section as well as in the fine details of a water-bearing reservoir. Figure 3.8a displays the results of this network in the reservoir section of well 31/4-3. The comparison made between predictions and core helium porosity reveals striking similarities indicating, after all, that core and well data may be fairly consistent.

Porosity measured in the laboratory is normally not corrected for overburden pressure as a result of which the porosity measured in the laboratory may be higher than porosity in insitu conditions. It is only sometimes that special core analysis (SCAL) is performed in which the core is first compacted to reservoir pressure level and then further analysis for porosity, permeability etc. is done. Figure 3.8b shows the result of special core analysis studies performed in well 30/6-9 where the variation of porosity was studied as a function of pressure when a core was brought from the in situ to the laboratory conditions. The effect is different for different samples. Thus it is difficult to standardise a particular formula for pressure correction of porosity. The effective correction could vary from 5 to 12 porosity units in extreme cases e.g. as shown in Figure 3.8b.

For calculating porosity from grain density the pore fluid was assumed to be brine of density  $1.03 \text{ g/cm}^3$  since no samples were taken from hydrocarbon bearing sections. In order to adjust the initial network to account for the various pore fluids we added a few data points taken from the hydrocarbon reservoirs. The training patterns cover the porosity range 0.02 – 0.55 from different fields. Out of a total of 81 facts only 14 facts are taken from the main test area which is

the Oseberg-field. The capability of the resulting modified network to account for different pore fluids can be appreciated in Figure 3.9. We compare the porosity predicted by the neural network with the one predicted by the density-porosity transform equation (3.3) using a constant grain density  $\rho_g=2.64 \text{ g/cm}^3$  and with alternative fluid densities  $\rho_f = 0.25, 0.75$  and  $1.03 \text{ g/cm}^3$  for gas, oil and brine respectively, in three wells with water saturated, oil saturated and brine saturated formations. The corresponding porosity transforms of  $\phi_\rho$  reveal strong sensitivity to the pore fluid density, with differences of 0.05 to 0.1 units between the results of assuming brine versus gas filled rock. There is a strong response to pore fluid in the transform as shown in Figure 3.10. Thus after knowing the pore fluid density the porosity can be calculated by using the density-porosity transform.

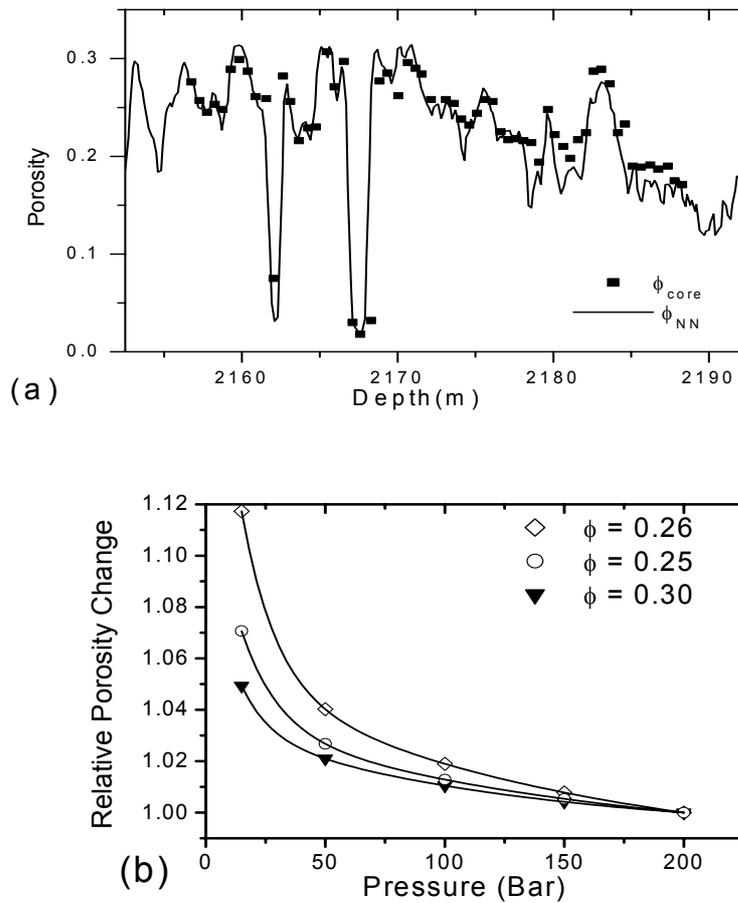


Figure 3.8: a) Comparison of core porosity and porosity predicted by neural network in well 31/4-3 b) pressure variation on porosity (SCAL 30/6-9)

An additional correction term for clay content (Sclumberger, 1989), and a variable  $\rho_g$ , are normally included to improve the accuracy of the transform implying the need for additional core data and input from log interpretation to obtain shaliness from the gamma ray log. The porosity predicted by the neural network in all the three wells reveals quite a good match with the core porosity. The porosity predicted by neural network follows the porosity calculated by density transform but the advantage of the neural network approach is that no information about pore fluid and no correction for shaliness is essential, only log data is required, once the network has been properly tuned to the area.

In Figure 3.9a the neural network porosity and porosity from the density transform are better correlated than with core porosity. The reason is because of the different conditions of measurements for core and log. There is an underprediction of porosity by neural network by 0.5-2 porosity units as this is a test well and so the training dataset may not be fully representative of the well data. Whereas in case of porosity from the density transform we can tune the fluid density which fits the core porosity.

In Figure 3.9b the neural network porosity is predicting about 2 porosity units higher than the porosity from the density transform and the core porosity. The reason is that this is also a test well in comparison to the density and sonic values in training dataset this well is having a low density value of about 0.75-1 gm/cc and a high sonic travel time of about 5-7  $\mu$  sec/ ft. Due to this there is an overprediction of porosity in this well. This can be remedied by building more regional networks so that the wells in that area are fully represented in the training dataset.

In Figure 3.9c there is a small scale variation in core data due to thin layers at the bottom of Oseberg. These layers have been also observed in many other wells of the Oseberg field. The layers are beyond the resolution of standard log data. As a result of this the porosity predicted by logs both by the density transform and neural network are uniform in comparison to high variations in core porosity. There is a similarity in the trends of the three porosities that below a depth of 4090m are all increasing slightly but due to the thin layers the core porosity has higher variations. The deviation between neural network porosity and the density porosity transform is due to the difference in density and sonic log response below the depth of 4118m as density started increasing leading to low porosity from the density transform whereas sonic has also a rising trend due to which neural network porosity is increasing. Also at the shale-sand transition at the top of the reservoir interval there is a peak in the neural network porosity, which is not present in the core, or the density porosity. While the density is virtually constant the sonic reveals two distinct peaks at the top and bottom of the reservoir. The peak in the neural network porosity comes due to the peak in sonic travel time. This feature has been observed in several of the wells in the Oseberg field and is thus considered to be a real low-velocity event. Thus, here the neural

network prediction based on input from three logs compares less favourably with the helium core porosity than that from density alone.

In Figure 3.11 we show cross plots for wells 30/9-1, 30/9-B-40 and 30/9-B-2 for a comparison between the performance of neural network porosity and density porosity with core porosity. The cross plots reveal that the two predictions are very similar.

For the quantification of the error we calculate the difference between the output from the neural network and core porosity. We generate a histogram to see the error distribution, on the error histogram we apply a Gaussian fit and calculate the mean and standard deviation of the probability distribution function. A Gaussian fit to the error distribution for the three wells gives a mean error of 0.01 in porosity and a standard deviation of 0.015, which is sufficiently low for all practical purposes.

Being motivated by the benefits of CM architecture, with the introduction of ensemble averaging and OLC approach which further improves the results, we performed the second study (Bhatt and Helle et al., 2001a) by using CM for predicting porosity. The argument is that while one network reproduces the main patterns, the others may provide the details lost by the first. If all the errors of committee members are perfectly correlated, then using a committee does not improve performance. However if the error patterns are uncorrelated, performance of the committee dramatically increases and asymptotically approaches perfect performance. Therefore it is beneficial to make the errors among the networks in the committee less correlated in order to improve the committee performance. The common method for constructing a committee is to train an ensemble of networks independently, using the same learning dataset. The networks in the committee have been reported as achieving significant improvement over individual network performance in regression and classification tasks (Hashem, 1997; Parmanto, et al., 1996).

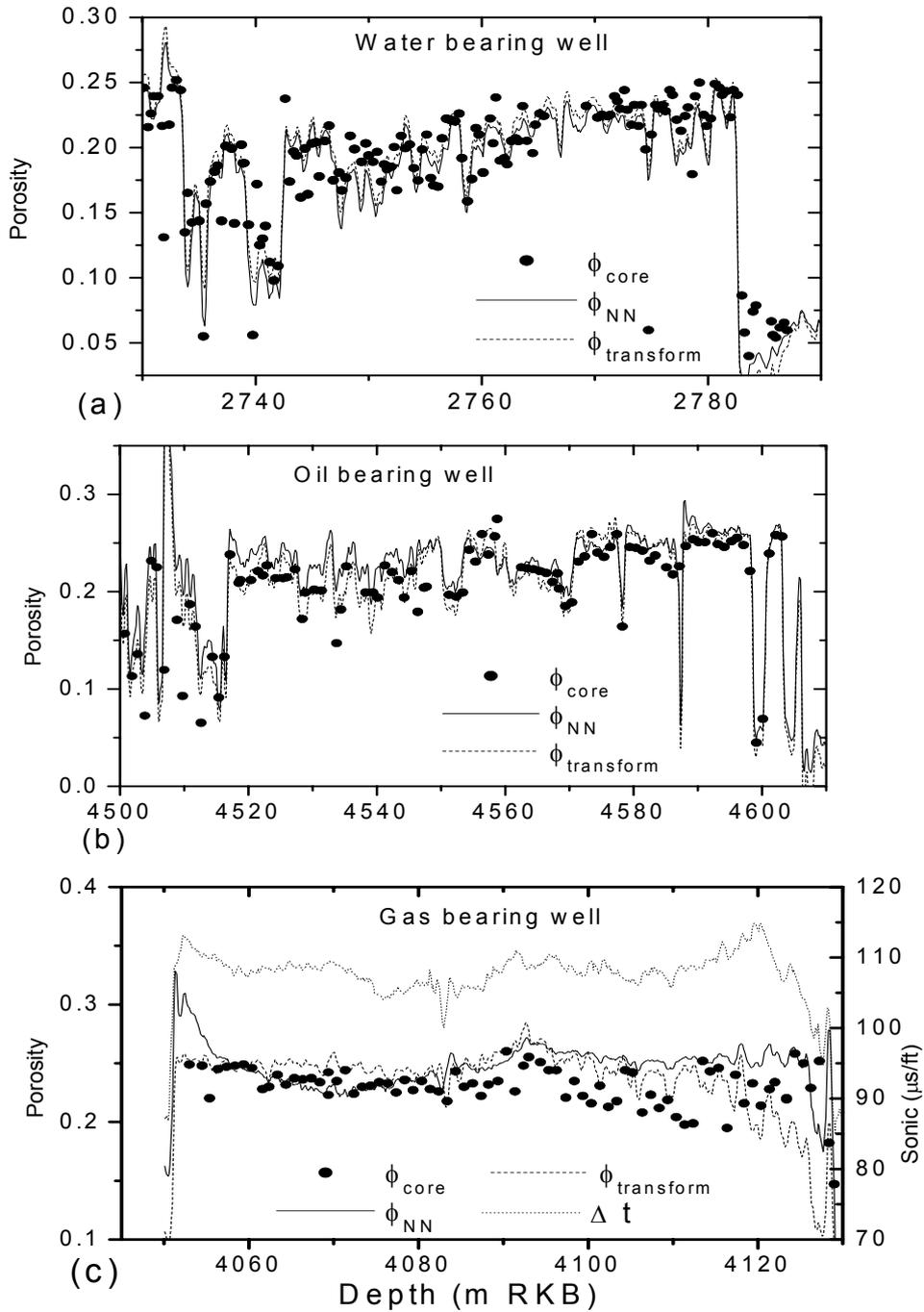


Figure 3.9: Comparison of neural network porosity and density-porosity transform with core porosity for well; a)30/9-1, b)30/9-B-40, c)30/9-B-2

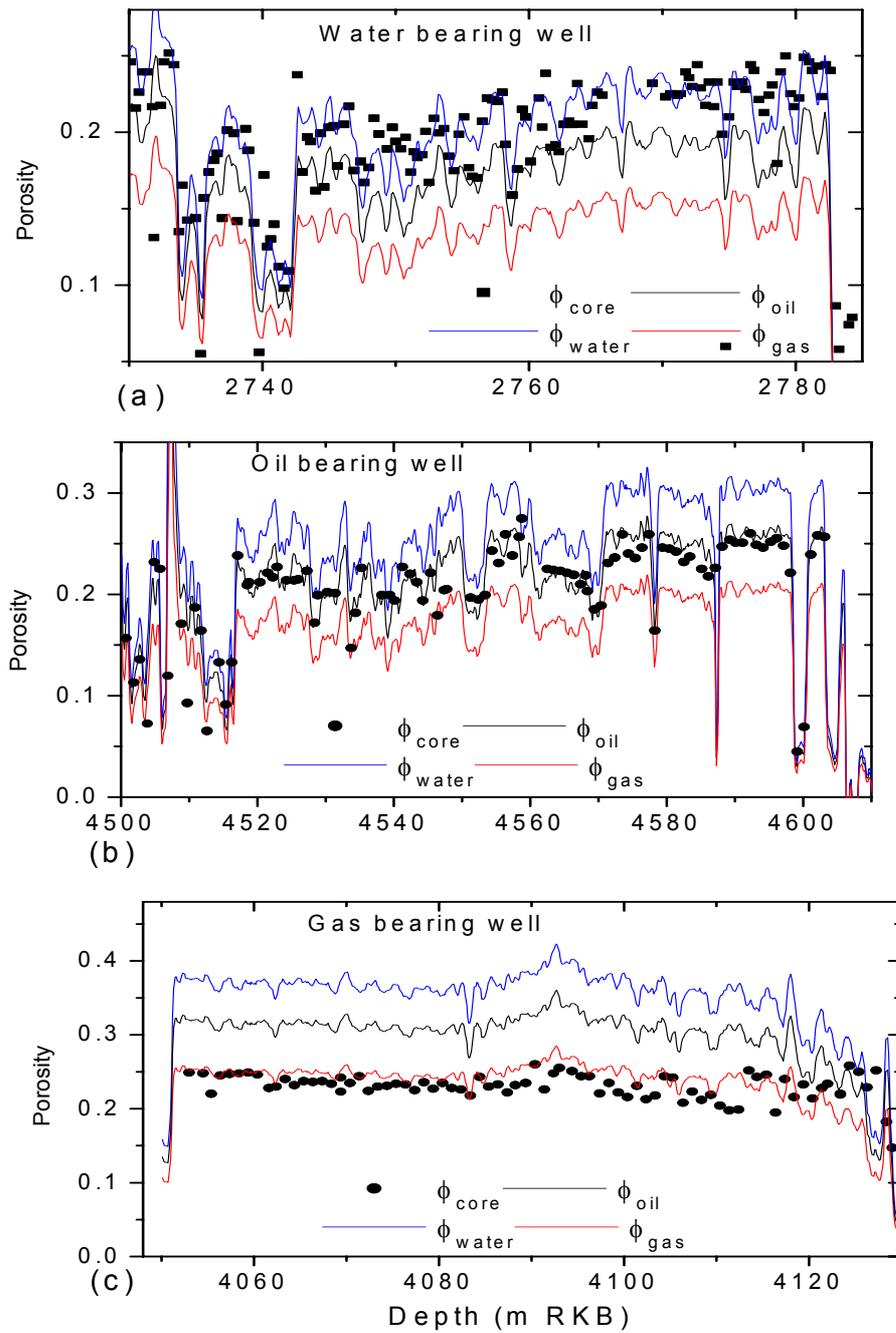


Figure 3.10: Testing the sensitivity of density- porosity transform for different pore fluids in well; a)30/9-1, b)30/9-B-40, c)30/9-B-2

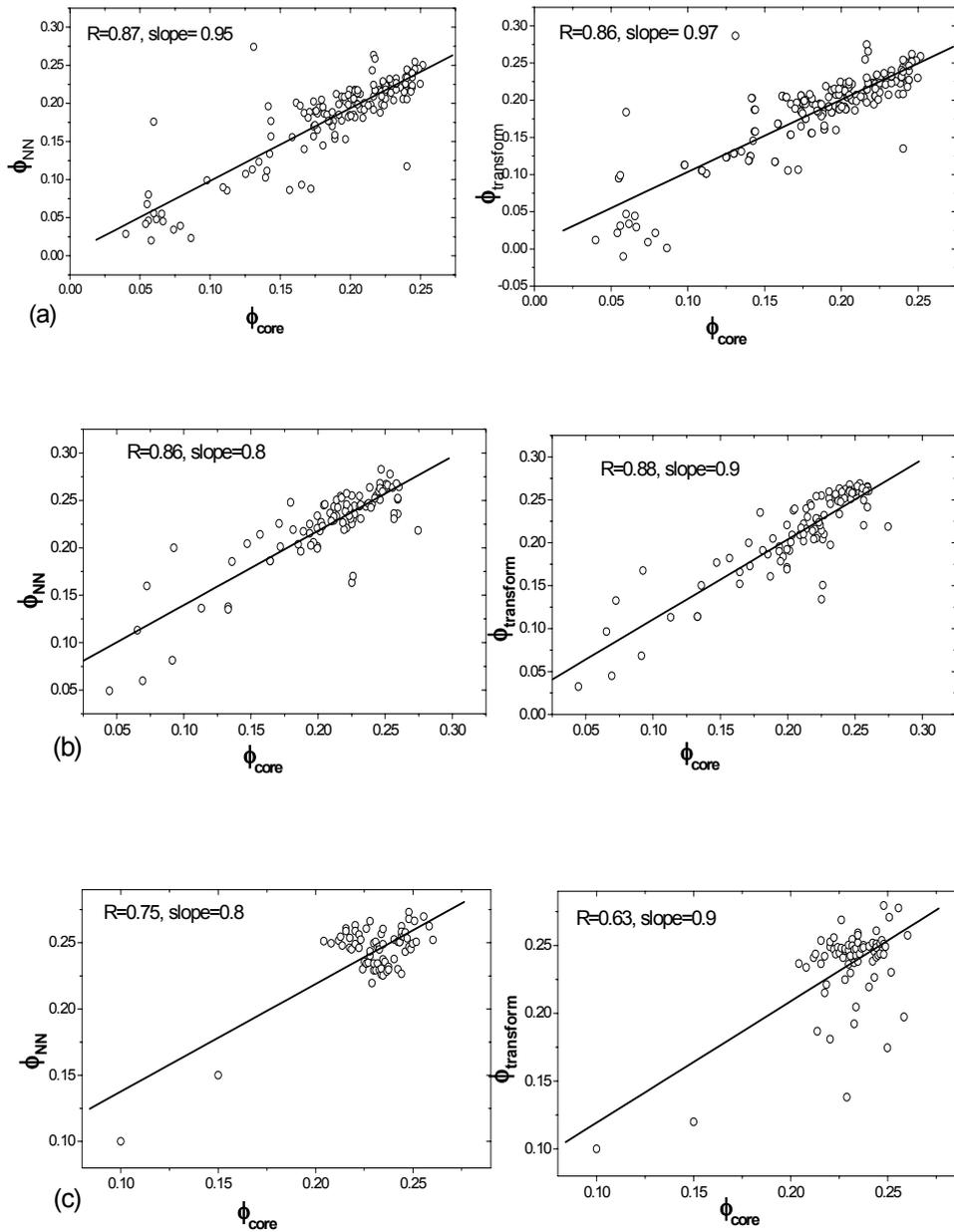


Figure 3.11: Crossplott of core against NN porosity and core against porosity from density transform in well: a)30/9-1, b)30/9-B-40, c)30/9-B-2

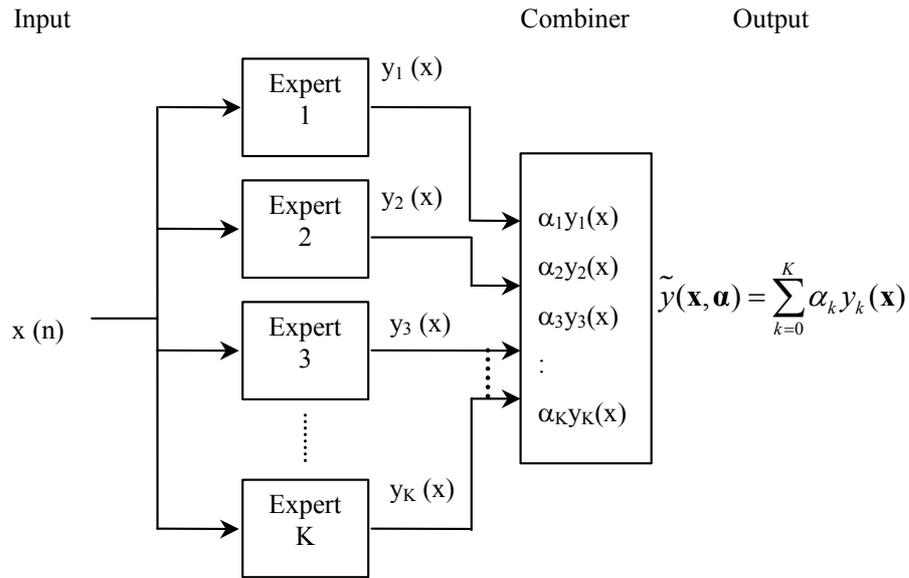


Figure 3.12: Block diagram of a CM for porosity prediction

The architecture of a committee is shown in Figure 3.12. Here each expert is a single neural network having 3 neurons in the input layer, 10 neurons in the hidden layer and 1 neuron in the output layer as used with the synthetic data. We train 20 such networks with the same input data but with different initial weights. The networks are overtrained to reduce the bias at the cost of variance. Out of these 20 networks 9 networks with minimum bias were selected. As suggested in Sharkey et al. (2000) the performance of neural networks can be improved by combining several redundant networks. The ensemble creation can be done by ‘test and select’ approach, which involves testing the performance of individual networks on a test set and then selecting the best out of them. The output from them is combined using the ensemble average (Naftaly et.al., 1997) or the unconstrained OLC approach (Hashem, 1997). The ensemble average gives an average of the bias and variance of all the networks whereas the OLC approach gives an additional fit to the data by removing the bias and reducing the variance showing that it is the best technique for porosity prediction.

Based on the results from synthetic data as shown in Figure 3.3 and Figure 3.4 as discussed in section 3.2 the optimum number of hidden neurons that give minimum bias and variance should be between 6-12 and the number of training patterns should be more than 150. Therefore we chose the optimum number of hidden neurons as 10 for training each expert of the committee. More training data from different wells of the Oseberg field have been added to the training dataset of the previous study, which had 81 training patterns. The total number of training patterns is now 168 for training each expert of the committee.

The variance distribution for the experts of the committee is surprisingly constant as shown in Figure 3.13a, and confined to a level that reflects the practical limits in accuracy for matching core data and wireline logs, equivalent to a standard deviation of about 0.02 in porosity. Ensemble averaging and OLC imply only a minor reduction in the variance while the bias is effectively removed by OLC in accordance with the theory. The neural network predicted porosity correlates very well with core porosity for the training dataset as shown in Figure 3.13b.

The technique has been tested on several wells of Oseberg field and found to be successful. Figure 3.14 shows the results on the gas and oil bearing well 30/9-B-20 with a gas oil contact at 3198m. The well is inclined with a hole deviation of 40 degrees. The porosity has been predicted by neural network and compared with the core porosity in the Brent formation. The core data consisted of 380 core points. In general the correlation between the neural networks predicted porosity and core porosity is good as shown in Figure 3.14b. There is an insignificant bias and a standard deviation that essentially reflects the scatter in the underlying measurements. The Gaussian fit to the difference between core and neural network predicted porosity gives a negligible mean and standard deviation of 0.04, as shown in Figure 3.14c which is mainly due to the scatter in the core porosity and neural network porosity. The scatter is mainly in the Ness formation (3140-60m) because it consists of thin beds of sand, shale and coal, which are beyond the resolution of log data. Below this interval is the Oseberg formation, which is a homogeneous sand body, and therefore the match between the two porosities is very good. Above the Ness formation i.e. in the interval from 3110 to 3135m is the Tarbert formation, which is fairly homogeneous within each lithology of sand and shale sequences. In general there is a good match between the two porosities. Figure 3.15 shows the comparison of porosity predicted by neural network and core porosity in an oil-bearing well, 30/9-B-24. This well is also inclined with a hole deviation of 35 degrees. The porosity has been predicted by neural network and compared with core porosity in the Brent formation. The well has 161 core points. The porosity predicted by neural network matches very well with the core porosity as shown in Figure 3.15b. The Gaussian fit to the difference between core and neural network predicted porosity gives a negligible mean and standard deviation of 0.01 as shown in Figure 3.15c. The core data is available in the Etive (3380-90m), Rannoch (3390-3393m) and the Oseberg formation (3393-3448m). The match between the two porosities is very good in the Etive, Rannoch and the upper part of the Oseberg formation (3393-3428m). In the lower part of Oseberg formation (3428-3448m) there is very fine layering which has a spatial scale of less than 0.1m. These fine layers cannot be detected by the logging tools, which have a spatial resolution of about 1m for resistivity tool and 0.3m for density and sonic tools. Figure 3.16 shows a core photograph displaying the small-scale heterogeneities in the core, which are beyond the resolving power of logs. The neural networks, which are trained on log data, thus give a low-resolution prediction in comparison with cores.

However an important observation is that porosity is correctly predicted in all fluid zones by the same network, without explicit knowledge of the pore filling fluid, implying that also the network for real data is sensitive to pore fluid type and saturation.

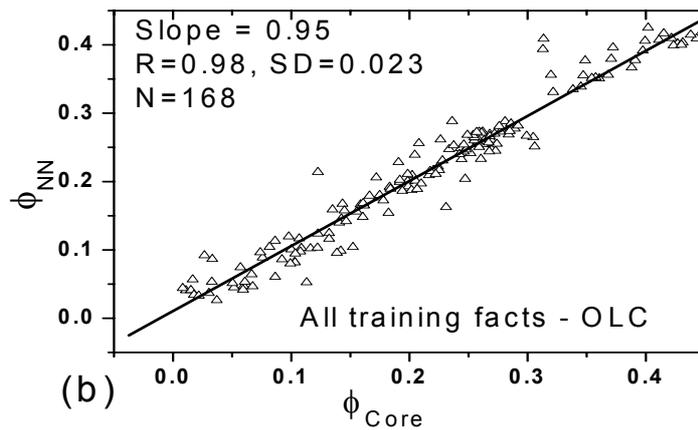
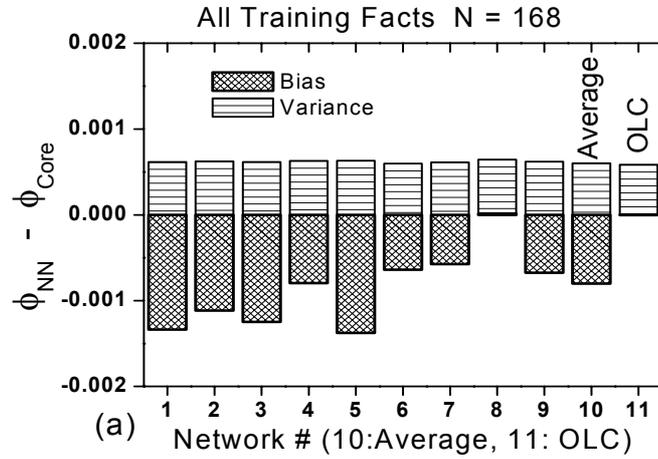


Figure 3.13 : a) Comparison of bias and variance of networks with ensemble average and OLC b) correlation of neural network porosity with core helium porosity

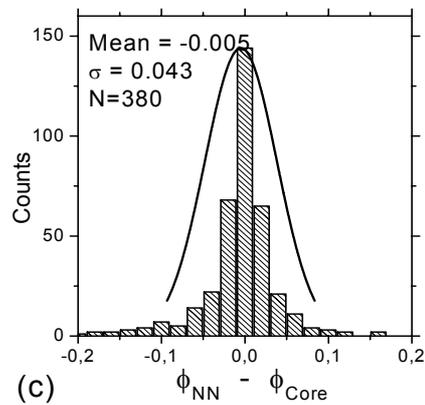
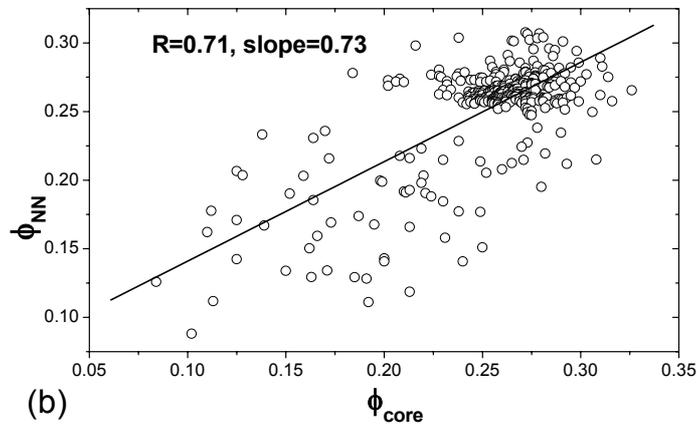
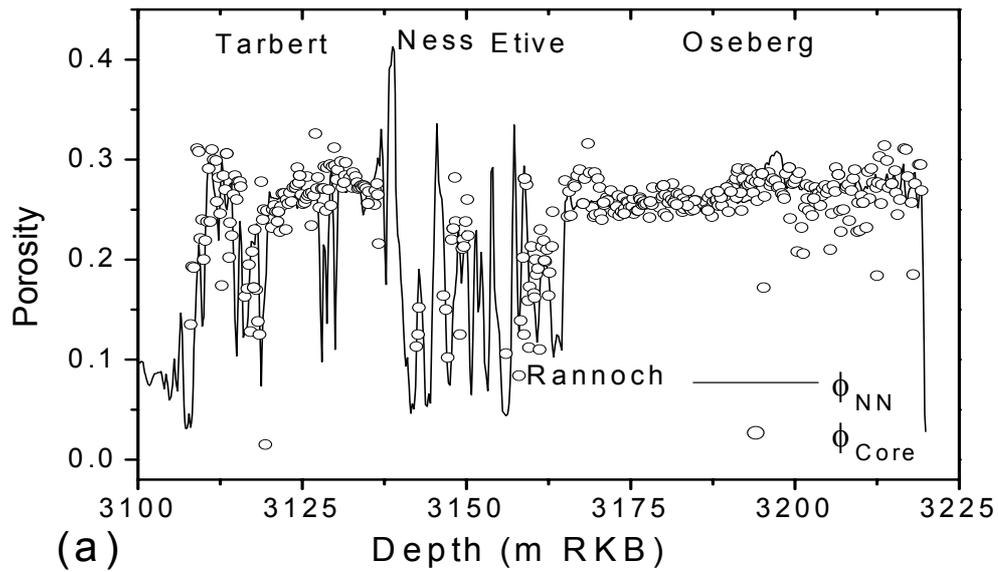
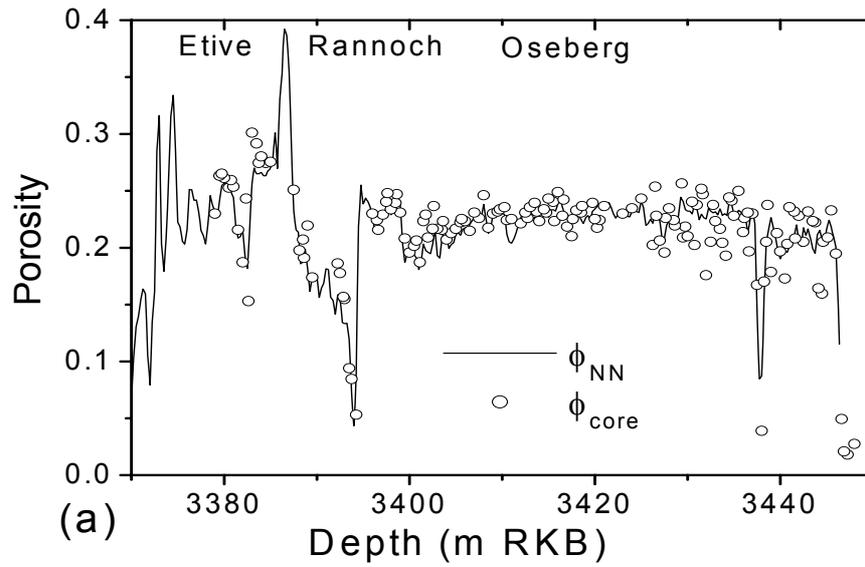
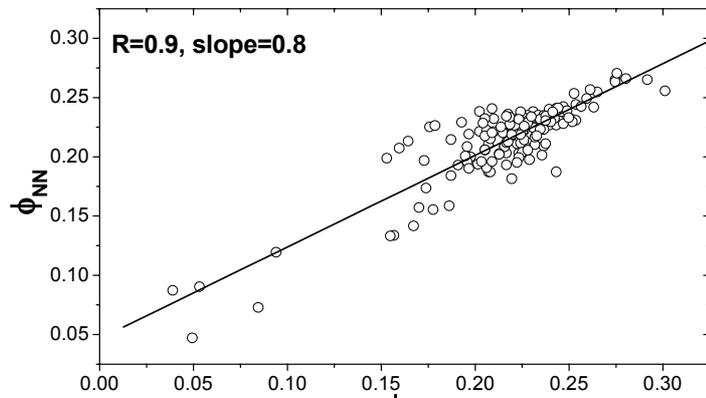


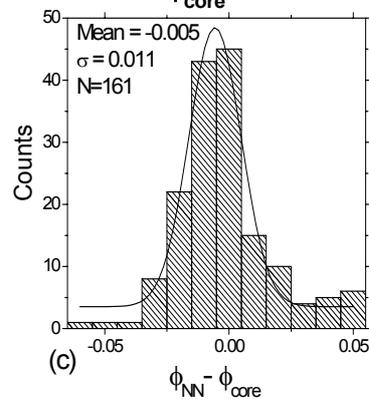
Figure 3.14: a) Comparison of core porosity and neural network porosity by OLC in well 30/9-B-20 b) crossplot of core against NN porosity, c) error distribution



(a)



(b)



(c)

Figure 3.15: a) Comparison of core porosity and neural network porosity by OLC in well 30/9-B-24. b) crossplot of core against NN porosity, c) error distribution

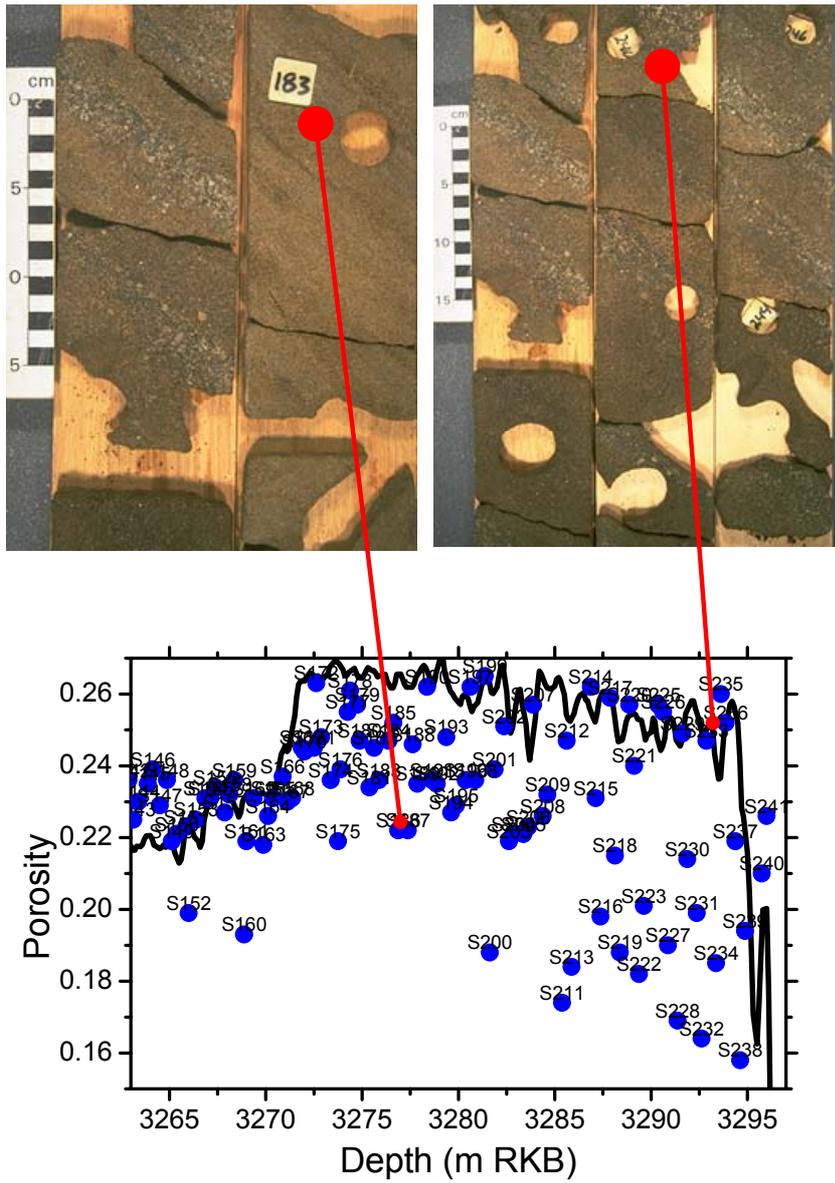


Figure 3.16: Comparison of log and core derived porosity values with the core photos.

## 3.7 Conclusions

The neural network approach to porosity conversion has a number of advantages over conventional methods, including empirical formulas based on linear regression models or the common semi-empirical formulas such as Wyllie's and the density equation. The neural net method represents a pragmatic approach to the classical log conversion problem that during the years has caused problems for generations of geoscientists and petroleum engineers. Instead of searching for complicated interrelationships among geological/geophysical properties the neural net approach requires no underlying mathematical model and no assumption of linearity among the variables.

The main drawback of the method is the amount of effort required to select a representative collection of training facts, which is common for all models relying on real data, and the time to train and test the network. On the other hand, once established the application of the network requires a minimum of computing time. For the porosity network we find that porosity values calculated from grain density (used as matrix density) and in situ bulk density data give more consistent results than using standard helium core porosity data. The benefits over the single neural network of the committee neural network where the predictions are redundantly combined are obvious.

It is essential to determine the design of the optimal architecture of the network using synthetic dataset. The optimum architecture in this case is 3 neurons corresponding to density, sonic and resistivity logs in the input layer, 6-10 neurons in the hidden layer and 1 neuron in the output layer. The number of training patterns should be about 150. More training data do not harm the accuracy, but increase the time for training the network.

From 20 porosity networks trained by the same patterns, but from random initial conditions, we have included the nine best in the CM based on a least error criterion when compared with test data. The unconstrained optimal linear combination of Hashem (1997), is the most suitable ensemble approach for the porosity CM and the accuracy is mainly limited by the accuracy of the training patterns and the accuracy of the log data themselves. The benefit of neural networks compared with the MLR technique has been briefly touched upon by showing that MLR fails to reproduce the minor non-linearity imbedded in the common log-to-porosity transforms, whereas the neural network reproduces the same data with high accuracy. In application to real data for one well the standard deviation error of the difference between prediction and helium core porosity data is 0.04, while for another well the standard deviation error is only

0.01. These differences, however, reflect the limitations in the techniques of measurement rather than errors in the numerical technique of neural networks.

Our porosity predictions are sufficiently accurate to satisfy most practical needs and are comparable with the accuracy obtained from the density equation. The network approach, on the other hand, requires no a priori knowledge of the grain material and pore fluid, and can thus equally well be applied while drilling without prior petrophysical evaluation. The network is sensitive to pore fluid type and saturation.

In this study we have not considered the effect of different drilling fluids for example using oil based mud or water based mud. It will be very important to see the behaviour of network with respect to different drilling fluid in future study.

# Chapter 4

## Permeability prediction

### 4.1 Introduction

Permeability is a critical parameter for the evaluation of hydrocarbon reservoirs. It is primarily dependent on porosity, clay content and grain sorting. There are several models relating the permeability with the grain size. The Kozeny Carman model (Dullien, 1991) is one of the most commonly known. Based on this model permeability is expressed as:

$$k = \frac{B\phi^3 \bar{D}^2}{T} \quad (4.1)$$

where  $\phi$  is the porosity,  $T$  is the tortuosity,  $B$  is a geometrical factor known as Kozeny constant and  $\bar{D}$  is effective grain size. The value of  $B$  can vary from 5 to 100 (Rose and Bruce, 1949). For an ideal, uniform and consolidated rock  $B=5$ . Due to the large variation in the value of  $B$  this model is not much used now. The most accurate method to date for measuring such an important property is core analysis. It is well known that closely spaced core permeability values are often not available because of unfavourable borehole conditions and high cost of coring. Well log data, however are abundant and are frequently used to infer permeability along the drilled wells. Although no well log is currently capable of measuring permeability directly, some of the methods for calculating permeability from well logs are as follows:

- i) Permeability estimates from porosity and irreducible water saturation:

Based on Kozeny Carman equation Wyllie and Rose (1950) proposed an empirical relationship between permeability, log derived porosity and irreducible water saturation ( $S_{wi}$ ) given by

$$k = \frac{C\phi^x}{S_{wi}^y} \quad (4.2)$$

where  $C$  is a constant depending on the hydrocarbon type (79 for dry gas and 250 for medium density oil)  $x$  and  $y$  are the parameters to be determined from log-log plot of  $\phi - k$  at a known value of  $S_{wi}$ . Several other investigators e.g. Tixier, Timur, Coates and Dumanoir (Schlumberger, 1989) have also proposed similar formulas. All these relations are restricted to sandstone.

ii) Permeability estimates from nuclear magnetism resonance (NMR) log:

The free fluid index (FFI) which is the volume of fluid that is not bound electrically or chemically to the clay lattice, to the surface of the rock matrix or to some other mineral lattice is measured by the nuclear magnetism resonance log. This is related to irreducible water saturation by

$$S_{wi} = 1 - \frac{FFI}{\phi} \quad (4.3)$$

Thus irreducible water saturation can be determined if porosity and FFI are known. Then equation 4.2 or a similar relation can be used for calculating permeability. Thus by this technique permeability can be determined in oil as well as water bearing formations.

(iii) Permeability estimates from well test:

Permeability can also be estimated from the well test in different intervals. However using this method we do not obtain a continuous estimate of permeability and the estimated permeability is representative of a very large area.

The empirical equation discussed above (equation 4.2) requires a labour intensive exercise to adjust constants or exponents. The theoretical relations such as the Kozeny Carman equation suffers from the problem that the highly complex porous medium is treated in a very simple manner and it ignores the influence of conical flow in the constrictions and expansions of flow channels. The nuclear magnetism resonance tool is very recent and still is not included in the commonly recorded log suite. Well testing also gives permeability only in selected intervals and the recovery data is from a very large zone.

Therefore the common industry practice is correlating well logs with core permeability in the cored well. A porosity permeability correlation is generated which can then be used to predict permeability at the uncored intervals and wells, providing appropriate well logs are available.

The statistical approach is a comparatively more versatile approach. It makes use of the available core permeability as a dependent variable and

develops functional relationships with the well log data, which are the independent variables. The multivariate linear regression and multivariate histogram technique are the most commonly practised to date. However, the techniques also require the assumption and satisfaction of multi-normal behaviour and linearity too in case of multivariate linear regression, and hence must be applied with caution.

Besides statistical methods neural networks (MLPs) have become very popular in well log analysis. This technique is non-linear, nonparametric and has been applied to permeability prediction from well logs. Comparative studies (Balan et al., 1995; Malki et al., 1996) have shown that MLP gave better performance than empirical equations and multiple linear regression (Rogers et al., 1995; Huang et al., 1996).

Wendt et al. (1986) reported a study of permeability from well logs using the multiple regression technique, showing that the correlation coefficient increases when other variables (i.e. logs and log-derived parameters) than porosity were included in the prediction. They found, however, disadvantages using multiple regression as a predictive method mainly on two accounts: (i) the regression will result in a distribution that is narrower than the original data set, and (ii) when prediction of the permeability extremes is a concern, the high and low values are enhanced by the weighting scheme in the regression. Rogers et al. (1995) arrived at the same conclusion when comparing regression and neural network techniques for predicting permeability from porosity. In contrast to linear regression models, neural networks do not force the prediction towards the mean value and the extreme values far outside the range of the training patterns automatically will be truncated due to the neuron activation functions.

In Huang et al. (1996) a comparison has been made between the permeability predicted by neural network, MLR technique and multiple non-linear regression (MNLR) technique. The results from MLR analysis were much poorer than the neural network. Then MNLR technique using the Levenberg - Marquardt procedure assigning a weighting function was used for predicting permeability. The results were marginally better than MLR analysis but compare poorly with the neural network results.

However one important problem in predicting permeability is its large dynamic range of  $10^{-6}$  ( $10^{-3}$  in case of reservoirs) to  $10^4$  mD. One single network may not have the proper resolution to cover this large range. This study provides first an attempt to predict permeability using a single neural network and then a CM approach by splitting the permeability range (Bhatt and Helle, 2001a).

However when core and wire line data are combined to establish the networks for quantitative prediction of petrophysical quantities such as porosity

and permeability we should keep in mind the following possible errors in data. These are discussed in detail in Helle et al. (2001).

## 4.2 Errors in data

### 4.2.1 Measurement conditions

Permeability of the core plugs is normally measured at atmospheric pressure using air, and the Klinkenberg correction is subsequently applied to convert to equivalent fluid permeability. Standard core permeability data thus represent values at the surface while logs are obtained at in situ conditions in the reservoir where the confining pressures are more than 200 bars. Compression of the rock changes the pore and the pore throat size distribution. Changes in the pores may increase the tortuosity and close some of the fluid-flow paths. At the surface the permeability of a core sample may be overestimated by a factor of two compared with its in situ value.

Enforcing the same measurement conditions for laboratory and log data requires core data at simulated reservoir conditions. The industry practice, however, is to use core data measured at ambient conditions to calibrate log data measured in situ. This practice, which is sometimes necessary for financial reasons or because of technical shortcomings, is scientifically unsatisfactory.

In particular, cores collected at large depths when brought to the surface are exposed to mechanical deformation and micro cracking that significantly increases the surface values of permeability and porosity compared with those in situ. We may also expect significant scatter in the porosity and permeability data since the mechanical impact may differ for individual rock samples due to different composition and sampling history of the core plug. Figure 4.1 show values of Klinkenberg corrected air permeability on 11 core plugs from well 30/6-9 at different confining pressures from special core analysis (SCAL) of the well. The changes with pressure are particularly strong at pressures approaching atmospheric pressure when the micro cracks tend to open. The porosity and permeability vs. pressure curves are similar for the majority of the core samples while a few are highly offset from the average curve, indicating that large scatter in the surface values of porosity and permeability may be due to the different pressure effects on the individual core plugs. While the general trend for the permeability reveals that high-permeable rocks are more prone to pressure effects than low-permeable rocks, one of the samples shown in Figure 4.1a demonstrates the opposite behavior. A local or generally valid pressure correction formula is thus not easy to establish. Thus correction for the pressure effects is a difficult problem that cannot be solved within the present industry practice where only a

small collection of core samples from a field is used for investigating the effect of overburden pressure. While porosity at low effective pressure may be overestimated by 5-15 % (as shown in Figure 3.9b), the corresponding permeability data may be in error by 20-100% depending on the rock texture and history of the individual sample.

On the other hand, by reassembling the SCAL data we find a relatively consistent relationship (Figure 4.1b) that may be used to convert usual air-permeability data to equivalent values of water permeability at depth. Since the air-to-water-permeability conversion seems to be a strong function of permeability itself, some of the scatter observed in the permeability data could obviously be removed by presenting water-permeability at reservoir pressure instead of air-permeability at atmospheric pressure. There is no obvious procedure to convert air-permeability data at atmospheric pressure to fluid permeability data at in situ conditions. From the results in Figure 4.1b we find e.g. that an air-permeability of 10 D at atmospheric pressure reduces by 40% to a water-permeability of 6 D at 200 Bar, while for a 100 mD sample the corresponding reduction only amounts to 15% (85 mD). The scatter in data, however, is too high for accepting the corresponding empirical formula for pressure corrections. Thus, to avoid introducing erroneous overburden corrections to the core data we have in the study used the raw Klinkenberg permeability supplied by the core laboratory. However, the problem is of significant practical importance and hence should be subjected to further studies.

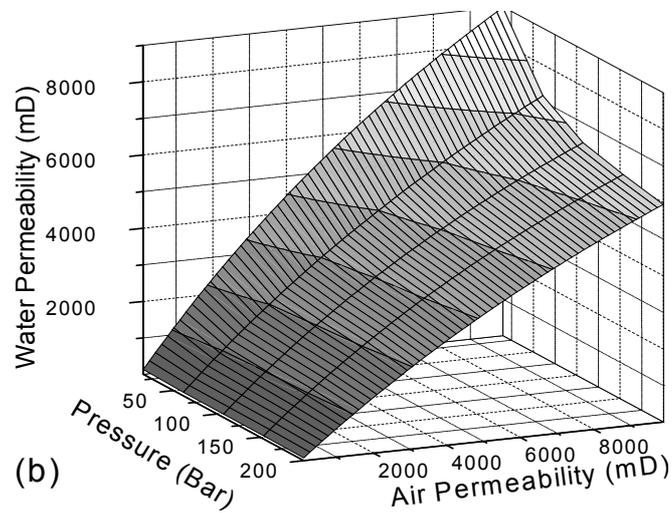
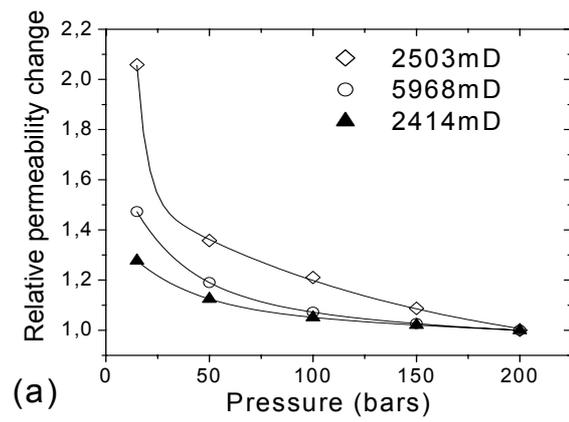


Figure 4.1: a) Relative changes in water permeability with confining pressure from SCAL study in well 30/6-9 b) water permeability for a range of confining pressure vs. Klinkenberg permeability at atmospheric pressure

## 4.2.2 Resolution and spatial sampling

Worthington (1991) provides a review of the problems encountered when comparing down hole and core measurements. As with any attempt at combining well logs and core data, shifts between recorded well log depths and sample depths are possible for a number of reasons. While every attempt is made to remove these depth shifts, undetected depth shifts could cause significant errors in porosity and particularly in the permeability predictions.

The spatial scale of the well log measurements is not equivalent to that of the rock sample measurements. Well log measurements are more spatially averaged than core data. Permeability and porosity measured from cores are representative of only a small rock mass, while a single well log reading is a composite result of petrophysical properties within a radius of cm to m depending on which tool is being used. Small-scale heterogeneity between core samples a few centimetres apart may not be resolved by well logs at all.

Due to strong heterogeneity in petrophysical properties, and the anisotropic nature of permeability in most natural rocks, it is often difficult to define a characteristic volume that is suitable for numerical calculations. We must keep in mind that a measured value from core plugs can serve as an estimate of the property over a very small interval. However there are errors in well log data also, caused by poor borehole conditions. Washout, caving, abnormal mud cake, etc. are all capable of adversely affecting well log responses.

## 4.2.3 Anisotropy

Porosity is a scalar quantity, whereas the rock permeability is a tensor owing to the directional alignment of the pore structure of natural sediments. Even in the reservoir rocks at hand we find that the ratio of in bedding to normal-bedding permeability may be one to two orders of magnitude. Since logging tools are confined to the direction of the drill bore it is expected that the log readings are affected by anisotropy to various degrees, depending on the drilling angle. But as the variations in the permeability anisotropy are confined to a much smaller scale ( $\sim 0.1$  m) than the spatial resolution of the logging tools ( $\sim 1$  m for resistivity tool, 0.3 m for density and sonic tool and 0.5m for neutron porosity tool), anisotropy variations seem to have less impact on the log readings than expected.

With all the above-discussed uncertainties in core and log data, in order to find out the optimum architecture of neural network for predicting permeability from log data we used synthetic data.

### 4.3 Synthetic data

As in the case of porosity, we established a model for the permeability as a function of formation properties and generated realistic log values for training and testing the permeability networks. The objective is an approximate model that is physically reasonable, and where the synthetic logs are functionally related to the common parameters such as porosity and clay content that characterize the formation.

In natural rocks the permeability varies by several orders of magnitude, from nano-Darcys in shaley formations to several Darcys in the best reservoir sands. Because permeability has been observed to be a strong function of the clay content (e.g. Klimentos, 1990), a model for realistic permeability profiles should explicitly contain clay in its list of variables. We follow Carcione et al. (2000) and introduce a two-component rock consisting of sand and clay particles. Permeability for a rock of mixed particle size may be expressed by equation (4.1). The effective grain size is defined by

$$\frac{1}{D} = \frac{\sum_i \frac{M_i}{D_i}}{\sum_i M_i} = \sum_i \frac{f_i}{D_i} \quad (4.4)$$

where  $D_i$  is the pore or grain dimension for the class  $i$  of the distribution,  $M_i$  is the corresponding particle mass and  $f_i$  is volume fraction of each particle size. Following Mavko and Nur (1997) we introduce the percolation porosity  $\phi_p$  and obtain the following equations for the permeability of a sand-clay mixture

$$k = B \frac{(\phi - \phi_p)^3}{T \left( \frac{1-C}{D_s} + \frac{C}{D_c} \right)^2} \quad (4.5)$$

where the clay content  $C = \frac{\phi_c}{\phi_s + \phi_c}$ ,  $D_s$  and  $D_c$  are the effective particle size of sand and clay, respectively, and  $\phi_s$  and  $\phi_c$  the corresponding volume fractions satisfying the relation

$$\phi + \phi_s + \phi_c = 1 \quad (4.6)$$

To express the effect of clay content on  $T$  we introduce the partial tortuosities  $a_{21}$  and  $a_{23}$  of Berryman (1980) for fluid flowing through the sand and clay matrix, respectively,

$$a_{21} = \frac{\phi_s}{\phi} r_{12} + 1 \text{ and } a_{23} = \frac{\phi_c}{\phi} r_{23} + 1 \quad (4.7)$$

where  $r_{12}$  and  $r_{23}$  depend on the geometry of the boundaries separating the sand grains from clay and clay from fluid, respectively, normally with values in the range  $0 < r_{ij} < 1$  but here set to  $r_{ij} = 1/2$  for spherical grains, giving the following expressions for tortuosity and permeability of the sand-clay mixture

$$\frac{1}{T} = \left( \frac{\frac{\phi_s}{2\phi} + 1}{\frac{\phi_s}{2\phi} + 1} + \frac{\frac{\phi_c}{2\phi} + 1}{\frac{\phi_c}{2\phi} + 1} \right) \frac{1}{(1-\phi)} \quad (4.8)$$

By fixing the parameters  $\phi_p = 0.035$ ,  $D_s = 100 \mu\text{m}$ ,  $D_c = 2 \mu\text{m}$  (see e.g. Mavko et al., 1997, p. 261-263) and  $B = 50$  (which is a fair mean of the values for  $B$  determined to be in the range 5-100 by Rose and Bruce, 1949) we can now model permeability from (equation 4.5) as a function of porosity and clay content. Then we follow a similar procedure as for porosity, but substitute the resistivity by the neutron porosity log based on practical experience with real data and, moreover, we add the gamma-ray log which is the common indicator for lithology and clay content. The recipe applied is then the following:

- i) From the density-porosity equation (3.3) we establish a porosity profile from models of  $\rho_f$ ,  $\rho_m$  and  $\rho$  ( as shown in Figure 4.2). For the fluid saturation and  $\rho_f$  we use the same model as for porosity (as shown in Figure 3.1), but introduce larger variations in the model for bulk density  $\rho$  in order to obtain the appropriate range of permeability values (equation 4.5) by a corresponding expansion of the porosity range.
- ii) A model for the clay content  $C$  is then designed. The introduction of clay reduces the lower limit of permeability of the model and thus expands the effective range of variation to cover the entire reservoir range from micro-Darcy to a few Darcy. As shown in the following steps the variations in  $C$  with depth, are not only reflected in permeability but also in the gamma log, the sonic log and the neutron porosity log.
- iii) The gamma log is then produced from the clay model by using the empirical equation of Larionov (1969)

$$C = (2^{2\Delta\gamma} - 1)/3 \quad (4.9)$$

where  $\Delta\gamma = (\gamma - \gamma_{Min}) / (\gamma_{Max} - \gamma_{Min})$  and where  $\gamma$  is the actual gamma log reading,  $\gamma_{Min} = 20$  API is the reading for pure sand and  $\gamma_{Max} = 139$  API is the reading for pure clay.

- iv) From the  $C$  model we also establish the sonic transit time for the rock mixture by the equation  $\Delta t_m = C\Delta t_c + (1 - C)\Delta t_s$ , with  $\Delta t_s = 55 \mu\text{s}/\text{ft}$  and  $\Delta t_c = 75 \mu\text{s}/\text{ft}$  (Brigaud et al., 1992, Table 2)
- v) Given the porosity model, a model for  $\Delta t_m$ , the fluid saturation model provides  $\Delta t_f$  from (equation 3.5) and we compute the sonic log  $\Delta t$  from Wyllie's equation (3.4). The input synthetic logs for the model are shown in Figure 4.2.
- vi) From the partial fluid saturation model (Figure 3.1), the  $\phi$  and  $C$  profiles we now obtain the neutron porosity log by the relation

$$\phi_N = \phi(S_w H_w + S_o H_o + S_g H_g) + (1 - \phi)[C H_C + (1 - C) H_S] \quad (4.10)$$

where  $H_i$  is the hydrogen index of the constituent  $i$ . For the pore fluids the hydrogen indices are given by  $H_w = 1.1\rho_w$ ,  $H_o = 1.1\rho_o$  and  $H_g = 1.2\rho_g$ , and for the rock material we use  $H_C = 0.05$ ,  $H_S = 0.001$  (Schlumberger, 1989).

The resulting data set are four synthetic input logs; density, gamma, sonic and neutron porosity ( $\rho$ ,  $\gamma$ ,  $\Delta t$  and  $\phi_N$ ) and the output synthetic permeability data ( $K$ ), each consisting of 3000 samples. Duplicates were made by adding independent random noise of 2.5, 5 and 10%. Subsets consisting of 75-350 samples were then selected at regular depth intervals to use for training patterns. In Figure 4.2(b) is shown the data set with 10 % independent random noise added to each log.

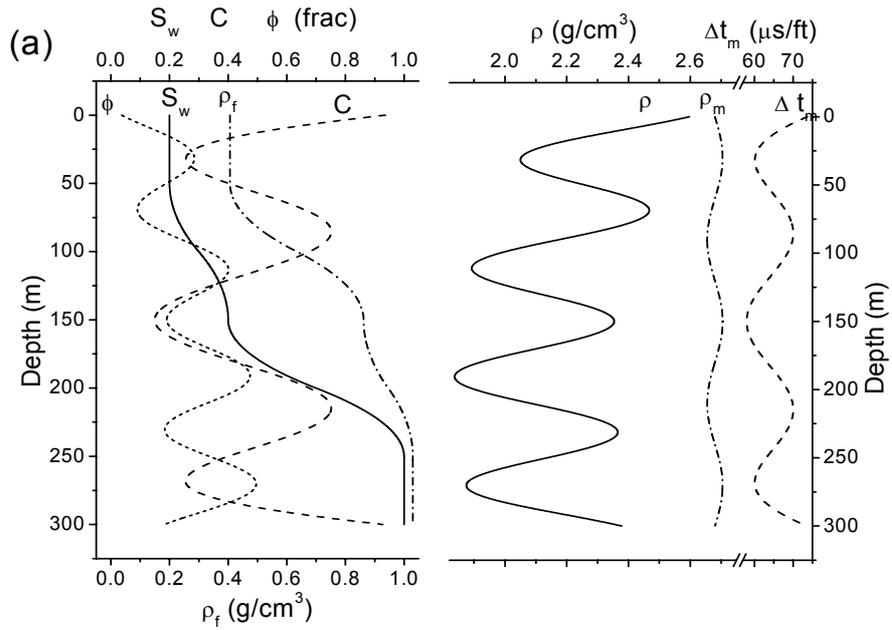


Figure 4.2a): Model inputs for permeability network

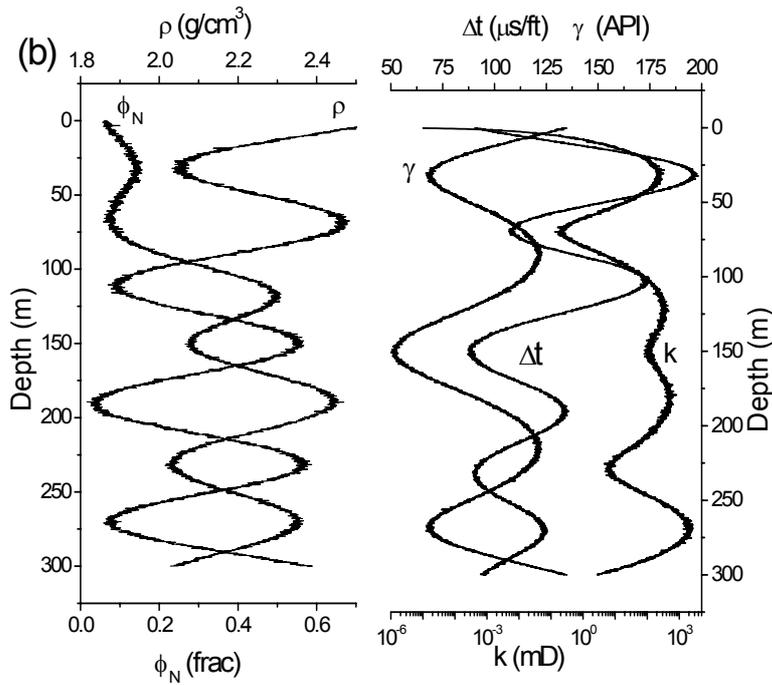


Figure 4.2b): Synthetic logs for permeability network with 10 % independent random noise added

## 4.4 Optimal design of the network

In order to find out the optimal architecture of the network in terms of number of hidden neurons and number of samples, we did a similar exercise as for the porosity network. The selection of the input logs is based on the criteria which are commonly available in the suite of logs and which are sensitive to lithology and porosity. The gamma ray log is highly sensitive to lithology type and density; neutron porosity and sonic logs are more sensitive to the porosity and to fluid type. Thus given four input neurons and a single output neuron i.e. permeability the appropriate number of hidden units remains to be determined by the same experimental procedure as for the porosity network. By varying the number of hidden units and the number of training samples (as shown in Figure 4.3 and 4.4) the errors follow essentially the same trends as obtained for the porosity neural network (as shown in Figures 3.3-3.4). The standard deviation has a more consistent trend showing that for low noise level data i.e. for noise free and 2.5% noisy data there is no increase in noise level, with the increasing number of neurons the complexity becomes a concern at a higher noise level. From Figure 4.3a it is clear that for 5% and 10% noise levels an increase in the number of neurons beyond 10 increases the noise level. No consistent trend could be found in the bias. Thus with moderate noise the minimum error is confined to a number of hidden units around 10, the same as for the porosity network. The error versus number of training samples, on the other hand, still reveals a strong decline at 150 samples and continues beyond 300 samples as shown in Figure 4.4a. A training set twice the size of that for the porosity network can thus be justified in order to reduce the error of the permeability network well below the “natural scatter” in the permeability data.

Thus in the experiments with synthetic data we use 300 patterns. However, because of the practical limitation in the access to a sufficient number of well-represented training samples, the network for real data is based on only 260 samples. Thus the optimal architecture of the network is 260 samples and 10 hidden neurons.

Using synthetic dataset and the optimal architecture of the network we made a comparison between the performance of a single neural network and MLR technique, the results are shown in Figure 4.5. It is evident from the figure that due to no linearity in permeability relation the performance of even one single network is much better than that of MLR technique. The maximum error in the MLR permeability is about 0.5 in terms of logarithmic permeability, whereas the model and the neural network permeability overlap each other.

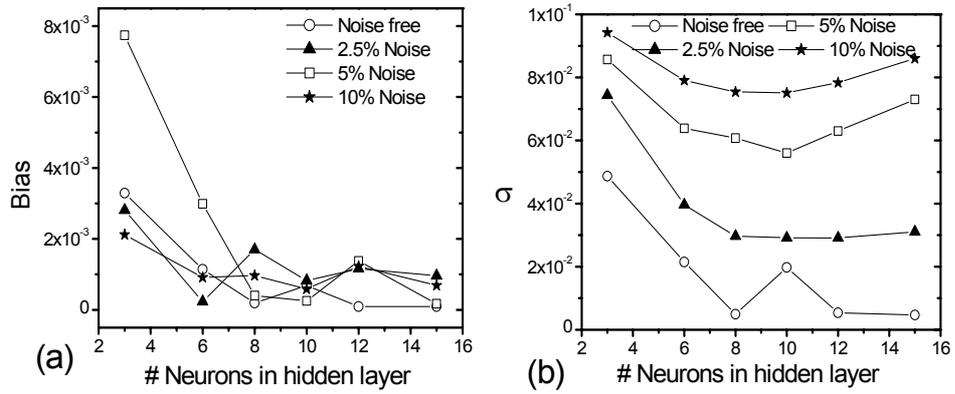


Figure 4.3: Variation of bias and standard deviation with number of neurons

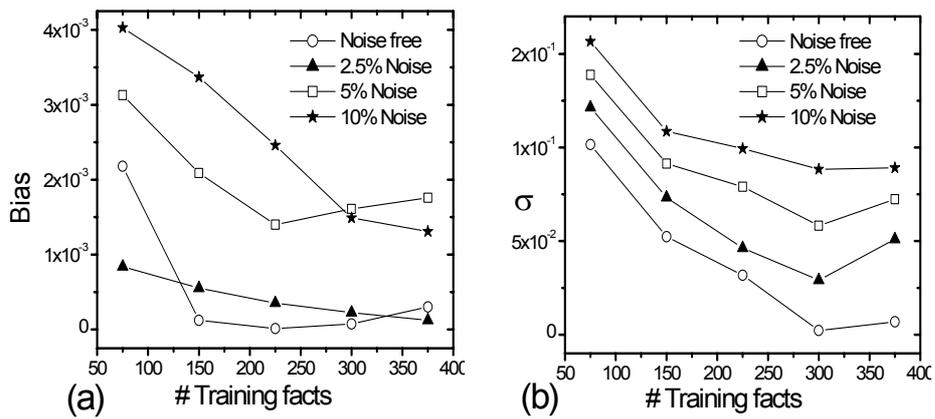


Figure 4.4: Variation of bias and standard deviation with number of training facts

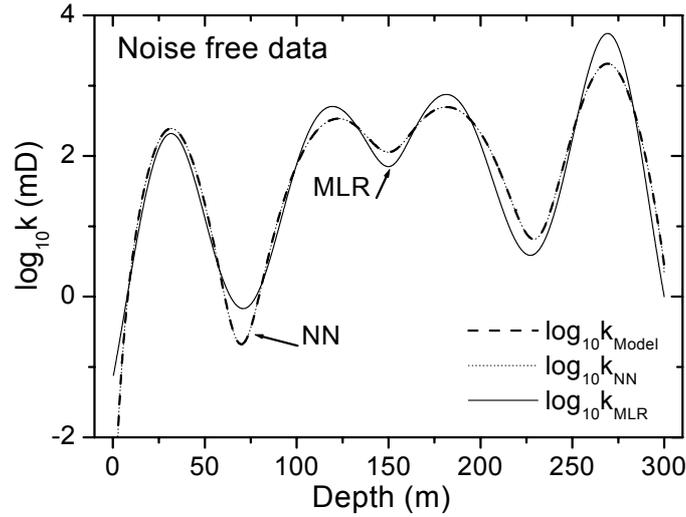


Figure 4.5: Comparison of MLR and neural network prediction for noise-free data

A traditional method of correlating permeability data is the  $\log(k)$  vs. porosity diagram to search for a relationship of the form  $k = C\phi^x / S_{wi}^y$  (Schlumberger, 1989), where the exponents and constant are to be determined by fitting with data. Also since permeability in natural rocks varies over a wide range ( $10^{-6} - 10^4$  mD) the logarithmic representation of permeability is more practical in the context of neural networks since the data usually will be normalized and mapped into an interval  $[-1 \ 1]$ . A major concern, however, is the loss of resolution implied by fitting the entire permeability range into a single narrow interval. One of the objectives for a neural net architecture for permeability proposed by Zhang et al. (2000) was to compensate for this deficiency. They obtained range splitting by connecting two single neural networks in series; the first neural network predicting a sub-range indicator (six sub-ranges), which, in turn, provides an additional input to the second network predicting the final permeability values.

Whereas Zhang et al. (2000) obtained range splitting by connecting neural networks in series we exploit a parallel architecture for the same purpose using a hybrid of ensemble combination and modular systems. In this study we predicted permeability from well logs by two methods i.e. using a single neural network and then to enhance the resolution we used a modular neural network (MNN). The building blocks for MNN are single neural networks as shown in Figure 2.4. The single neural network was trained on the full range of permeability varying from  $10^{-6}$  to  $10^4$  mD where as in the case of MNN we split it into three ranges. In

the MNN architecture the training dataset was split into three permeability ranges and then we exploited the ensemble combination as well as the modular combination; i.e. each module is dedicated to a restricted permeability range (task) and the modules in turn are combined to cover the entire range. Moreover, the module for a given range is itself an ensemble combination where a number of neural networks are redundantly combined. The underlying recipe is the following:

- i) The  $N$  training patterns are sub-divided into  $K$  sub-sets of permeability facts with over-lapping boundaries, implying that patterns are duplicated within the over-lapping zones, such that  $N < \sum_{i=1}^K N_i$  where  $N_i$  is the number of patterns in  $i^{th}$  subset.
- ii) The  $K$  modules are trained for each set of  $N_i$  over-lapping patterns.
- iii) Each module consisting of  $m$  neural networks connected in parallel (as shown in Figure 3.11), trained on the same pattern subset, and redundantly combined over the random initial weights.
- iv) A gating network has been designed to dictate output from the appropriate expert. It shares the input with the experts and has identical architecture to that of the expert networks, but is trained for the entire permeability range using all  $N$  patterns. Thus, the low-resolution gating network determines the actual range, and then triggers the corresponding high-resolution expert and the outputs are subsequently combined. So in the combiner the appropriate value of the permeability is assigned to each depth sample.

The architecture of the permeability CM is shown in Figure 4.6. In the example shown in Figure 4.7 we use three sub-ranges ( $K=3$ ) and with each expert consists of five neural networks ( $m=5$ ). The figure shows the model permeability and the permeability predicted by the three experts in different ranges. There is a very good agreement between the two. Because of the non-linear ( $\log_{10}$ ) representation of the output permeability we are faced with several problems in using the OLC as shown in Figure 4.8. This will be analyzed as follows:

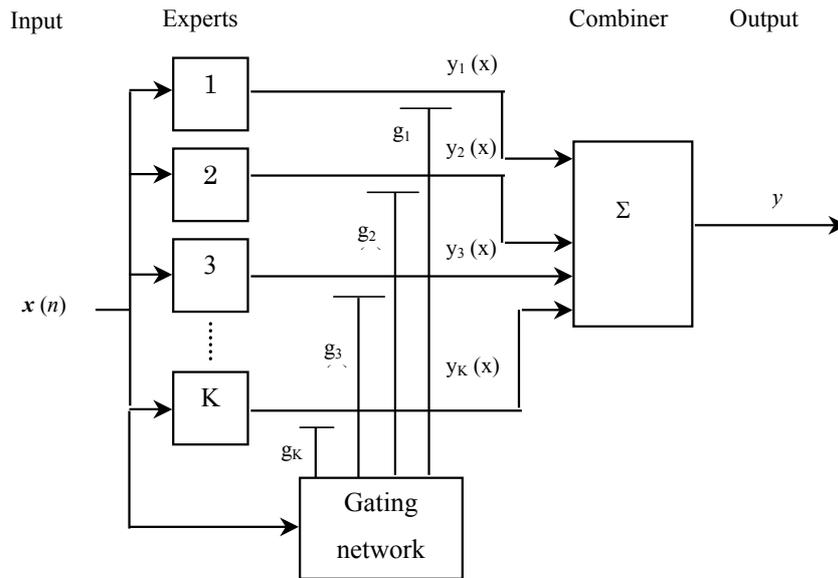


Figure 4.6: Block diagram of a CM for permeability prediction with range splitting

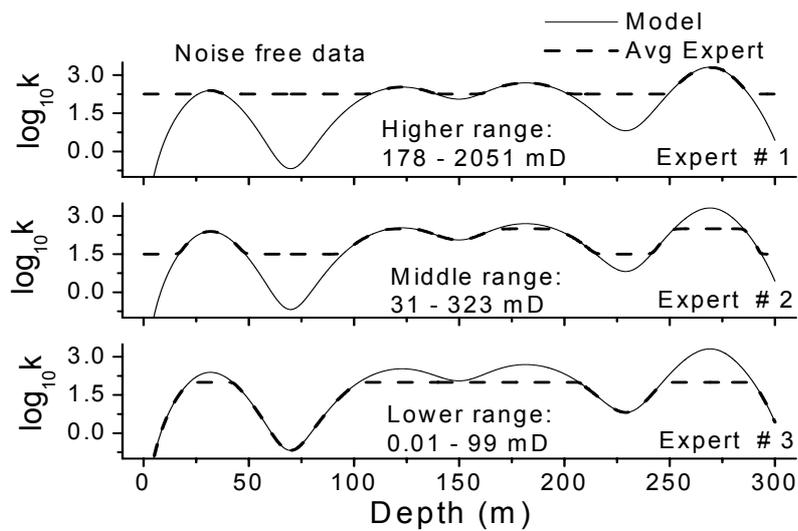


Figure 4.7: Example showing permeability range splitting with overlapping ranges

## 4.5 Error analysis of the committee networks

In order to form a CM of networks we have compared here the different approaches for training and the combination of networks. The networks have been trained by following the over training and the cross-validation approach and combined by using simple averaging, unconstrained and the constrained OLC approaches.

Let  $k_i$ ,  $i=1, m$  be the permeability values obtained from the  $m$  networks trained on the same patterns but with different initial conditions (randomized weights). Let  $k$  be the true permeability and  $n_i$ ,  $i=1, m$  be the error associated with each neural network in the CM. For a simple ensemble average of the CM output we may write

$$\overline{\log k} = \frac{1}{m} \sum_{i=1}^m (\log k_i + n_i) = \log \tilde{k} + n_{AV}, \quad (4.11)$$

where it is assumed that outputs from each network are similar i.e.

$$k_1 \approx k_2 \dots \dots \dots \approx k_m \approx \tilde{k} \quad (4.12)$$

For small values of  $m$  ( $m = 5-10$  in the experiments)

$$n_{AV} = \frac{1}{m} \sum_{i=1}^m n_i \neq 0 \quad (4.13)$$

implying that the error term  $n_{AV}$  contains a finite bias plus random noise. However, for large values of  $m$

$$n_{AV} \rightarrow 0 \quad (4.14)$$

For the unconstrained OLC with a constant term (to correct for the bias as in the porosity CM) we may write

$$\overline{\log k} = \sum_{i=0}^m \alpha_i (\log k_i + n_i) = \alpha_0 + \log \tilde{k} \sum_{i=1}^m \alpha_i + n_{OLC} \quad (4.15)$$

where the error term

$$n_{OLC} = \sum_{i=0}^m \alpha_i n_i \leq \sum_{i=0}^m \alpha_i \sum_{i=1}^m n_i = m n_{AV} \sum_{i=0}^m \alpha_i \quad (4.16)$$

implying that  $n_{OLC}$  has an upper bound that increases proportionately to  $m$  and thus

$$n_{OLC} > n_{AV} \quad (4.17)$$

In case of constrained OLC i.e.  $\sum_{i=1}^m \alpha_i = 1$  and with  $\alpha_0 = 0$  equation 4.15 becomes

$$\overline{\log k} = \log \tilde{k} + n_{OLC} \quad (4.18)$$

and still  $n_{OLC} > n_{AV}$  (from equation 4.17)

Thus we see that for combining the permeability networks the constrained OLC approach (Hashem, 1997) is better than the unconstrained approach but still the simple averaging (equation 4.11) is superior to the OLC as also shown by the numerical experiment in Figure 4.8 which shows the constrained OLC (equation 4.18) where the blow-up of noise in the case of OLC is evident. Since the OLC approach does not work properly in the case of  $\log_{10} k$ , and since the variance has become a serious problem, the idea of over-training to reduce bias on account of increasing variance is no longer applicable. The concern is now to reduce the variance of the individual neural networks in the CM and still keep the bias at a sufficiently low level. Instead of over-training the neural networks we validate the network output against the validation set. When the variance reaches its minimum the training stops. In Figure 4.8a we have compared the validation method with the over-training method for the permeability CM, using both the OLC (equation 4.18) and the simple average (equation 4.11). The results clearly reveal that the simple ensemble averaging, using the validation criteria for training the individual neural networks, is the optimum approach for the permeability networks at hand.

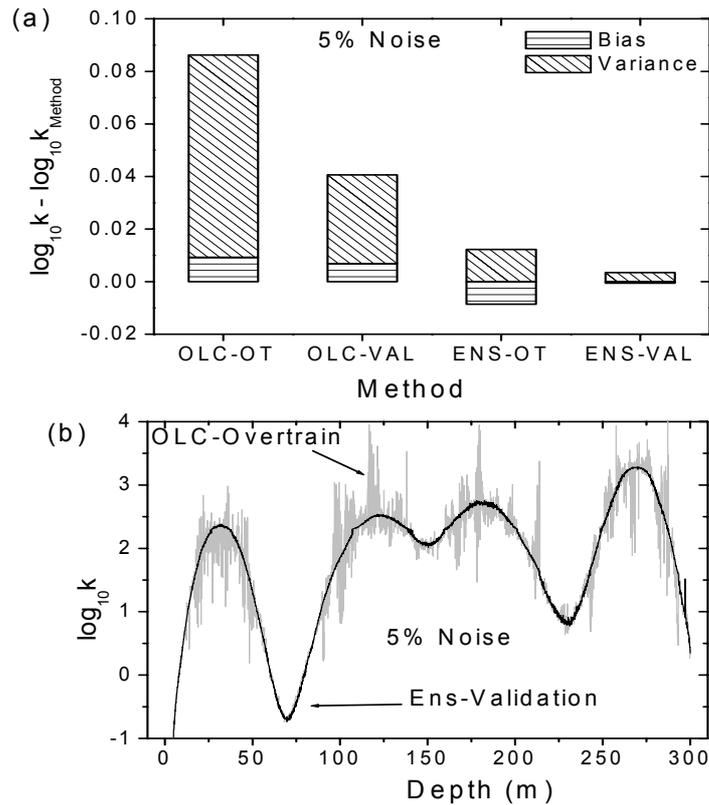


Figure 4.8: Comparison of bias and variance (a) for alternative training and combination methods (b) predictions based on OLC-overtraining method

## 4.6 Real data

In the first study (Helle et al., 2001) we trained a single neural network having 4 input neurons (density, gamma ray, neutron porosity and sonic), 12 neurons in the hidden layer and one neuron (permeability) in the output layer. The training patterns have been taken from three fields in the North Sea. Most of the training patterns (72%) are from water bearing well (31/4-10) from the Brage field, the rest of the patterns (25%) are from the Oseberg field and from the Halten banken area (3%). The patterns from the Oseberg field are added from the oil and gas bearing well (30/9-B-20) and the water bearing well 30/9-11, therefore the network has a complete range of reservoir fluids. It is very important that the network should never be trained on the unresolved dataset. The shale permeabilities were added from the Haltenbanken area from the study of Krooss et al. (1998). This low permeability shale data was added to tune the network for basin scale applications (Bhatt, 1998). By adding the six low-

permeability shale points in the range 0.5-39  $nD$  to the standard core analysis permeability in the range 34  $\mu D$  - 12  $D$  we have covered most sediments within the prospective depths in the Viking Graben. Most of the training facts are conventional Klinkenberg corrected permeability measurements on core plugs. While the porosity network is based on samples both from Tertiary and Jurassic, all training facts for the permeability network are confined to cored sections from upper Jurassic. The training patterns are dominated by wells outside the test field i.e. the Oseberg-field, and the majority of facts are from Brage-field which is in the same area. The trained network has been tested on several wells from the Oseberg field. The results of this network for permeability predictions in the cored reservoir intervals of two wells are shown in Figures 4.9 and 4.10.

Well 30/6-4 is completely unknown to the network. It is an oil bearing well with a hole deviation of 0-1 degrees. The core data is available in the Rannoch and the Oseberg formation. The permeability prediction by neural network matches very well with the Klinkenberg corrected core permeability in most of the intervals as shown in Figure 4.9b and c. The error distribution is shown in the form of the histogram of the difference between the logarithm of core permeability and that of the neural network predicted permeability. The error distribution fits the Gaussian model so the mean values and the standard deviation shown are for the Gaussian model. In Figure 4.9b the mean error is very close to zero. The reason for the standard deviation of 0.28 is the fine layering in the bottom part of the Oseberg formation (2668-85m). Due to the small scale heterogeneity there is scattering in the core data whereas the neural network prediction of permeability gives about a mean value in this interval. There are two reasons for this mismatch, firstly the spatial resolution of log data is poorer than that of core data and secondly less resolution of the network, which is trained on the whole, range of permeability. As a result the network does not have a high resolution because the transfer function normalises the whole dataset in the range of -1 to +1. The sonic log has the best resolution i.e. about 0.3-0.6m and then the density but for the rest of the logs the resolution is about 0.5m to about 1m. Here also in the sonic and density logs a marked increase in the amplitude of short-length variations coincides with intervals where core data exhibit maximum scattering. So there is not much we can do to improve the log resolution except that we should include the high resolution logs available as inputs. The second problem is remedied to some extent by either reducing the permeability range or using a range splitting permeability CM architecture as shown in Figure 4.6.

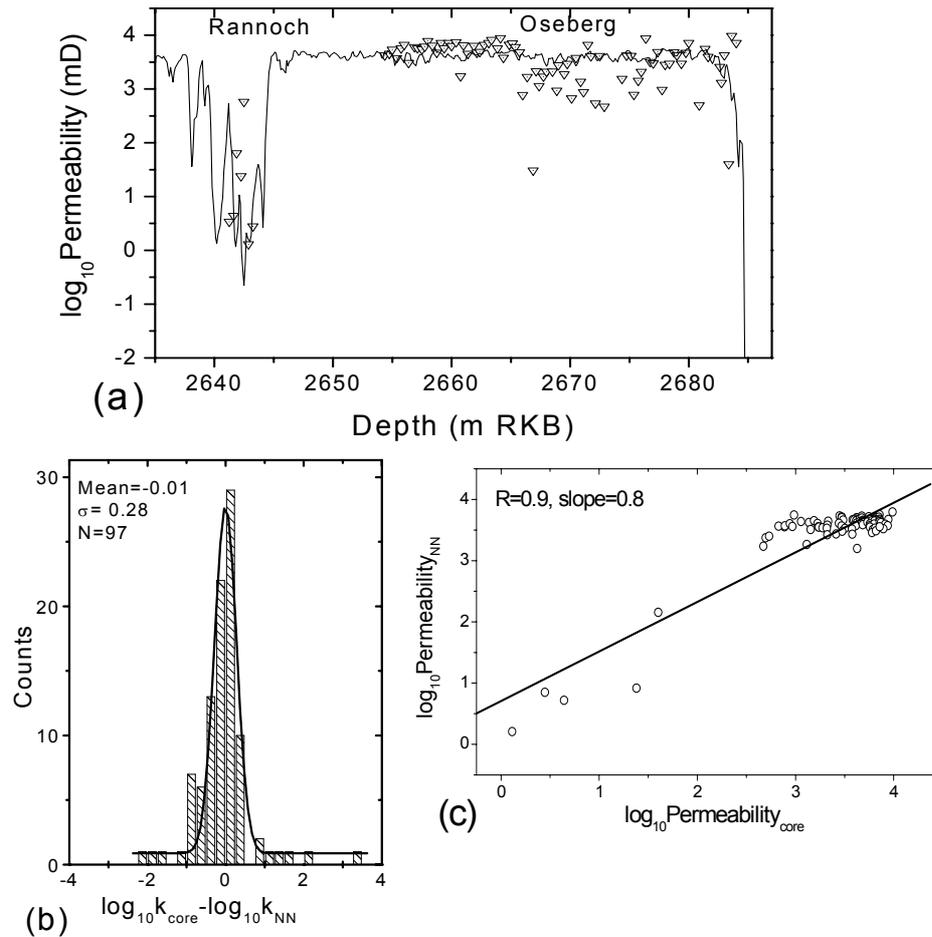


Figure 4.9: (a) Comparison of permeability predictions with core data in well 30/6-4, (b) error distribution

Figure 4.10 shows the comparison of Klinkenberg corrected core permeability with the permeability predicted by neural network in well 30/9-B-20. In the training dataset we took 10 training points from this well but the rest of the data is unknown to the network. The well has an inclination angle of 38-42 degrees. The top formation is the Tarbert (3110-3138m.), then Ness (3138-3158m.), then Etive (3158-3162m), then Rannoch (3162-3167m.) and finally the Oseberg formation (3167-3220m.). The Tarbert formation, which is fairly homogeneous, shows a fairly good match between the two permeabilities. There is however, an overprediction of permeability in the top part of Tarbert formation. The scattering is more in the heterogeneous Ness formation due to thin bed heterogeneities. There is a good match in the Etive, Rannoch and in top

part of the Oseberg formation (3164-3195m.). There is an overprediction of permeability by neural network in bottom part of Oseberg formations (3195-3220m.) due to fine layerings so the log gives a mean value of the permeability. In Figure 4.10b the mean error is 0.15 logarithmic permeability and a standard deviation of 0.48 logarithmic permeability. However the reason for the high standard deviation is mainly due to the large scattering in the core data and the spatial resolution of the logging tools as discussed. Figure 4.10c shows the correlation between the neural network predicted permeability and the core permeability. The reason for the poor correlation coefficient is mainly due to the discrepancy in core and neural network predicted permeability towards low permeability end. This is because in the training dataset we gave 6 points of low permeability i.e. nano darcy from Krooss et al. (1998). In general in laboratory so low permeabilities cannot be estimated but in real formations it is not unnatural to have low permeability cemented and carbonate streaks. Thus the network is predicting very low permeabilities in some streaks in Ness formation where as the core data doesn't.

Figure 4.11 shows the core photograph of the Etive and the Rannoch formations. The core photograph shows that the Etive is clean sand whereas the Rannoch is a silty sand. The Figure 4.12 displays the small-scale heterogeneities, which are beyond the resolution of logs.

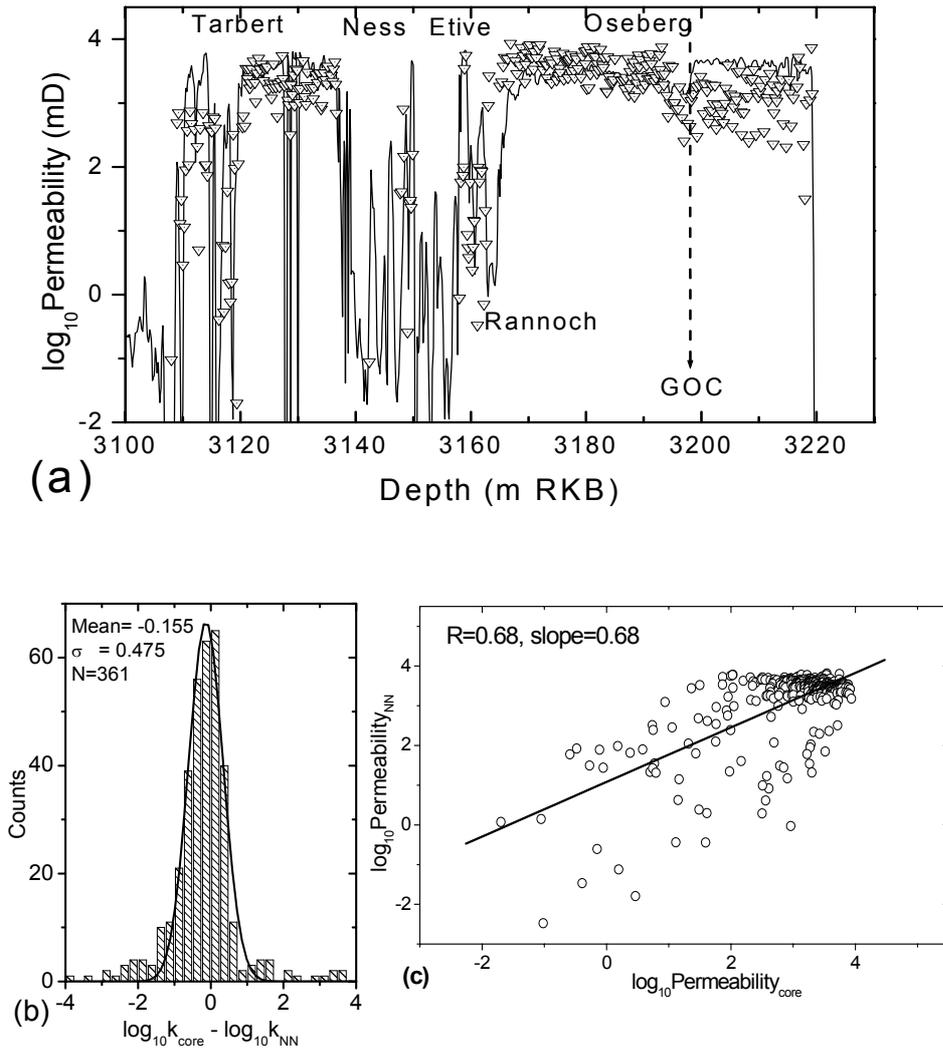


Figure 4.10: (a) Comparison of permeability predictions with core data in well 30/9-B-20, (b) error distribution (c) crossplott of core against neural network permeability.

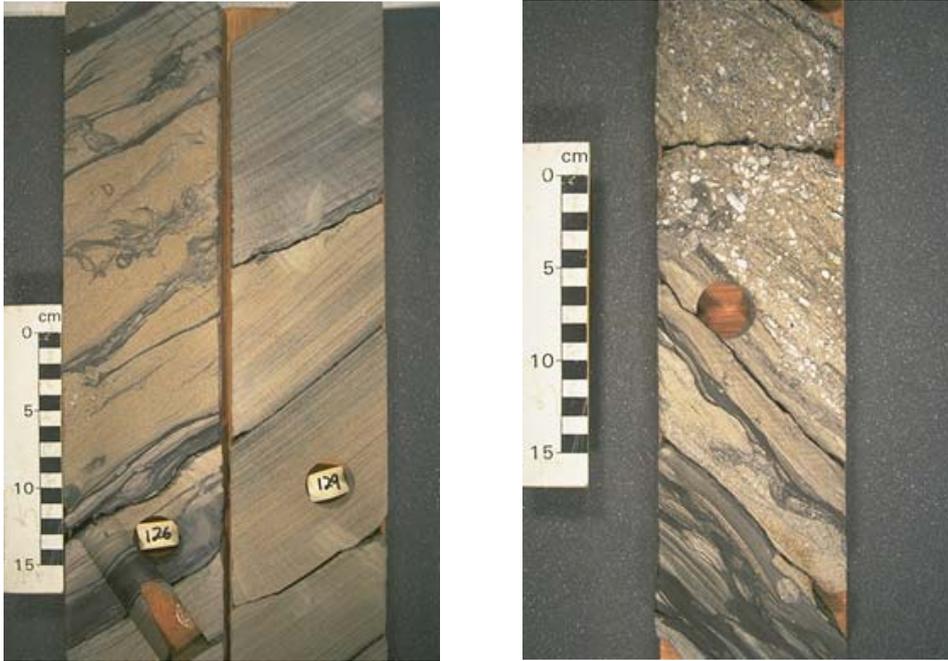


Figure 4.11: (a) The Etive formation well 30/9-B-3 2781-2782 m (b) Transition of the Etive into the Rannoch formation well 30/9-B-50 H, 3238.5-3239.0 m

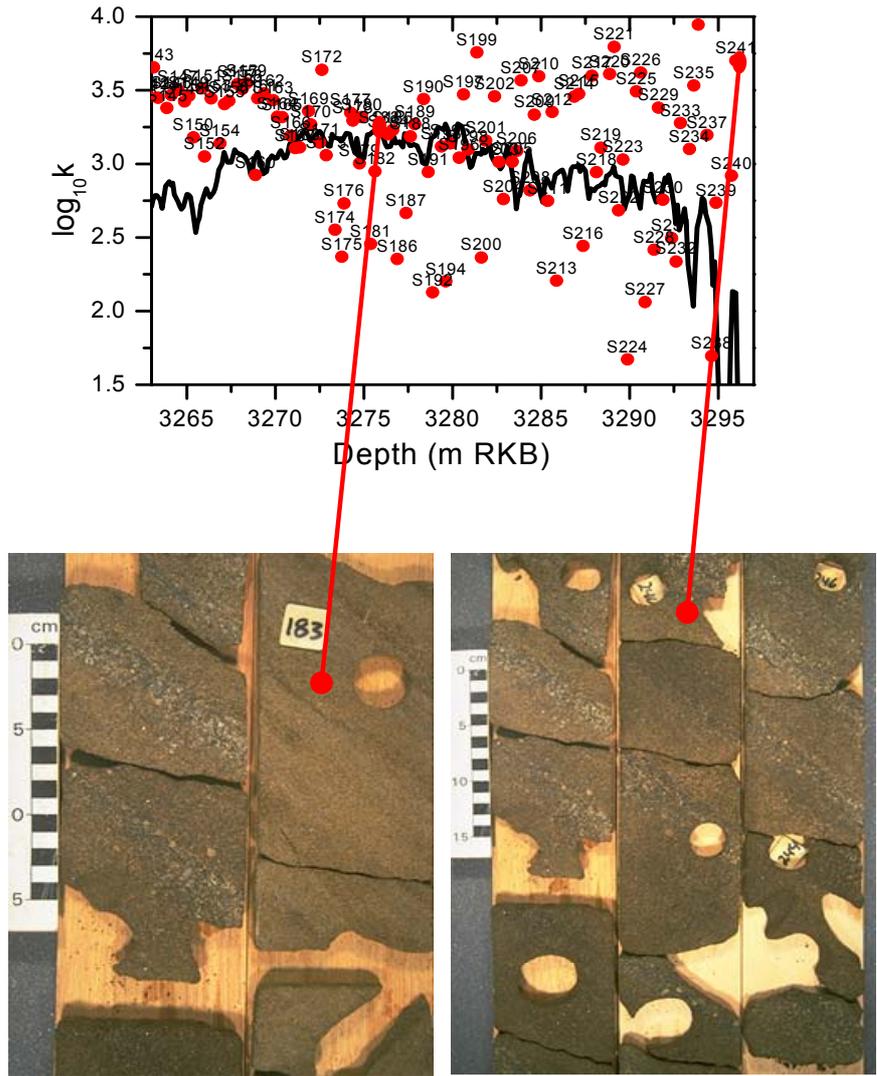


Figure 4.12: Comparison of log and core derived permeability values with the core photograph.

Motivated by the improvement in results by using CM instead of single network for predicting porosity we also applied the CM for predicting permeability. The architecture for permeability CM using real data is same as we used with the synthetic data (Figure 4.6).

Provided that the precautions discussed with the architecture of the network have been taken into account during training and combining networks, the permeability CM is more capable of handling the non-linearity and noise than

MLR and a single network as shown in Figures 4.5 and 4.13. As discussed in Bhatt and Helle (2001a) range splitting by the CM helps to resolve details in the combination of log that otherwise are invisible. The latter is demonstrated in Figure 4.13a where we show details in the CM predictions, which are not captured by a single neural network. This example is taken from a water bearing vertical well 30/9-1. The encircled portions show the improvement by CM architecture. The underprediction and poor resolution of permeabilities by single neural network which are due to the large dynamic range of the training dataset are, improved by the range splitting in the CM (encircled portions). As shown in Figure 4.13b there is a reduction in bias and variance also along with the improvement in the resolution of the networks by using CM architecture. The same has been illustrated in Figure 4.13c.

The next example shown in Figure 4.14 is taken from well 30/9-B-20. Compared with previous predictions based on single neural network (as shown in Figure 4.10) more details of the core data are now reproduced. Due to the increased resolution of the network because of range splitting the network is able to predict permeabilities in the bottom part of the Oseberg formation (3195-3220), which also has fine layering, and in the top part of the Tarbert formation. The encircled portion shows the improvement. The overall error between predictions and core measurements has been also significantly reduced. As shown in Figure 4.14a the mean error has been reduced from 0.155 to 0.04 and standard deviation from 0.475 to 0.3. There is also higher correlation between the core permeability and CM permeability (Figure 4.14c shows the improved results). There still remains the discrepancy between core and CM predicted permeabilities in the Ness formation because of the higher heterogeneity in thin layers.

In most of the results discussed yet on real data showed a standard deviation of about 0.3 logarithmic permeability which is quite low compared with standard industry practice (multivariate linear regression technique) keeping in mind all the errors between core and log data. In Huang et al. (1997) also the predicted permeability by neural network had an average error of less than  $\pm 0.5$  logarithmic permeability.

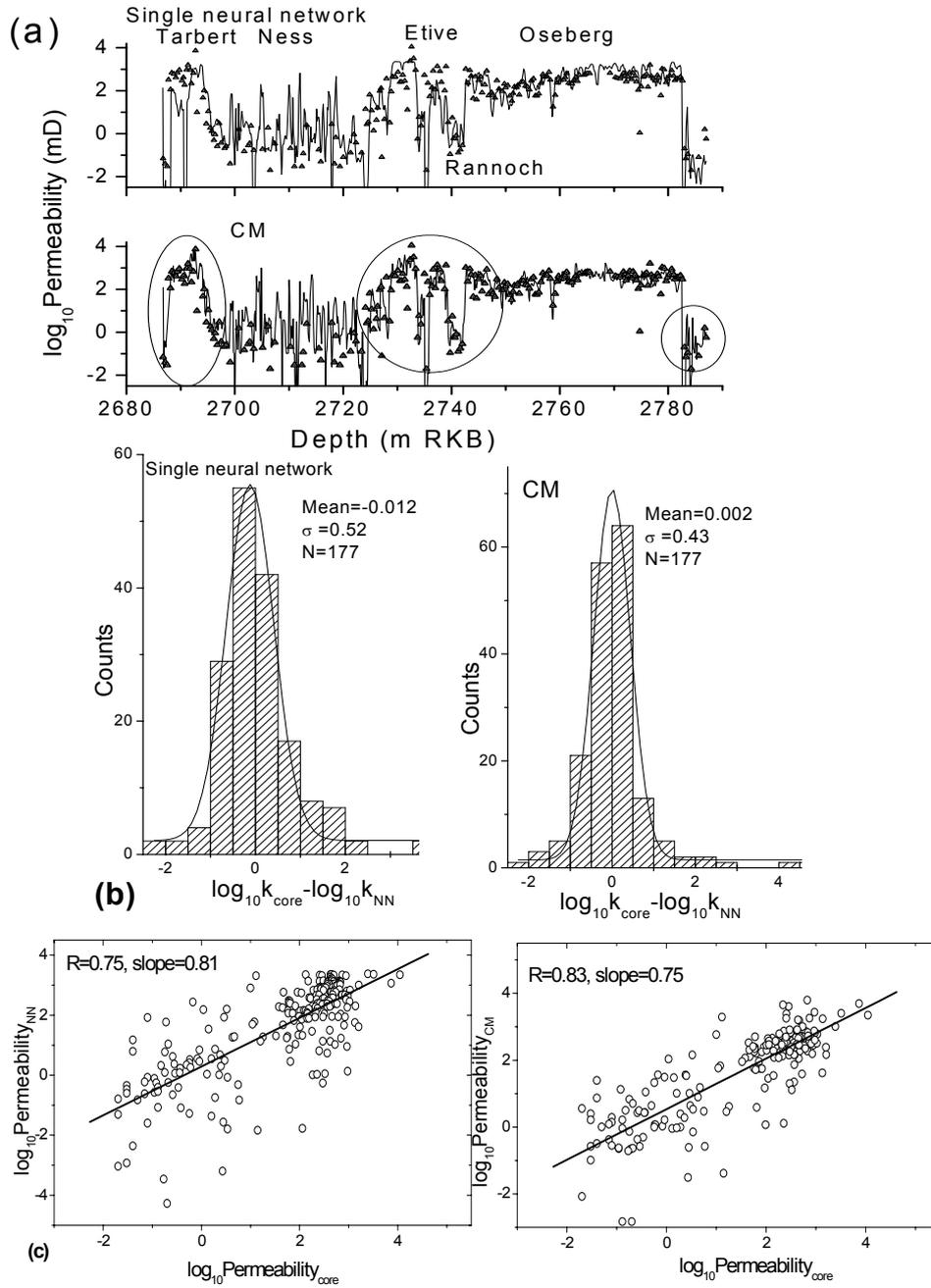


Figure 4.13: (a) Comparison of permeability predicted by neural network and a CM in well 30/9-1, (b) error distributions (c) crossplot of core against CM permeability.

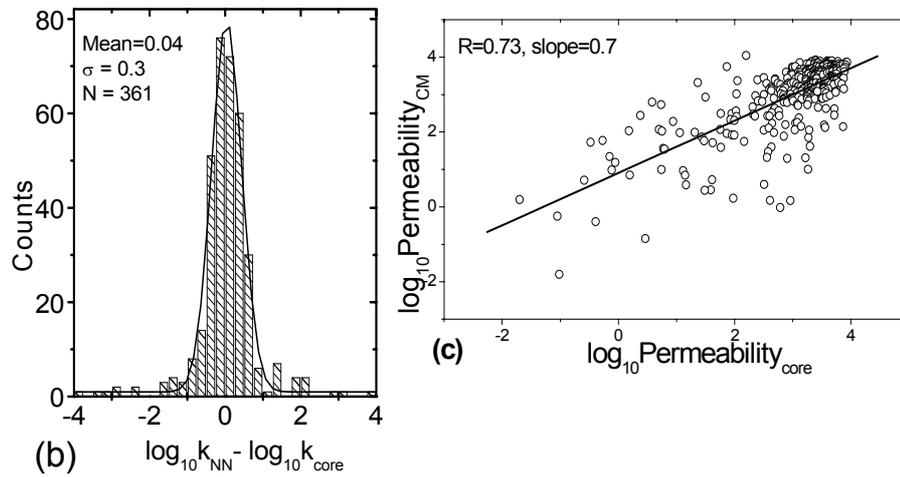
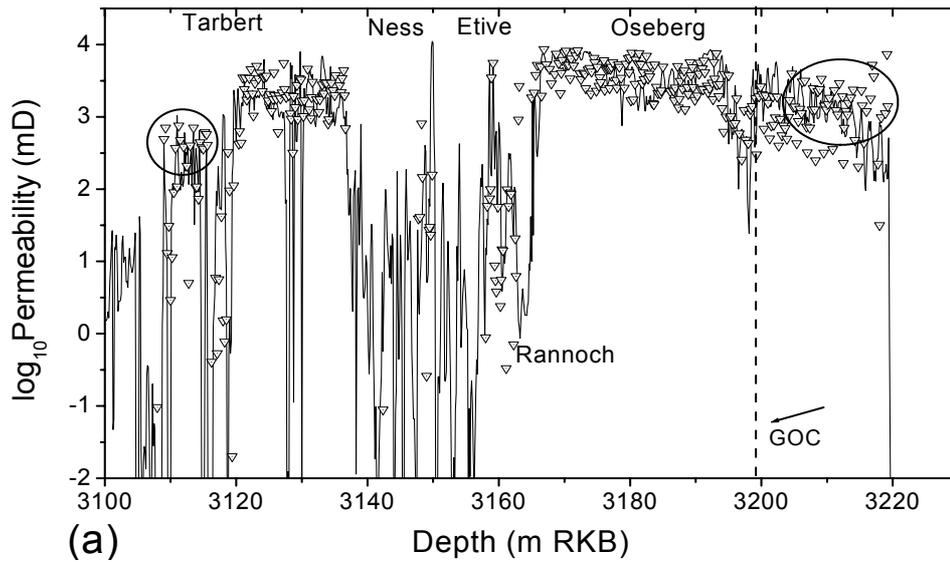


Figure 4.14: (a) Comparison of permeability predicted by a CM in well 30/9-B-20 with core permeability (b) error distribution (c) crossplot of core against CM permeability.

## 4.7 Conclusions

The neural network approach to permeability prediction is much more advantageous than any conventional method, which includes empirical formulas based on linear regression models or the common semi-empirical formulas such as the Wyllie and Rose model or the Kozeny-Carman equation because knowledge of a mathematical model is not necessary.

The benefits of modularity by decomposing the permeability range into a number of sub-ranges increases the resolution in comparison with a single network trained on the whole range of permeability. The increase in resolution could also be achieved by reducing the dynamic range of the training dataset. But in this study we desired to predict permeabilities on the basin scale so we kept a large dynamic range of the training dataset.

Synthetic data based on a background model of Kozeny-Carman type with porosity and clay content as the independent variables helped in evaluating the optimal architecture of the network, training procedure and the size of training dataset. With the four inputs; i.e. sonic, density, gamma, neutron porosity, the optimal number of hidden units of the permeability neural network is confined to the range 8-12 where the variance and bias are at their minima. In general, the errors steadily decrease with the number of training facts. A practical lower limit has been set to 300, or twice the size of the training set required for the porosity network due to the increased complexity of the background relationships with the log readings.

Since we use a logarithmic permeability scale rather than a linear scale, the success of OLC in the porosity CM is not repeated when it is applied to the permeability CM. In fact noise amplification takes place. Simple ensemble averaging is shown to be the preferred method of combining the outputs. However, with a relatively small number of neural network components in the CM, the variance associated with the individual networks becomes a major problem. The normal success of over-training to reduce bias is replaced by errors due to increasing variance. A different training strategy must be applied using the validation approach, which requires the training to stop when the level of minimum variance has been reached.

Provided that precautions are taken, the permeability CM is more capable of handling the non-linearity and noise than MLR and a single neural network. In application to real data a minimum standard error of the difference between prediction and Klinkenberg corrected permeability data seems to be around 0.3 in logarithmic units (of mD), mainly due to limitations in the techniques of measurement.

Thus our permeability predictions are sufficiently accurate for most practical purposes, given the limitations due to the spatial resolution of the logging instruments, depth shifting between core and logs and the expanded range covered by the permeability values. Application to real-time data (MWD) is the obvious extension of this technique.

# Chapter 5

## Fluid saturation prediction

### 5.1 Introduction

Apart from porosity, permeability of the reservoir rock and the type of fluid it is important to know the hydrocarbon saturation of the reservoir in order to estimate the total reserves and to determine if the accumulation is commercial. The saturation of a formation is defined as the fraction of its pore volume occupied by the fluid considered. The direct sampling of the reservoir fluid is not technically and commercially efficient so the preferred method to date is to use well logs for fluid saturation prediction. Moreover the logs provide a continuous record of the formation also.

Although the common saturation models such as those of Archie (1942) and Poupon (1971) are based on sound scientific and technical reasoning, they are still non-universal and non-linear empirical relations that need to be fitted to real data. These are the main justifications for employing the neural network techniques in predicting fluid saturation. The neural network approach is very pragmatic and non-linear, and may even be trained to display the expertise of a skilled petrophysicist. Helle et al. (2001) demonstrated that a network trained for porosity prediction provides excellent accuracy for all pore fluids implying that, after training for different fluids and partial saturation, knowledge of the fluid properties is embedded in the network.

The purpose here is to establish networks for fluid saturation only using the log readings, without relying on functions that explicitly depend on porosity and auxiliary parameters derived from laboratory measurements. Since the network has to learn from data provided through a careful petrophysical analysis, the idea is not to eliminate the petrophysical work behind the saturation logs, but to transfer into the neural network for future application the effort and expertise already embedded in the petrophysical database. An obvious application is predicting while drilling, when the data required for conventional petrophysical evaluation are not yet available.

In this study we test the performance of alternative neural network configurations for saturation using model data and real data. We generate synthetic logs, with various levels of noise, for evaluating the optimal network

architecture by employing a model of mixed fluid saturation and the common formulae relating water saturation to well log response. From a large number of individually trained networks (~20) we select the best subset (~5-10) to be included in a committee neural net, or committee machine (CM), following the “test and select” approach suggested by Sharkey et al. (2000). We compare the performance of the individual expert of the CM with various methods of combining the ensemble to obtain best accuracy. For completeness, we also compare the neural network results with those of the conventional multiple linear regression (MLR) technique to demonstrate that the accuracy of the MLR approach is less than that of the simplest possible neural network architecture needed to represent the problem. Using a generalised committee neural network for fluid properties we apply the new technique to real data from the North Sea.

## 5.2 Water saturation measurement on cores

Apart from calculating water saturation ( $S_w$ ) on logs, which is the common industry practice, occasionally  $S_w$  measurements are made on cores also. Following are some of the methods by which  $S_w$  can be estimated on cores:

### 5.2.1 Retort method

This is a technique for measuring the fluid saturations in a core sample by heating the sample and measuring the volumes of water and oil driven off. The sample is crushed and weighed before being placed in the retort. It is then heated in stages or directly to 650°C during which the fluids are vaporized, collected, condensed and separated. Plateaus in the rise of the cumulative water volume with temperature are sometimes analysed to indicate when free water, surface clay-bound water and interlayer clay-bound water have been driven off. The volumes of water and oil are measured directly, but corrections are needed to account for alterations in the water and oil because of the dissolved gas. The volume of gas also is needed for accurate results. This is measured on a separate, adjacent sample by injecting mercury under pressure and measuring the volume absorbed. Before injection, the sample is weighed and its bulk volume determined by mercury displacement. The total pore volume is then the sum of the volumes of gas, oil and water. The saturation of each component is the ratio of its volume to the total pore volume.

### 5.2.2 Dean – Stark extraction

This a method for the measurement of fluid saturations in a core sample by distillation extraction. The water in the sample is vaporized by boiling solvent,

then condensed and collected in a calibrated trap. This gives the volume of water in the sample. The solvent is also condensed, then flows back over the sample and extracts the oil. Extraction continues for a minimum of two days until the extracted solvent is clean or the sample shows no more fluorescence. The weight of the sample is measured before and after extraction. The clean and dried samples are measured for porosity. The water volume from the extraction gives the water saturation values. The oil saturation can be calculated indirectly by using weight before and after extraction.

### 5.3 Log responses to pore fluids

The type of pore fluid, gas, oil or brine is clearly reflected in several well logs. Gas and water are significantly different in density and sonic velocity, while the differences are smaller for water and oil. Assuming a mixed pore-fill of water, oil and gas averaged over the scale of measurements, the density  $\rho_f$  of the composite pore fluid is given by

$$\rho_f = \rho_w S_w + \rho_o S_o + \rho_g S_g \quad (5.2)$$

where typical values for the densities of the constituents are  $\rho_w = 1.03$ ,  $\rho_o = 0.75$  and  $\rho_g = 0.25 \text{ g/cm}^3$ , and the partial saturations satisfy the equation

$$S_w + S_o + S_g = 1 \quad (5.3)$$

The fluid saturation profile is the same as used in the synthetic data for porosity prediction. The bulk density of the reservoir rock is given by

$$\rho = \rho_f \phi + \rho_m (1 - \phi) \quad (5.4)$$

where  $\phi$  is the porosity and  $\rho_m$  the density of the rock material, implying that the density log is sensitive to the pore-filling fluid as well as the properties of the rock itself. On the other hand, while  $\rho_f$  may vary within the wide range 0.2 - 1.1  $\text{g/cm}^3$  for gas and brine, respectively, the density of siliciclastic rock material varies only within a narrow range 2.64-2.80  $\text{g/cm}^3$  (Brigaud et al., 1992). The variations in bulk density within a typical North Sea siliciclastics reservoir may thus mainly be in response to variations in fluid content and composition of the pore fluid.

The transit time for the bulk can be approximated by

$$\Delta t = \Delta t_f \phi + \Delta t_m (1 - \phi) \quad (5.5)$$

where  $\Delta t_m$  is the transit time of the rock material, which varies in the range 56-75  $\mu\text{s}/\text{ft}$  (Brigaud et al., 1992). The sonic log is therefore a sensitive indicator of fluid content and fluid type.

Brine and hydrocarbons, in general, are highly different in resistivity with low resistivity for brine ( $\sim 1$  ohmm) and high resistivity for the hydrocarbon-bearing reservoirs ( $\sim 100$  ohmm or more). Resistivity, in fact, is the most important hydrocarbon indicator and, moreover, the resistivity for clean sandstone is directly related to the water saturation through Archie's equation (Archie, 1942)

$$R_t = \frac{FR_w}{S_w^n} \quad (5.6)$$

where the resistivity of the formation water  $R_w = 0.14$  ohmm, the formation factor  $F$  is given by  $F = \frac{a}{\phi^m}$ , where  $a = 0.625$  and the exponents  $m = n = 2$  using typical values for a North Sea sandstone reservoir. For sandstones containing clay various modifications of Archie's equation have been proposed which have the general form (Schlumberger, 1989)

$$1/R_t = \theta_0 S_w^2 + \theta_1 S_w \quad (5.7)$$

where  $\theta_0$  is the predominant sand term that is dependent on the amount of sand, its porosity, and the resistivity of the saturating water. The term  $\theta_1$  is the shale term that depends on the amount and resistivity of the shale. For clean sandstone (5.7) reduces to Archie's equation (5.6). One of the favourite models for calculating effective water saturation  $S_w$  in shaley formations has been provided by Poupon et al., (1971) who claim that their model is independent of the clay distribution. Based on a modified Archie's equation the relationship between the true resistivity and the formation parameters has been established by the equation

$$\frac{1}{\sqrt{R_t}} = \left( \frac{C^{(1-C/2)}}{\sqrt{R_C}} + \frac{\phi_e^{m/2}}{\sqrt{aR_w}} \right) S_w^{n/2} \quad (5.8)$$

where  $R_C$  is the resistivity of the clay and  $C$  is the volume fraction of clay in the formation as determined from the clay sensitive logs such as gamma ray, or from a combination of neutron and density. Here  $\phi_e$  is the effective porosity of the formation, i.e. excluding shale porosity. It is calculated from the density log, the

grain density measurements of matrix density and the fluid density using the standard density-to-porosity transform (equation 5.4). Then the latter is combined in a weighted sum of the density porosity and neutron porosity. Several logs other than resistivity thus form the input to the saturation models, plus supplementary laboratory data to determine the constants and exponents. Because of lack of data the common assumption in petrophysical evaluation is, however, that the formation factor  $a$ , the cementation exponents  $m$  and the saturation exponent  $n$  as well as the resistivity of the saturating water  $R_w$  are constants over the field, though significant variations are seen in the laboratory data. Provided a sufficient number of core samples has been analysed the actual values of  $m$  and  $n$  may be used rather than their mean values.

All the models from which water saturation is calculated are empirical, either determined during laboratory investigations, or from field experience. There are significant uncertainties in estimation of formation parameters. The relative change in  $S_w$  from Archie's equation arising from errors in all the measured variables can be given by:

$$\frac{\Delta S_w}{S_w} = \frac{1}{n} \left[ -\ln S_w \Delta n + \frac{\Delta a}{a} + \frac{\Delta R_w}{R_w} - m \frac{\Delta \phi}{\phi} - \ln(\phi) \Delta m - \frac{\Delta R_t}{R_t} \right] \quad (5.9)$$

Thus all the parameters  $R_w$ ,  $R_t$ ,  $a$  and  $\phi$  contribute to the total error  $\Delta S_w$ . If  $m=2$  porosity errors are twice as significant as resistivity errors. Errors in  $n$  could lead to significant errors in  $S_w$  at small water saturations, while errors in  $m$  can be important for low porosity media. Keeping all other parameters constant for a formation of 31.6% porosity and 31.6% water saturation with  $m, n=2$  a relative error of 2.5% in the measurement of  $a, m, n, \phi, R_w$  and  $R_t$  taken one at a time causes a relative error of 1.25%, 5.7%, 3%, 2.5%, 1.25%, and 1.25% in  $S_w$  values, respectively. The maximum errors in  $S_w$  are due to errors in  $m$  and  $n$ . A relative error of 2.5% in the measurement of  $n$  is very optimistic. In reality it can be up to 20% which can lead to large uncertainties in  $S_w$  values.

The parameter  $m$  is also affected by a large number of factors including grain texture, pore configuration and size, constrictions existing in a porous system, tortuosity, type of pore system (inter-granular, inter-crystalline, vuggy, fractured), compaction due to overburden pressure and presence of clay minerals. The main effect of these parameters is to modify the formation resistivity factor  $F$ . Consequently, their combination can produce a range of values of  $F$  and  $m$  for a given porosity. In a case study in a reef type limestone (Tiab and Donaldsson et al., 1996) when the overburden pressure increases from 0 to 35 MPa, the value of  $m$  increases from 1.99 to 2.23, causing a relative error of 0.5 % to 11.5% in the value of  $m$ .

Furthermore the parameters are generally obtained under ambient conditions instead of reservoir conditions. In the study made by Søndena et al. (1990) the results from the electrical resistivity measurement from cores from a North Sea reservoir showed that the saturation exponents obtained at the reservoir conditions are lower than those obtained under ambient conditions and that they also slightly increased when the effective stress was increased. Thus it is crucial to have careful laboratory measurements of  $m$  and  $n$  parameters. However in the present study we have not eliminated these errors from  $S_w$  as the errors are already there in the  $S_w$  calculation from CPI logs against which we have calibrated the neural network model.

Being aware of the uncertainties in the empirical relation Woodhouse (1998) suggested  $S_w$  measurement by the extraction of reservoir formation water from core plugs cored with oil-based mud. The study shows that after small systematic corrections the cores gave accurate in situ reservoir  $S_w$  measurements valid over a wide range of  $S_w$  values and throughout most of the transition zone (a reservoir interval extending from the fluid contact upwards, where water saturation is higher than the irreducible saturation). During coring no significant amounts of reservoir formation water were mobilised and displaced from the cores except those, which were taken from and below the OWC. Rapidly drilled cores, with incomplete penetration and pressure-retained cores provided conclusive proof. He found out that  $S_w$  measurements by this method are consistent with the  $S_w$  measured on logs or by capillary pressure method.

The neural network model is fully empirical, non-linear and may even be trained to display the expertise of a skilled petrophysicist. The idea here is thus to establish networks for fluid saturation prediction based on a petrophysical database.

## 5.4 Synthetic data

Before applying a new method to real data the common practice in development and testing of geophysical methods and algorithms is to use synthetic data in order to maintain full control. We use for simplicity the clean sandstone model (equation 5.6). We generate the three synthetic logs, which have clear physical relationships to the fluid properties for input to the prediction; i.e. density, sonic and resistivity. The generation of these three synthetic logs is the same as we have discussed in section 3.1 with porosity prediction. In addition we include the neutron porosity log, which is an indicator of the abundance of hydrogen nuclei, and a common indicator to distinguish whether hydrocarbons are in the gas or fluid phase.

From the partial fluid saturation model of equation (5.3) and with  $\phi$  from equation (5.4), we obtain the neutron porosity log from the relation

$$\phi_N = \phi(S_w H_w + S_o H_o + S_g H_g) + (1 - \phi) H_s \quad (5.10)$$

where  $H_i$  is the hydrogen index of the constituent  $i$ . For the different pore fill components the hydrogen indices are given by  $H_w = 1.1\rho_w$ ,  $H_o = 1.1\rho_o$  and  $H_g = 1.2\rho_g$ , and for the rock material (sand) we use  $H_s = 0.001$  (Schlumberger, 1989). The four synthetic input logs; density, sonic, resistivity and neutron porosity ( $\rho$ ,  $\Delta t$ ,  $R_t$  and  $\phi_N$ ) are shown in Figure 5.1 without noise (a) and with 10% noise added (b), respectively. Four different data sets, each consisting of 3000 samples, were created with independent random noise of 0, 2.5, 5 and 10% added to each log, including the saturations. Subsets consisting of 150 samples, or 5% of the total record, were then selected at regular depth intervals to use as training patterns for the neural networks.

## 5.5 Optimal design of networks

In the optimal design of fluid saturation network other than the two basic questions of 1) how much data and 2) how many neurons do we need we are faced with the additional problem of determining the number of outputs; i.e.

- i) Is prediction of a single component; e.g. the water saturation  $S_w$  sufficient?
- ii) Should we rather determine two outputs; e.g.  $S_w$  and  $S_g$  and compute the third saturation  $S_o$  from equation (5.2) ?
- iii) Would it be possible for a neural net to predict all three saturations simultaneously?
- iv) What is the optimal number of hidden neurons in cases of i), ii) and iii), and what is the corresponding number of training patterns required?
- v) Should we use separate networks for each fluid component rather than multiple outputs?

The answer to (i) may be that in the case of an oil field water and oil are the main fluid components. We thus assume  $S_g = 0$  and hence the oil saturation is given by  $S_o = 1 - S_w$ . A similar argument applies to a gas field when away from a gas-oil transition zone where the components may be present in comparable proportions. But the problem arises in oil- water and gas-oil transition zone for knowing the partial saturations of the three fluids independently.

In case (ii) we have better control since prediction of two components provides the value of the third from equation (5.3). Only two saturations are, however, independent estimates and hence there is no means of control by summing to unity.

In case (iii) the three saturations are independently predicted and, moreover, equation (5.3) provides an independent control of the prediction quality by the fact that the three output neurons should sum to unity within the estimated error. For the number of neurons in the hidden and the output layers, the obvious relation exists that the number of outputs cannot exceed the number of the neurons in the hidden layer, e.g. a single hidden neuron cannot simultaneously transfer non-trivial data to more than one output neuron. Thus, given the number of outputs, only the minimum number of hidden units can be fixed and the optimal number remains to be determined. The answer to (iv) is likely to be that the more outputs the more hidden units are required to reach comparable performance to that of the single output. More hidden neurons, on the other hand, imply a more complex network and hence more training patterns required to achieve the goal. However, the optimal number of hidden units and the corresponding number of patterns needed remain to be determined by experiments. For this reason we exploit the model data, with added noise levels 0, 2.5, 5 and 10%, to investigate the above questions in more detail. For training the networks we have selected subsets of 25-300 patterns, sampled at regular depth intervals, whereas for the error analysis all tests are made against the 3000 samples in the logs.

With the four input logs and a single output of water saturation  $S_w$  we find that with minimum number of hidden units, which is one in case of a single output, the network clearly fails to reproduce the model as shown in Figure 5.2 and Figure 5.3. Using the same data (150 patterns) to fit a multiple linear regression model we find, on the other hand, that the model fit is much worse even though the number of coefficients to fit is the same (four) for the two models. The assumption of linearity of MLR versus the embedded non-linearity in the neural network explains the difference. Moreover, by adding one more neuron to hidden layer we gain significant improvement in favour of neural network, in the noise-free data as well as in the case of 10% noise as shown in Figure 5.3. The latter is also evident from Figure 5.4a and b showing that the error drops significantly when the number of neurons changes from one to two. With a further increase in the number of neurons the bias still changes but the standard deviation remains almost constant for higher noise level. By adding neurons beyond 2 to the hidden layer we, in general, still gain accuracy up to 4 hidden neurons but thereafter the network becomes more sensitive to noise and the errors increase. With an optimal number of 4 hidden neurons the error of the  $S_w$  network is about 0.02, which is below the error level expected in practical situations. Similar results are obtained with two outputs, and with three outputs as

shown in Figure 5.5 and Figure 5.6. Here we find that the optimal number of hidden neurons is 8 and 10, respectively, in case of 2 and 3 outputs.

For the model used in this study the number of outputs thus seems to have minor effects on the accuracy as shown in Figure 5.7 for water saturation (a) and gas saturation (b). For all fluid saturations the standard deviation  $\sigma$  for noisy data (10 %) is almost constant at about 0.02 and, moreover, apparently independent of the number of output neurons. Once sufficient numbers of well-represented training samples are available the bias is negligibly small as shown in Figure 5.8 and the standard deviation clearly becomes the leading term in the overall error in noisy data. With a single output the appropriate number of training facts is in excess of 100 samples, however, in general, the error still continues to decrease with increasing number of facts. We have thus chosen 150 facts in the above experiments.

A major assumption in network design is that a sufficient number of well-represented training samples is available. In many real situations, however, training samples are expensive or difficult to obtain. Since the problem complexity dictates the size of the network, we may end up with fewer useful facts than required for a network of a given size. The challenge in network design has been the small number of available data, and various methods have been proposed for generating virtual samples (Cho et al., 1997; Wong et al., 2000) to compensate for an insufficient number of training samples. For the problem at hand the number of reliable estimates from fluid transition zones, with at least two fluids present and of known partial saturation, is normally the main limitation. Since the overall errors are the same for a simple network with one single output and four hidden neurons, as for the more complex network with three outputs and 10 hidden neurons, the choice of architecture that requires minimum training patterns is preferred. An additional benefit of a simple network specialised for a particular fluid, besides the reduced training time, is the modularity that can be achieved when such a network constitutes a building block of a committee network as will be demonstrated in the following section.

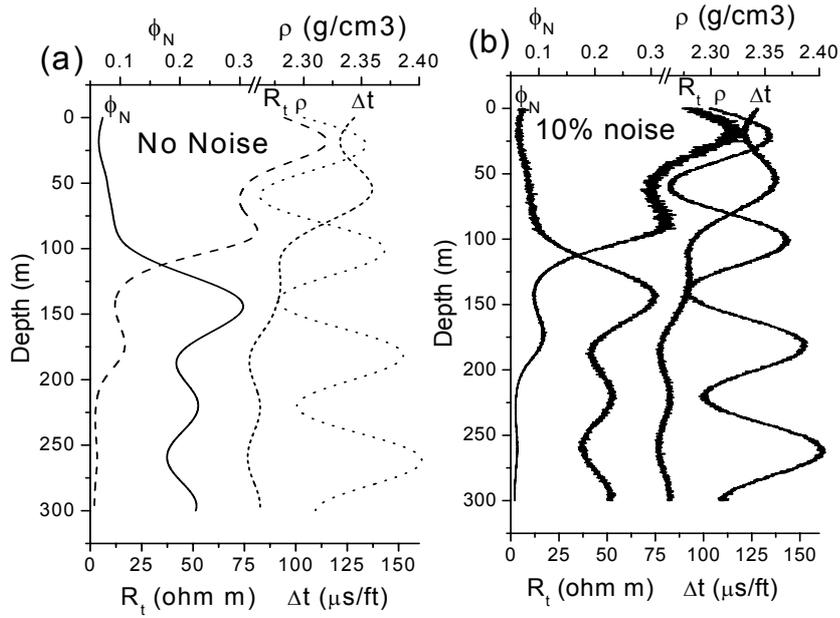


Figure 5.1: Model output of synthetic logs for the saturation network (a)Noise free and (b) added 10 % random noise

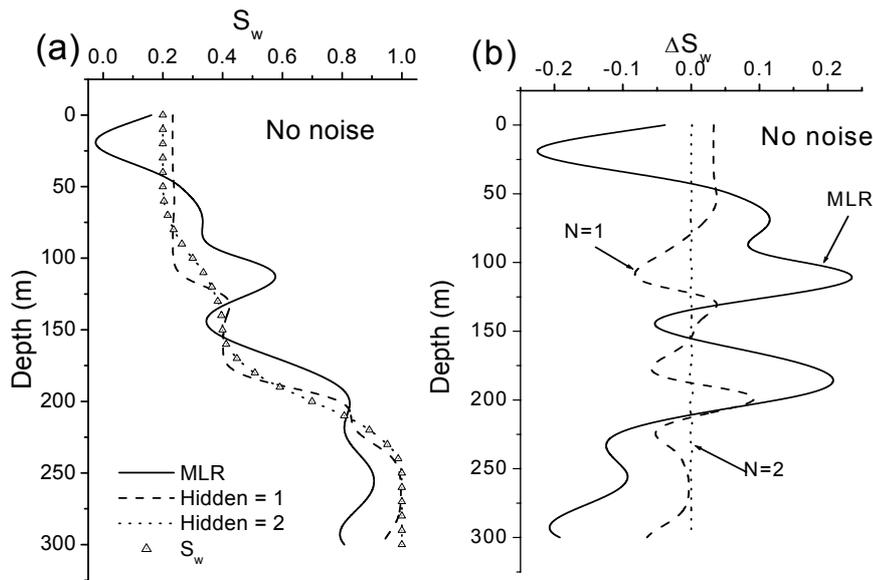


Figure 5.2: a)Comparison of  $S_w$  predicted by MLR and neural network with one and two neurons in hidden layer noise free case b)differences with model  $S_w$

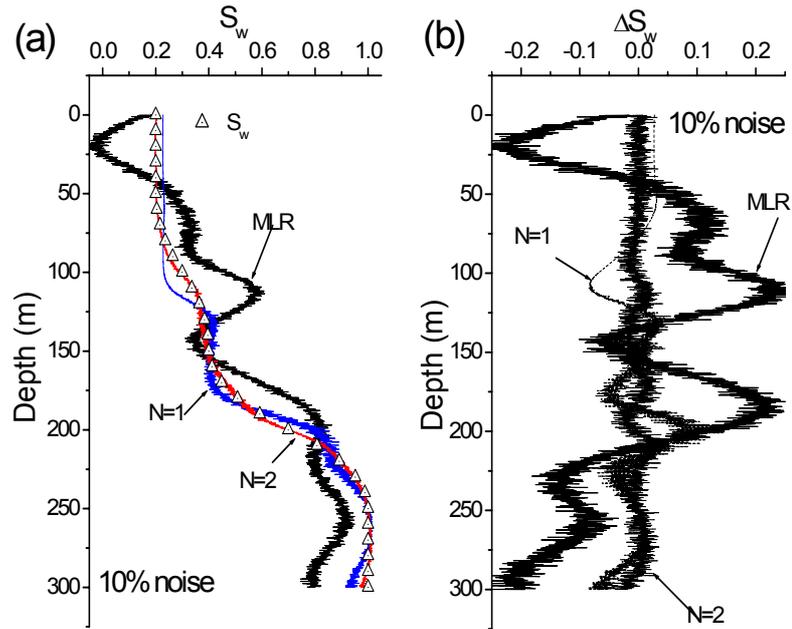


Figure 5.3: Comparison of  $S_w$  predicted by MLR and neural network with one and two neurons in the hidden layer for 10%

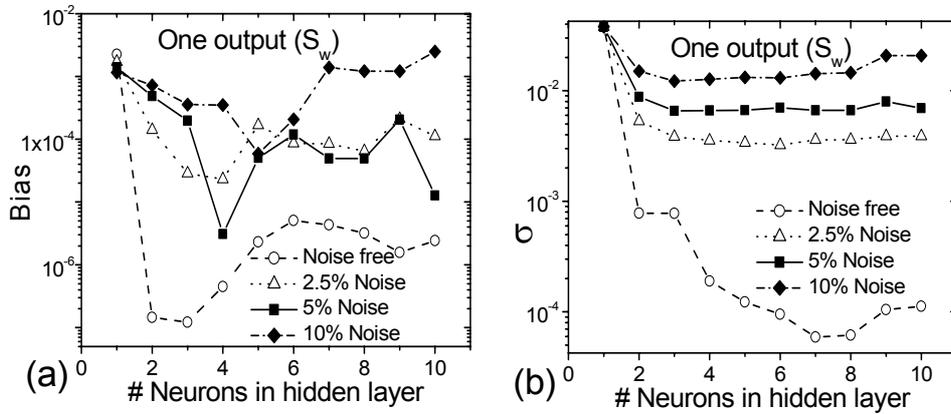


Figure 5.4: (a) Bias, (b) standard deviation for one single output ( $S_w$ ) with number of neurons in hidden layer for different noise levels.

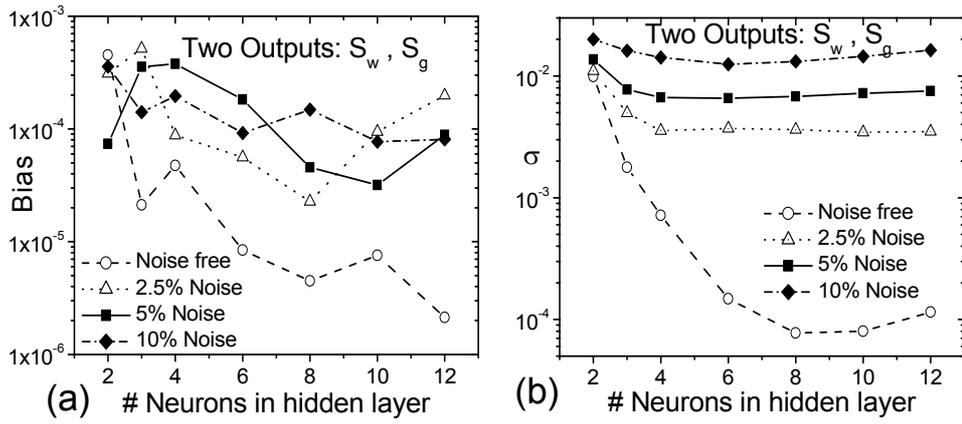


Figure 5.5: (a) Bias, (b) standard deviation from network with two outputs ( $S_w$  and  $S_g$ ) with number of neurons in hidden layer for different noise levels

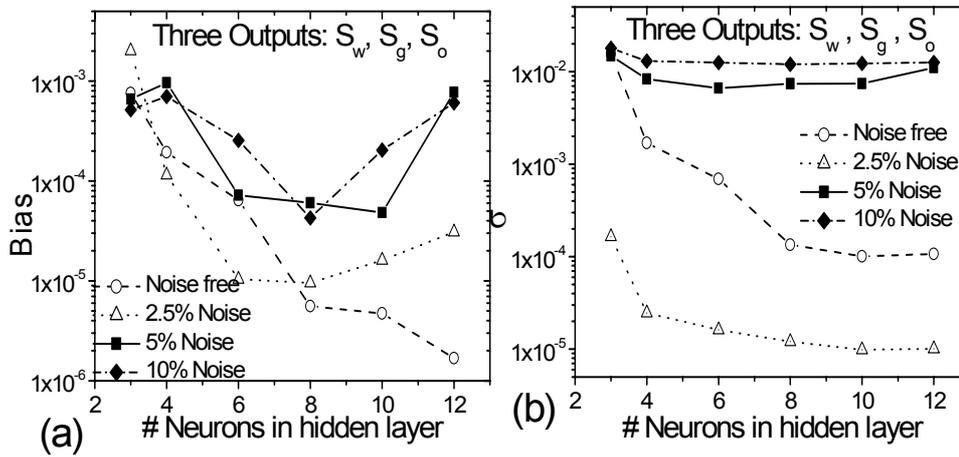


Figure 5.6: (a) Bias, (b) standard deviation from network with three outputs ( $S_w$ ,  $S_g$ ,  $S_o$ ) with number of neurons in hidden layer for different noise levels

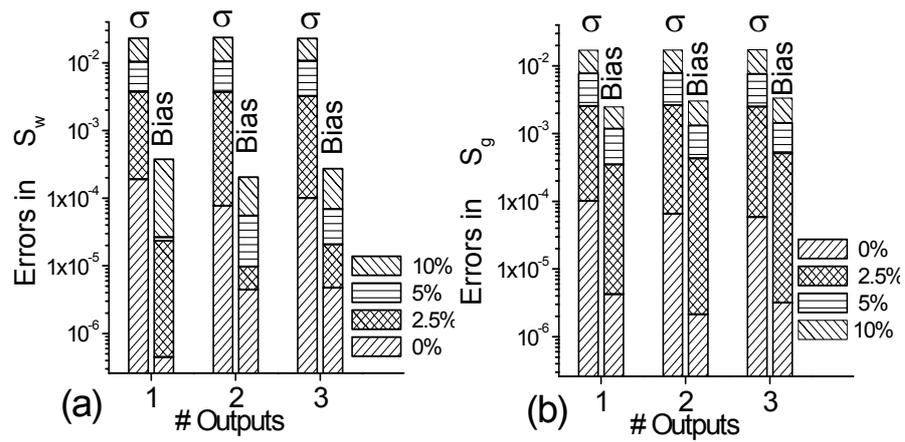


Figure 5.7: Errors in (a)  $S_w$  and (b)  $S_g$  with number of output neurons and optimal hidden neurons of 4,8,10 for 1,2,3 outputs respectively

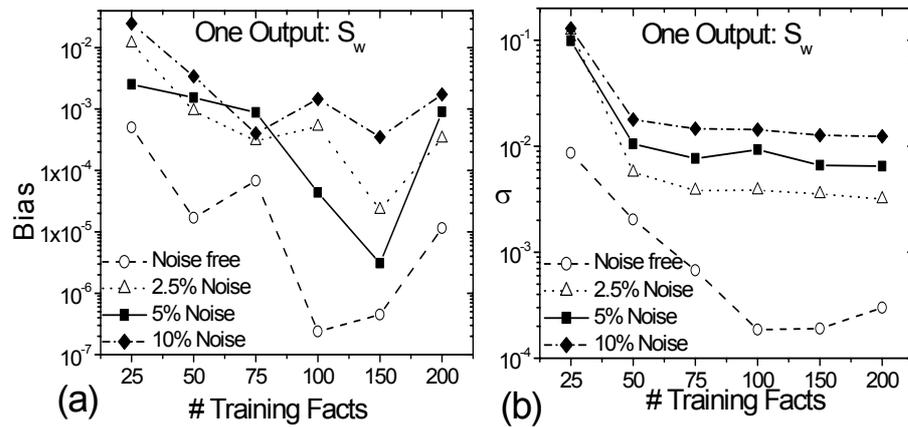


Figure 5.8: (a) Bias, (b) standard deviation for one output ( $S_w$ ) with the number of training facts for network with four hidden neurons.

## 5.6 CM architecture

Since we know from the previous discussion the advantages of CM architecture over the individual network we also used CM for partial fluid saturation prediction. The architecture of the CM is the same as used for porosity prediction as shown in Figure 3.11. Individual CMs have been developed for calculating the saturation of each fluid. The saturation CM for each fluid, with nine building blocks consisting of the simple 4-4-1 MLP (4 input-4 hidden-1 output) has been applied to our model data. The input neurons are the four logs i.e. density, neutron porosity, resistivity and sonic. For each saturation 20 networks have been trained by overtraining approach and 9 have been carefully selected using the 'test and select approach' as suggested in Sharkey et al. 2000. The output of the 9 networks is then combined by ensemble average and by the OLC approach. The resulting tests with 10% noise are shown in Figure 5.9 by the stacked bias and variance histograms that convincingly demonstrate the improved accuracy of the CM compared to the single MLP. Simple ensemble averaging of the output significantly reduces the errors, however, the power of the OLC is clearly demonstrated by the fact that the overall bias has been eliminated. The errors in the final saturation CMs have now been reduced to a variance of less than  $2 \times 10^{-4}$  or a standard deviation of less than 0.015 for  $S_w$ , and slightly smaller for  $S_o$  and  $S_g$ . The outputs from the three saturation CMs are shown in Figure 5.10, where their sum is also displayed as a means of independent quality control.

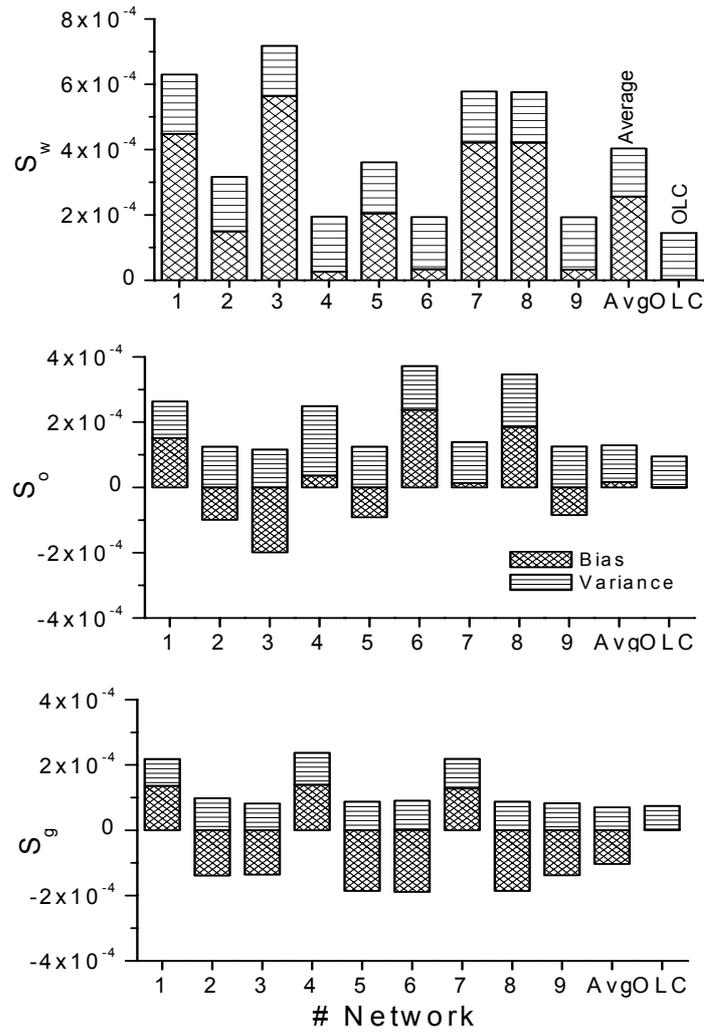


Figure 5.9: Stack of bias and variance for each experts (1-9) and ensemble average and OLC in three saturation committees for water, oil and gas

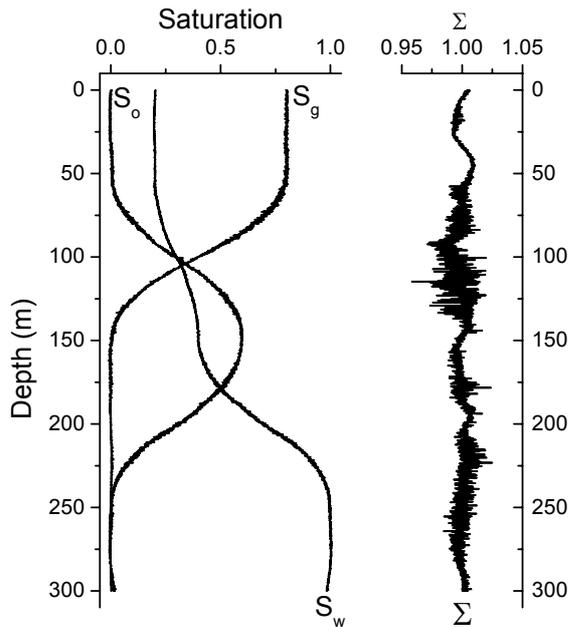


Figure 5.10: Output from the three saturation CMs and the sum of saturations  $\Sigma = S_w + S_o + S_g$  for independent control

## 5.7 Real data

Reliable direct measurements of fluid saturation on cores are expensive, time consuming and hard to obtain. In the laboratory the fluid saturation of cores can be measured by different methods as discussed but the main problem lies that some water might be lost while the core is cut and brought to the surface or may be evaporated at the rig when exposed to the atmosphere or fluids such as gas may be expelled when the pressure on the sample is reduced. So the  $S_w$  measured on the core may not be the correct representation of the in situ  $S_w$  in the reservoir. Moreover a small core plug may not be the true representative of the whole formation.

Thus we cannot collect sufficient well-represented training data unless we rely upon the log interpretation and careful analysis made by a skilled log analyst. The calculated water saturations are based on the standard models e.g. shaly sand model, dual water model etc. Here the main input is resistivity and porosity, but auxiliary data (such as shale fraction) is in advance estimated based on for example density, neutron, and/or gamma ray. Thus behind the water saturation

calculation, as in the neural network approach, there is more than one log involved to form the final output.

The saturation in the field above the plane of free water level ( $S_w=1$ ) can also be determined if we know the capillary pressure ( $P_c$ ) versus  $S_w$  curve for the cores from laboratory. Thus if we know the density of the wetting and the nonwetting fluid and height of the core above the free water level then capillary pressure can be found out. Once knowing the capillary pressure water saturation can be determined by  $P_c$  vs  $S_w$  curve (Bear, 1988). The free water level can be identified by plotting the pressure points of the respective fluid obtained from a modular formation dynamics tester (MDT) or equivalent tools. Thus to obtain a complete profile of the saturation by other conventional methods implies some effort and more data than the standard well logs. However, once reliable CPI data have been established for a field, we may collect a sufficient number of well-represented training facts to establish neural nets to accomplish the same petrophysical tasks, with equal accuracy, in future wells in that field. The idea behind this study was to make the analysis of water saturation faster and accurate for the new wells in the field by embedding the information from the previous wells in the neural net.

A generalised MNN for fluid properties is shown in Figure 5.11, consisting of three expert CMs with a number of MLPs. The architecture of MLP is as shown in Figure 2.4 with the number of neurons in the input, hidden and output layer as 4-4-1 respectively. For convenience we use 9 MLPs, selected from the best of a larger number (~20) of candidates. The output of the MLPs is combined by OLC method. Our training patterns are all based on the water saturation values from the standard CPI logs and the hydrocarbon identification. In order to add values from transition zones we have used the well logs and the relations (5.2)-(5.8) to solve for  $S_g$  and  $S_o$  in the fluid mixture. For obvious reasons, the patterns for water saturation cover the entire range and are well distributed, while the patterns for oil and gas are dominated by values in the upper and lower bounds. In particular gas saturation data of intermediate values are hard to obtain since the gas-oil transition zone is narrow. For example it is about 3m. in the Oseberg formation as it is a good porous, permeable and homogeneous reservoir and so the log gradients are steep. Overall we have selected a set of 150 training patterns that are shared by the three saturation networks having 40 training patterns from the transition zone of well 30/9-B-20. The error distribution after training the 9 MLPs in each CM, and those from the combined output from the three saturation CMs, are all shown in Figure 5.12. Compared with the synthetic data test as shown in Figure 5.9 where the variance and bias contribute by the same order of magnitude in the overall error, the variance here surpasses the bias by an order of magnitude in the case of real data. As before, we eliminate the bias by OLC while the variances for  $S_w$  and  $S_o$  are only slightly reduced compared with the simple average. For  $S_g$ , on the other hand the errors are completely eliminated.

The output saturation predictions from the generalised fluid MNN are shown in Figure 5.13a and Figure 5.14a. By access to the complete set of saturation logs we are able to discriminate between alternative fluid systems; i.e. predominantly oil-water or gas-oil which has a direct application while drilling.

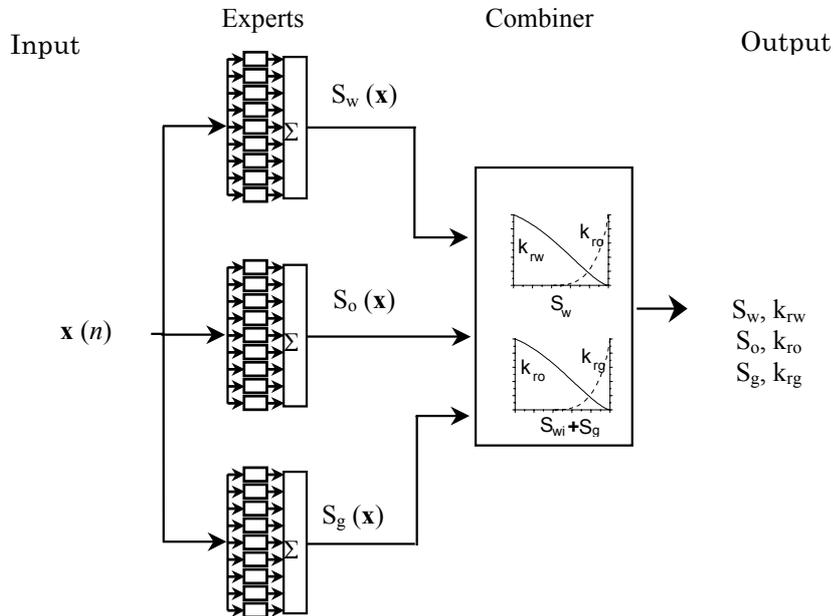


Figure 5.11: Block diagram of the generalised fluid property MNN

Well 30/9-B-20 is drilled through three main reservoir sands in the Brent group, where the Tarbert formation is oil-bearing, Ness is predominantly water-bearing but with patches of oil, while the main reservoir sand in the Oseberg formation is gas- and oil-bearing with a GOC around 3198m (Figure 5.13). As seen from the resulting saturation values, the changes in fluid properties are well detected by the network. There is a very good match between the  $S_w$  values predicted by the network and the  $S_w$  values from CPI log. In the gas and oil zones the  $S_w$  values are around 0.1, which is the irreducible water saturation. From this well mainly in the transition zone around the gas-oil transition in the Oseberg formation, we have selected 40 (out of total 150) training patterns. This well is thus partly known to the network in 40 out of the 900 depth samples shown. A well, which is completely unknown to the network, is therefore appropriate for testing the network as shown in Figure 5.14.

Well 30/9-B-24 K is essentially water bearing in Tarbert, except for oil patches in the upper part, while the Ness and the Oseberg formation are oil-

bearing. The hole deviation is 35 degrees. The well is unknown to the network. There are no fluid contacts in the well only the lithological boundaries. There are no intervals of gas in this well as reflected in the predicted gas saturation  $S_g \approx 0$ . No significant discrepancies between predicted  $S_w$  and the CPI logs are seen.

The absolute permeability of a rock is defined as the permeability of the rock when only one single fluid is present. Darcy's law uses it to describe the flow of a homogeneous fluid. However normally in hydrocarbon bearing reservoirs there are two or three fluids present together. In such cases effective permeability is the permeability of a rock to a particular fluid when that fluid has a pore saturation of less than 100%. Their relative viscosities and their relative permeabilities determine the relative rates of flow of the fluids. The relative permeability is defined as the ratio of the effective permeability to the absolute permeability. The relative permeability curves for a rock can be plotted in the laboratory by measuring the relative permeabilities of each fluid on cores. Figure 5.15 shows the normalised relative permeability curves for the Oseberg formation.

After the calculation of  $S_w$ ,  $S_o$  and  $S_g$  by the three committees, the saturations are collected into the combiner as shown in Figure 5.11. In the combiner we also add the normalised relative permeability curves of the Oseberg formation. So using the saturations and the equations generated on the normalised relative permeability curves we can generate relative permeability logs for the different fluids as shown in Figure 5.13b, Figure 5.14b in the Oseberg formation. The normalised gas oil relative permeability curves are used above GOC where  $S_g$  is varying and water is at irreducible water saturation and oil-water normalised relative permeability curves below GOC where  $S_o$  is varying and water is at irreducible water saturation. As a check, in gas zone we find the highest relative permeability, which is of gas, and in the oil zone it is of oil, which is as expected. This indicates that above GOC the mobile phase is gas and below GOC above OWC the mobile phase is oil. These relative permeability logs can be used in real time by reservoir engineers as an input to the reservoir simulation.

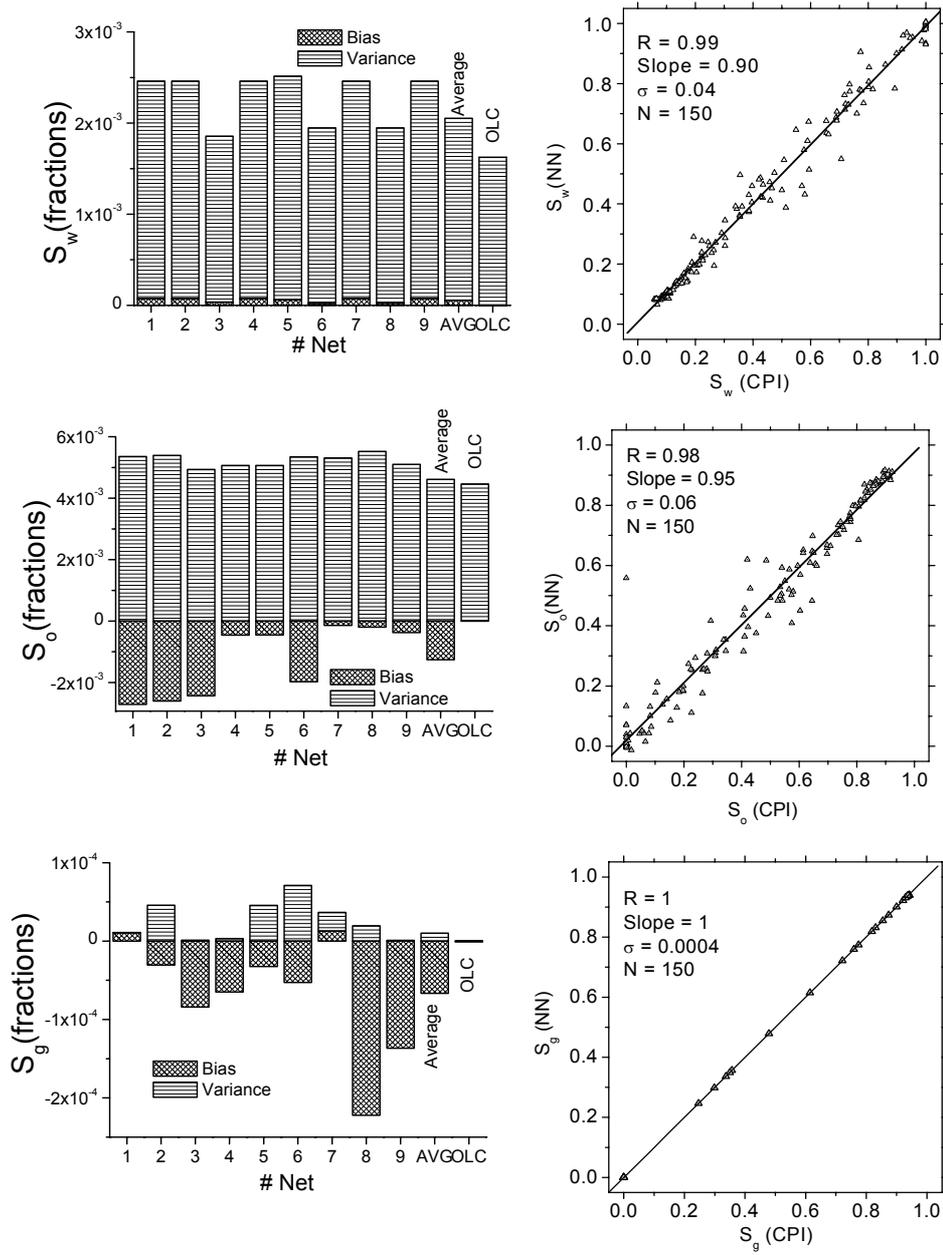


Figure 5.12: (left) Bias and std. deviation for one single output ( $S_w$ ) for 150 training facts (right) Crossplot of training facts and OLC outputs with linear fit

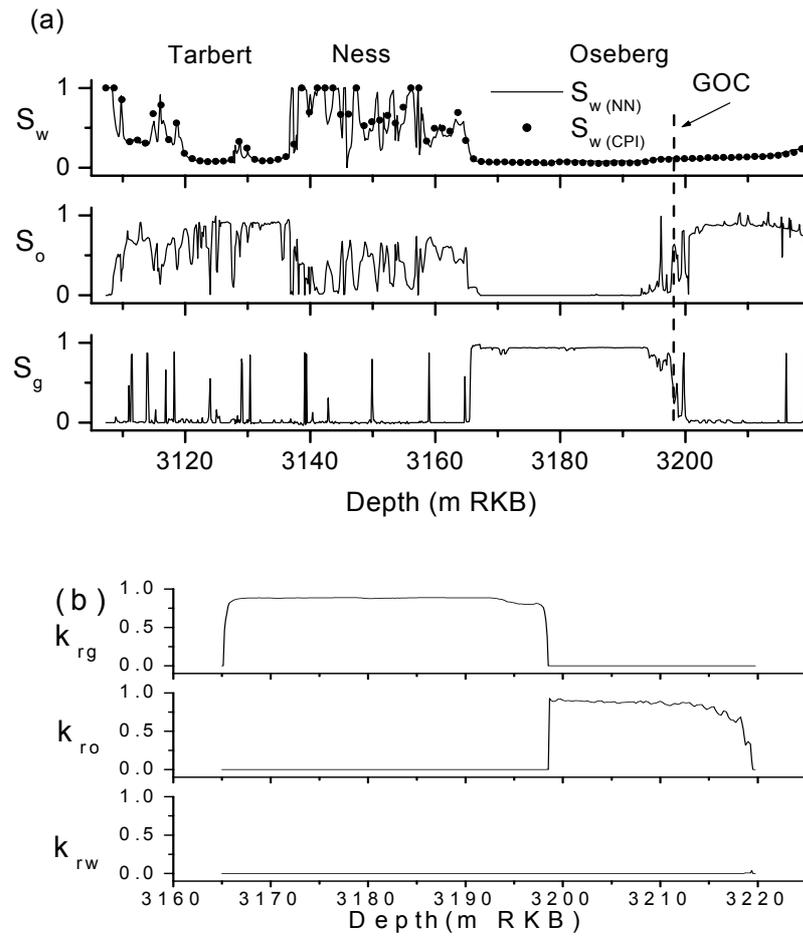


Figure 5.13: Prediction of (a) partial fluid saturation and (b) relative permeability logs in well 30/9-B-20 in Oseberg formation

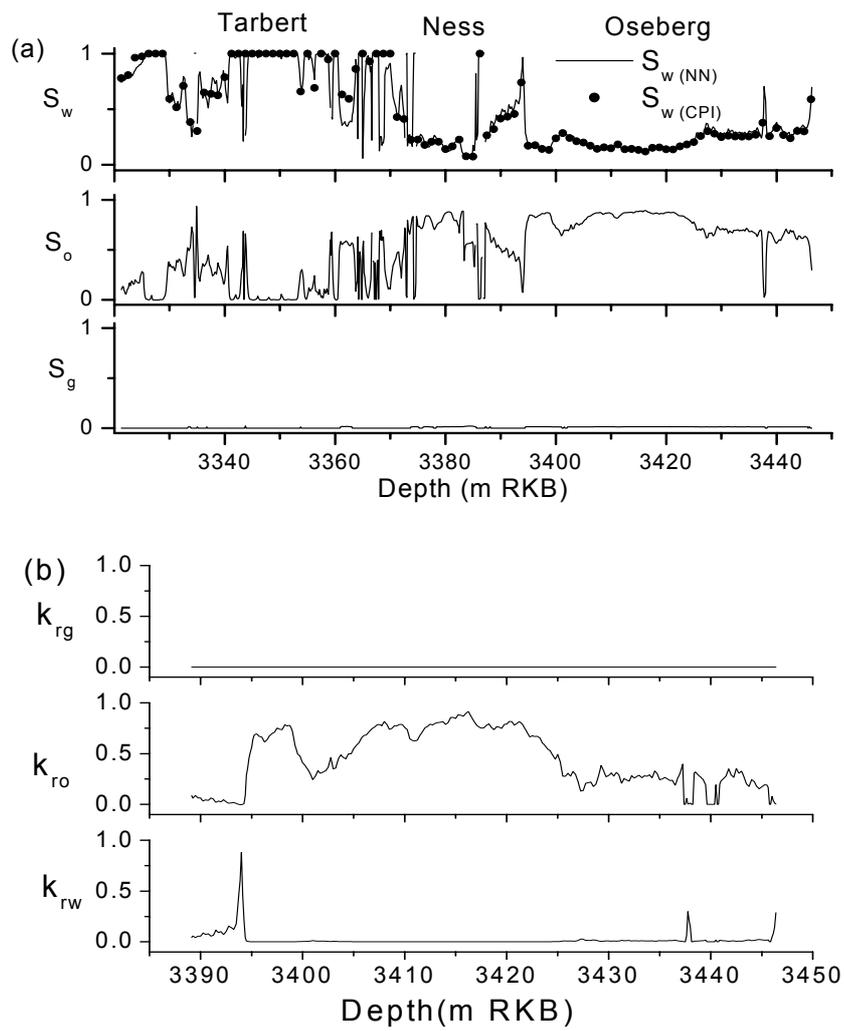


Figure 5.14: Prediction of (a) partial fluid saturation and (b) relative permeability logs in well 30/9-B-24 K in Oseberg formation

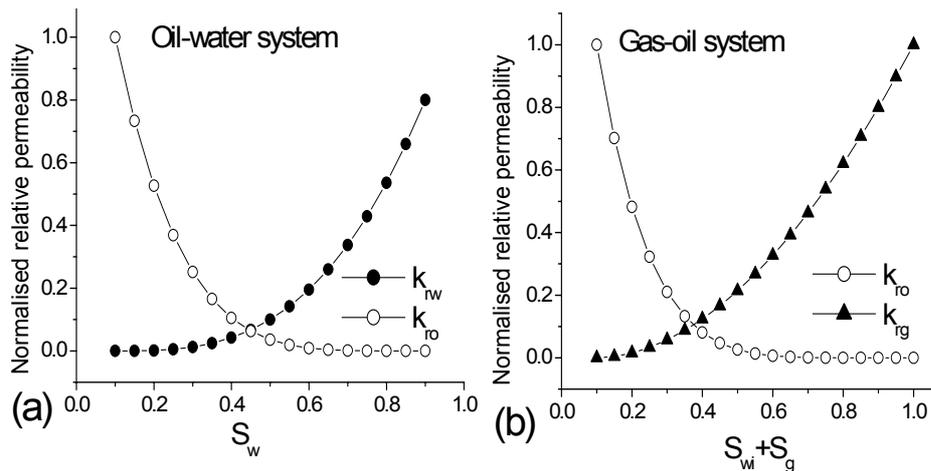


Figure 5.15: Normalised relative permeability curves for Oseberg field a) for oil-water system b) for gas-oil system

## 5.8 Conclusions

The prediction of partial fluid saturation within the reservoir using a neural network has many advantages. In this study we have developed and tested a modular artificial neural network system for predicting the underground fluids water, oil and gas, and their partial saturation directly from well logs, without explicit knowledge of the fluid and rock properties normally required by conventional methods. The idea of using a neural network for fluid saturation is not to eliminate the careful petrophysical evaluation behind the CPI log, but to transfer into the neural network for future application the effort and expertise already embedded in the petrophysical database.

The inputs to the neural network are density, sonic, resistivity and neutron porosity. The numerical experiments on the synthetic data based on a realistic petrophysical model are very useful in identifying the architecture of the network and the size of the training dataset. From this experiment we concluded that although output of three saturations from a single MLP (4-10-3) reveals the same accuracy as those of three individual MLPs with one output (4-4-1), the latter has the advantage of simplicity in terms of number of neurons, which implies fewer training patterns and faster training. Moreover, simplicity in the MLP improves modularity when used for building blocks in the multi-net system.

For the optimal architecture of MLP with 4-4-1 neurons in the input, hidden and the output layer the number of training patterns should be in excess of 100 to ensure negligible errors in case of data with moderate noise.

A committee neural network for each fluid type is the preferred solution, with each network consisting of a number of individually trained 4-4-1 MLPs connected in parallel and redundantly combined using optimal linear combination. Compared with a single MLP realisation the OLC approach implies an overall error reduction by an order of magnitude.

Building a modular neural network consisting of three CM's one for each fluid type where each CM contains a combination of redundant networks based on the optimal architecture whose outputs are combined using OLC approach gives a good saturation prediction of the fluids simultaneously in the reservoir. The saturation predictions from the fluid CMs are further combined in a modular neural network with laboratory measurements of normalised relative permeability curves for oil-water and gas-oil fluid systems to output relative permeability logs for the three fluids.

The accuracy in prediction saturation essentially depends on the accuracy of the training patterns, which are from the CPI logs, and the accuracy of the individual log measurements. Comparison of  $S_w$  values of the neural network with those of CPI logs, in wells that are unknown to the network, indicates a standard deviation error of less than 0.03, over the complete Brent formation in the North Sea which is sufficiently low for all practical purposes. Fluid saturation prediction and generation of relative permeability logs, which can be given as, input to reservoir simulation while drilling is an obvious application of this technique in real time.

# Chapter 6

## Lithofacies prediction

### 6.1 Introduction

The primary task of geological and engineering characterisation of a petroleum reservoir is to determine the various lithofacies of the reservoir rocks from the examination of cores taken from wells. For technical and financial reasons, however, cores are not always sampled in the entire interval of interest. In highly deviated and horizontal wells only log data are available. Geological stratification and identification of lithofacies, as well as estimation of reservoir and fluid properties thus rely upon interpretation of the well logs.

As a prerequisite for the zonation the continuous logs first have to be segmented into discrete zones of similar properties, which are the elementary units of reference for inferring the correlation between wells. The context and objectives of the analysis determine the nature of the zones, which may designate lithostratigraphic or petrophysical units or other meaningful geological entities. Facies from log measurements are equivalent but not identical to the lithofacies inferred from core data because logfacies are the combined results of indirect log response to lithology and fluid, whereas lithofacies are defined directly from the visible features of rocks.

Manual interpretation of facies from well logs is a labour-intensive process that implies a considerable amount of time by an experienced log analyst. Since the early days of the introduction of computers to geology, computerised numerical procedures mixed with expert reasoning have been employed to emulate the heuristic intuitive pattern recognition criteria in the zonation. There are several advantages in computerising these procedures. Numerical methods are more powerful than the human brain in simultaneously comprehending a suite of multivariate data. Moreover, the results become objective, consistent, reproducible and free from personal bias by the log analyst.

Numerous numerical methods to carry out these fundamental tasks have been proposed. Wolff and Pelissier-Combescure (1982) used FACIOLOG for automatically zoning logs into lithofacies. The technique used includes principal component and modal distribution analysis. With a suitable selection of input logs and zoning algorithm parameters a set of lithofacies can generally be related

to the actual geological facies. Boundaries are first detected to identify thin beds, transitions and unstable zones for special treatment.

Delfiner et al. (1987) and Busch et al. (1987) used discriminant factor analysis to correlate the log values with lithological units. Discriminant factor analysis projects the original log space into a target space of lower dimension such that the projected cluster centres are as far apart as possible and the projected points of the same cluster are closest to each other. In other words, the emphasis is on the selection of the parameters that provide the most discrimination. Delfiner et al. use a database of identified lithologies to calibrate their logs whereas Busch et al. use only the core data available from a given field.

Reviews of the early works on computer-aided log-correlation are provided in Fang et al. (1992) and Gill et al. (1993). Fang et al. (1992) employed pattern recognition techniques (dynamic waveform matching) to identify logfacies. In this method manual or machine zonation of the logs is performed first. Then log attributes of two logs are compared and matched in a pass. Each matching cycle comprises three passes, thus ensuring optimal global correlation.

Gill et al. (1993) used multivariate-clustering procedure to reach the same objective. In this method based on the discrimination of zones and logfacies from log suites clusters were defined such that within each cluster dispersion is minimal. There are two clustering techniques, which have been discussed. In constrained analysis, clusters of lower level are grouped to form new, higher level clusters only if their members are vertically next to each other whereas in the unconstrained analysis this is relaxed and the programme is free to form clusters as it sees fit, based solely on log readings. The unconstrained analysis successfully identified logfacies, specific to a particular zone.

Neural networks in facies classification were introduced during the early development of the technology. Rogers et al. (1992) applied a feed forward back-propagating neural network (BPANN) to classify lithology from well logs. The network configuration is 3 neurons in the input layer corresponding to gamma ray, density and neutron porosity logs. The hidden layer consisted of 3 and 4 neurons corresponding to two datasets. The output layer consisted of 4 neurons corresponding to the classification of four lithofacies, which are limestone, dolomite, shale and sandstone. In a comparative study of Wong et al. (1995) a similar neural network approach was shown to be in favour of the discriminant analysis in logfacies classification.

Zhang et al. (1999) identified lithofacies by integrating back propagation and self-organising neural networks. The log data was first divided into intervals according to the lithofacies identified from cores and then clustered to form electrofacies categories by using a self-organising neural network. The average

values in each of the electrofacies categories are then consequently selected to be the data to train the back propagation network.

Although BPANNs for the most part give satisfactory results, Saggaf and Nebrija (2000) argue that the BPANN fails to generalise and therefore suggested self-organising competitive supervised and un-supervised neural nets (Kohonen, 1989) as a better alternative for logfacies prediction. However, to resolve ambiguities in prediction they had to introduce additional constraints by prior knowledge of the stratigraphy.

Neural network methods are in general superior to knowledge-based and rule-based expert systems, as they have better generalisation and fault tolerance. The network is able to generalise from the training data so that its output is similar for inputs that are almost, but not quite the same. Conventional expert systems, on the other hand, are based on logic rules and the concept of similarity cannot be attributed to logical rules.

Improved generalisation of the BP-ANN can be achieved by several means: Firstly by optimum training approach, validation or weak over-training. Secondly by training several neural networks using independent (disjoint) training sets and thirdly by forming ensembles from networks trained from random initial conditions.

In this study we combine all the listed methods. The approach relies upon combining back-propagation neural networks in ensembles and modular systems, where the multi-class classification problem of facies identification has been reduced to a number of two-class problems. A fundamental property of a logfacies, i.e. that it is composed of several sequential points along the well bore, has been exploited by using a recurrent back propagation neural network adopted from time-series analysis. We optimise the modular system by using synthetic logs and apply the networks to facies zonation in North Sea wells.

## 6.2 Multiclass classification using modular neural networks and stacked generalisation

A single MLP network consisting of an input layer, a hidden layer and an output layer (Figure 2.4) is the most commonly used neural network. The network trained with BP algorithm is capable of approximating any static function provided sufficiently representative input-output sample pairs are given. The latter also include generalized functions such as the Heaviside function. However, the disadvantage with an MLP network is the inherent difficulty of generalizing beyond the set of examples on which they were trained and thereby

they may make errors. A remedy for this deficiency is to construct an ensemble of neural nets, where several outputs to each input are combined in some fashion (e.g. simple averaging or stacking). This is in contrast to the traditional approach of choosing the best performing net. Bhatt & Helle (2001a) have demonstrated improved porosity and permeability predictions from well logs using ensemble combination of neural nets rather than selecting the single best by trial and error (Helle et al., 2001). Moreover, Helle & Bhatt (2001) successfully applied the ensemble approach to predict partial fluid saturation.

For multi-class classification problems such as discriminating between logfacies in a sequence, the use of a single MLP implies that one output neuron for each class would be required. Many output neurons, on the other hand, require a corresponding number of units in the hidden layer which, in turn, implies more network weights to be determined and a larger number of training patterns to satisfy the increased network complexity. The idea here is to reduce the multi-class classification problem to a number of two-class classification tasks in order to maintain a simple architecture of the MLP with a minimum of hidden units. The initial multi-class classification problem is then, in turn, accomplished by constructing a modular neural network (MNN) using the simple MLP as the building block. Each component of the MNN, or group of MLPs, has been assigned the task of predicting a given logfacies, enabling the MNN to solve the multi-class classification problem by voting. The latter assumes that a training set is available consisting of sample patterns of known classes (Table 6. 1) and the MLP is to be trained to assign these (training) and other (test) samples to their respective classes. As shown by Anand et al. (1995) the modular approach significantly improves the training performance in classification tasks such as character and speech recognition compared with that of the non-modular approach. In a nonmodular network, conflicting signals from different output nodes retard learning. Modular learning is more efficient since weight modification is guided by only one output. However, whereas the training errors were slightly reduced, the errors when applied to test data remained the same, indicating that the generalisation performance was not improved by modularity alone.

Generalisation performance, on the other hand, can be greatly improved by creating ensembles of the MLPs. The main motivation for combining nets in redundant ensembles is to guard against the failure of individual components. Combining a set of imperfect estimators is a way of managing the recognised limitation of the individual estimators; each component net is known to make errors but they are combined in such a way as to minimise the effect of these errors. Since we break down the multi-class classification problem to a two-class task using simple MLPs, the training and combination of the MLPs in a modular system is also greatly simplified.

To link these ideas to the current terminology in neural science we follow Sharkey et al. (2000) and assign the term ensemble combination to a set of redundant networks. In a committee the component networks are redundant in that they all provide a solution to the same task. The networks are generated by varying the initial weights. Although the committee normally achieves a better result, any one of the individual members could be used on its own to provide a solution to the task. By contrast, under a modular approach, a task is decomposed into a number of sub-tasks and the task solution requires contribution from all the several modules.

In the present study we improve generalisation by exploiting the ensemble as well as the modular combination; i.e. each module is dedicated to a given logfacies and the modules in turn are combined to classify all logfacies in the problem at hand. Moreover, the module for a given logfacies is itself a committee where a number of MLPs are redundantly combined.

Log type	Facies 1		Facies 2		Facies 3	
	Mean	Range	Mean	Range	Mean	Range
$\rho$ (g/cm <sup>3</sup> )	2.382	0.040	2.391	0.046	2.401	0.042
$\Delta t$ ( $\mu$ s/ft)	86.852	1.790	86.495	1.463	86.142	1.434
$\gamma$ (GAPI)	97.357	10.35	98.654	11.63	99.933	11.72
$R_t$ (ohm-m)	3.283	0.146	3.315	0.147	3.348	0.124
$Nphi$ (fract)	0.219	0.022	0.214	0.023	0.209	0.020

Table 6. 1: Mean values and range of noise for the three log-facies used for synthetic well logs.

### 6.3 Recurrent networks for enhanced layer detection

Recurrent neural networks have been widely used as time-series forecasters. Typical examples of this approach are market predictions, meteorological and traffic forecasting. Time series are generally sequences of measurements of one or more variables of an underlying dynamic system, whose state changes with time as a function of its current state vector  $\mathbf{x}(t)$  such that  $\mathbf{x}(t+1)=F(\mathbf{x}(t))$ . The problem is normally to predict the future based on the history; i.e.  $\mathbf{x}(t+d)=F(\mathbf{x}(t), \mathbf{x}(t-1), \dots, \mathbf{x}(t-N+1))$  where  $\mathbf{x}(t+d)$  is the future state predicted from present time  $t$  based on  $N$  time steps back. Such dynamic system may evolve over time, or space, to a set of points that have a

regular and smooth appearance. However, in a real-world system such as the stock market in time, and geophysical logs in space, the nature of the state space is obscure; so that the actual variables that contribute to  $\mathbf{x}(t)$  are unknown or noisy. The task of recurrent prediction can thus be rephrased; given measurements of one component of the state vector of a dynamic system is it possible to reconstruct the chaotic dynamics of the phase space and thereby predict the evolution of the measured variable? According to the embedding theorem of Mañé & Takens (see Abarbanel et al., 1993) this is fully possible if the size  $N$  of the lagged vector, i.e. the embedding dimension, is sufficiently large. Davey et al. (1999) demonstrate that using a recurrent BPANN the number of false nearest neighbour predictions  $\mathbf{x}(t+1)$  drops from 77% to 0.3% by increasing from  $N=2$  to  $N=4$ , but further increase of  $N$  does not reduce the error rate.

Well logs are sequential in distance along the well bore that intersects the geological layering at various angles. While exploration wells may be drilled normal to bedding, modern production wells are often sub-horizontal and intersect the layer boundaries at low angles. As a result the thickness of lithofacies will vary depending on the well trajectory. Thus the interpretation of lithofacies from well logs implies detection of changes while crossing the layer boundaries as well as identifying the facies from their characteristic log responses. However in order to compose a layer from the log readings, the thickness must be assumed greater than the spatial sampling i.e. layers of thickness less than two log samples should be rejected while layers of greater thickness should be enhanced. This property of finite layering will be exploited as prior data in a layer enhancement using the recurrent network shown in Figure 6.1. The mode of training and operation is a two-step procedure:

Let  $\mathbf{x}(z)$  be the log at time  $t$  at depth  $z$  and  $h$  is the output of BPANN after prediction. Then firstly, from a single BPANN (Figure 2.4) trained for a predefined logfacies  $s$  to give  $h=1$  (true) if the log measurements  $\mathbf{x}$  are within the actual logfacies  $s$ , and  $h=0$  (false) if  $\mathbf{x}$  falls outside, we generate depth-shifted logs  $\mathbf{x}(z-p), \dots, \mathbf{x}(z), \dots, \mathbf{x}(z+q)$  and predict the outputs  $h(z-p), \dots, h(z), \dots, h(z+q)$  from the BPANN. The outputs from BPANN fall in the interval  $h \in (0,1)$  when applied to training and test data.

Secondly, we establish the recurrent network (RBPANN) similar to the first BPANN (Figure 2.4) but with additional input neurons. The additional inputs are predictions  $h(z)$ ,  $h(z-2)$ ,  $h(z-1)$  and  $h(z+1)$  where  $h(z)$  is the prediction from BPANN corresponding to depth  $z$ . The RBPANN is now trained against the values of  $h(z) \in (0,1)$ . Here we exploit the results of Davey et al. (1999) and set the size  $N=4$  for the lag vector for  $\tilde{h}(z)$ , while keeping zero lag for the logs  $\mathbf{x}$ . The latter is based on the experimental results with synthetic logs as discussed below. Also notice that we use “future” predictions  $h(z+q)$  for input to

prediction of the “present” value  $\tilde{h}(z)$ . The number of input neurons has been identified by using synthetic data. The output from RBPANN is denoted by  $\tilde{h}(z)$ .

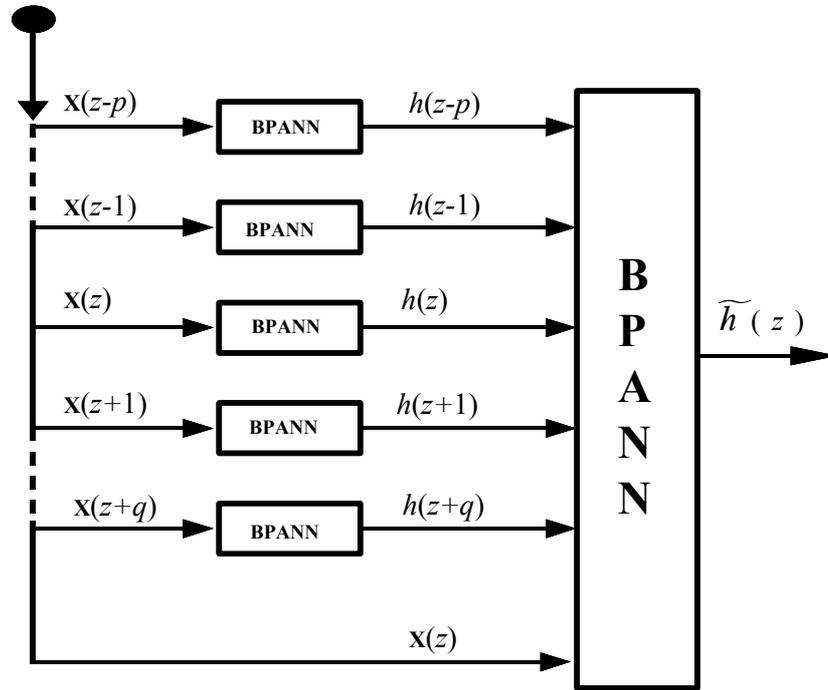


Figure 6.1: MNN architecture for lithofacies prediction.

In Figure 6.2 we show the combination of the BPANN and RBPANN (Figure 6.1) imbedded into a committee consisting of  $M$  members individually trained to detect a specific logfacies. The output from a CM is denoted by  $\bar{\tilde{h}}(z)$ . While the RBPANNs enhance the layer continuity and reject isolated miss predictions, the redundant combination of component networks ensures better generalisation. Yang et al. (1996) applied a similar approach to classify a suite of six different carbonate lithofacies from well logs.

The final MNN network architecture is shown in Figure 6.3 where each logfacies  $s=1, K$  has been dedicated to a committee machine CM- $s$  (Figure 6.2). Further layer enhancement has been introduced by the RBPANN at the output from each CM before the logfacies are presented to the combiner. The output from a RBPANN is denoted by  $\bar{\bar{\tilde{h}}}(z)$ . These outputs are then fed into the

combiner, which finally produces the results giving presence of one lithofacies at a time.

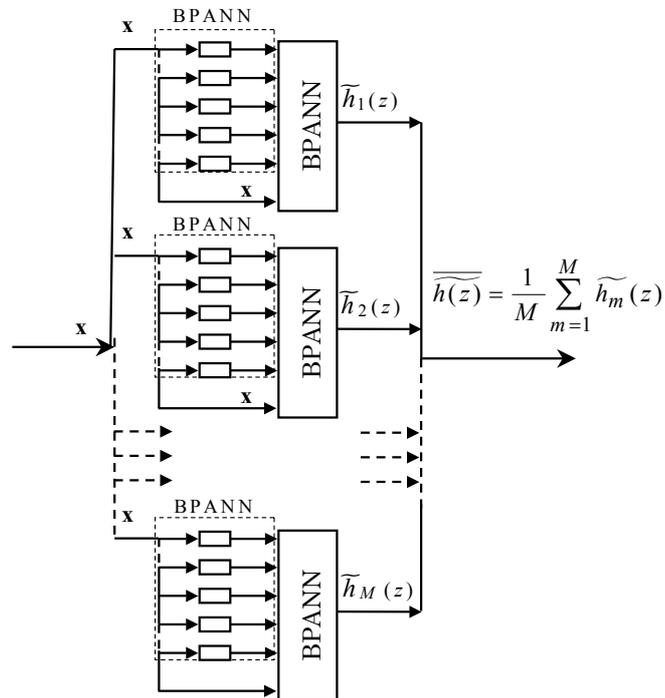


Figure 6.2: Architecture of CM for combination of  $M$  modular networks

## 6.4 Optimal design and training of networks for classification

Designing an optimal architecture of the network with respect to the number of training patterns, selection of training patterns, number of hidden neurons, training strategy, architecture of CM etc. is equally important for pattern recognition as for function approximation.

Determining the members of an ensemble is at least as important as deciding how the output should be combined. Selecting from the best performing of several tested networks is generally recommended. What is needed is a set of networks that generalises well and makes small errors. However, where errors are made, it is important that all the networks in the ensemble do not share them. This leads to the concept of diversity to eliminate error dependency. Randomising the initial network weights and varying the training patterns (boot-

strapping by replacement) are two popular methods for creating diversity to improve reliability of the network.

Apart from the general guidelines in network design and data selection, the optimal design and training procedure can only be achieved by testing for the actual problem and data at hand. In order to maintain full control of the data we adopt the common practice of using synthetic data in development and testing of geophysical methods.

The synthetic data was created by varying only the matrix parameters keeping the fluid parameters constant to identify only the lithological variations and not the fluid effects. We generated 8 models with the reduction in contrast between the facies each time. So the model 1 has a first layer (sand) with log parameters as  $\rho_b=2.157$ ,  $\Delta t_b=96.6$ , gamma ray = 20 and  $\phi_N=0.34$ ,  $R_t=2.97$ , a third layer (shale) with log parameters as  $\rho_b=2.63$ ,  $\Delta t_b=78.6$ , gamma ray = 140 and  $\phi_N=0.09$ ,  $R_t=4.47$  and a second layer (shaly sand) with the log parameters in between the two. Considering the contrast between first and third layer in this model to be 100 % then in model 8 the contrast between first and third layer is reduced to 2.5% and half of it in between the first and the second and the second and third layer. The identification of lithofacies when log data has higher contrast was trivial but the challenge lies in the typical range of variations within a North Sea reservoir at Jurassic level (Brent Group). The underlying model is presented in Table 6. 1 and the five synthetic logs, which are commonly available from the wireline logging; i.e. density, sonic, gamma, resistivity and neutron porosity are shown in Figure 6.4 and Figure 6.5. The small numerical separation between the facies is evident in the figures. From the statistical distributions of the log values, for each facies, shown in Figure 6.5 we see significant overlap but still there is reasonable separation between the cluster centres except for those of the gamma log. From 3000 samples, at a nominal sampling distance of 10 cm, we have selected at regular intervals a number of samples for training patterns for the various tests. A typical selection of the training patterns is shown in Figure 6.4 with 60 depth points for the entire log, or 20 patterns per logfacies. For testing the network performance we use all 3000 samples.

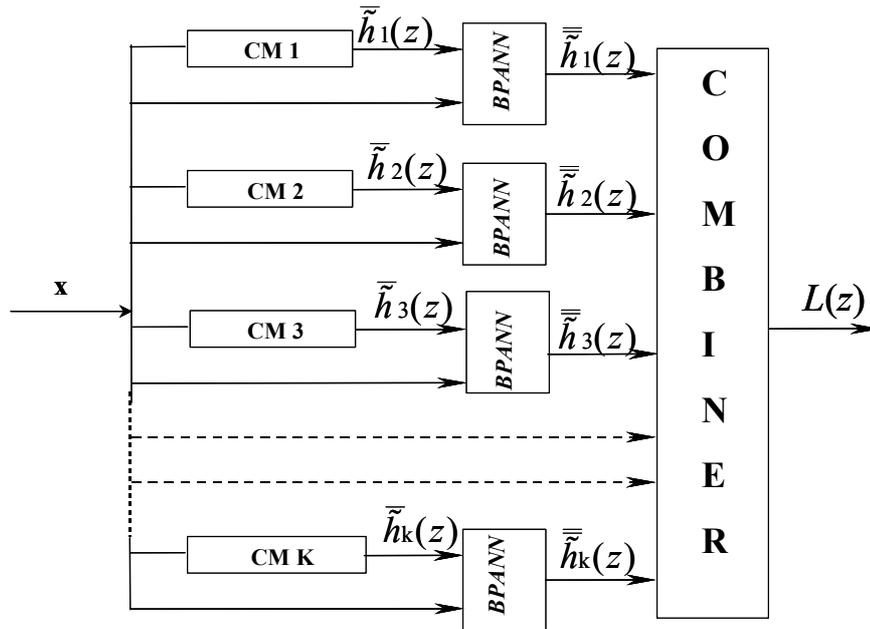


Figure 6.3: Architecture of MNN for predicting  $K$  log-facies

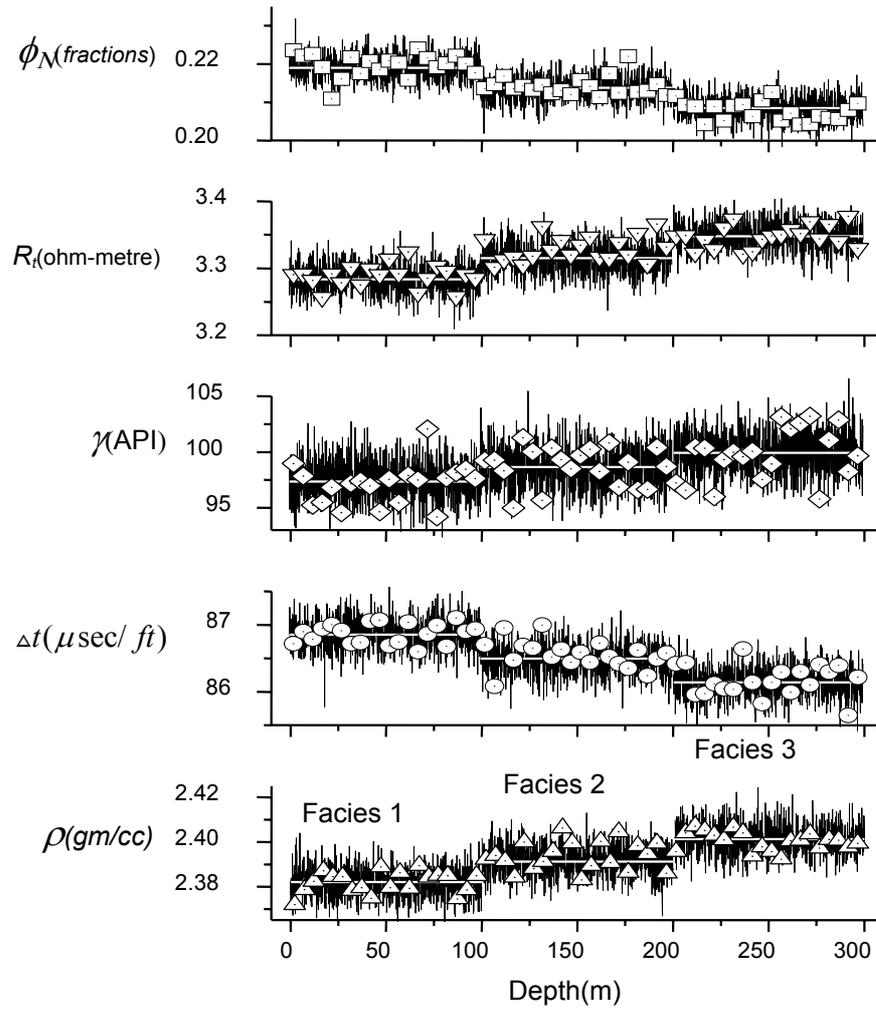


Figure 6.4: Log values and training patterns for 3-layer model with 10% noise

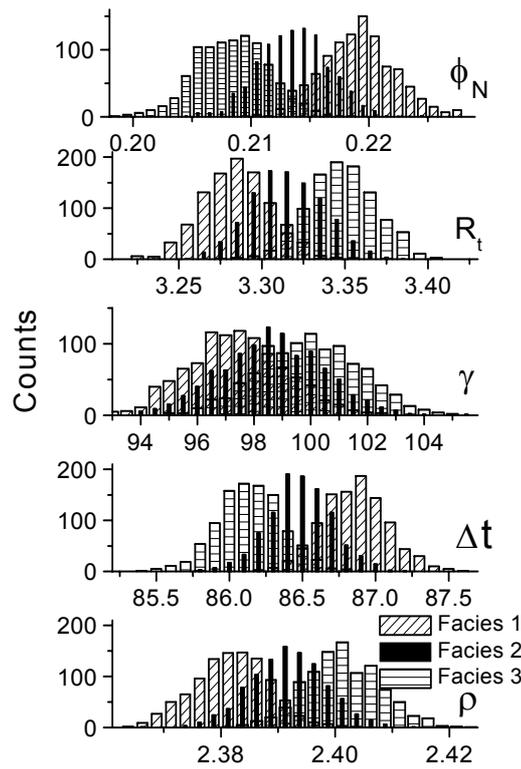


Figure 6.5: Distribution of log values for 3-facies model with 10% noise

### 6.4.1 Back propagation neural network for classification

For this problem of pattern recognition none of the training methods i.e. over-training or cross validation is ideal while the validation training stops too early with resulting large bias, heavy over training force the outputs to lie near 0 and 1, with only a small number of points distributed in the intermediate range. The latter implies that ensemble averaging may increase rather than reduce erroneous classification. Hence we used soft over-training approach to preserve a certain degree of variance in the output. The results shown in Figure 6.6 are based on a two-facies model with Facies 1 and 3 (Figure 6.4) using weakly over-trained neural networks.

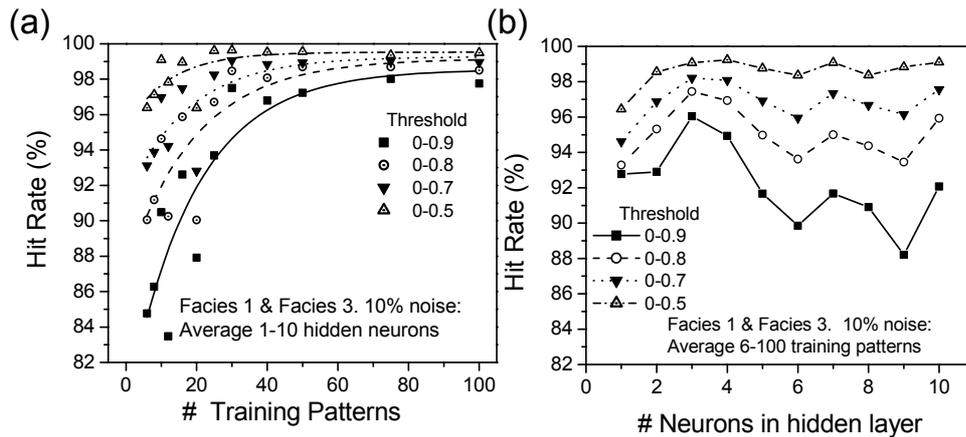


Figure 6.6: Performance of BPANN for 2-class classification problem applied to a model consisting of two formations. Hit rate versus number of (a) training patterns (b) number of neurons in the hidden layer

Hit rates versus training patterns, for a range of thresholds as shown in Figure 6.6a reveal that a saturation level of 98-99% is attained for all thresholds when the number of facts approaches 100. However, if the common threshold of 0.5 in a 2-class classification is applied, we achieve the same hit rate by employing less than 20 patterns, or 10 per facies. Hit rates versus number of neurons in the hidden layer (Figure 6.6b), on the other hand, display a less monotonic behaviour and there is a clear indication of a maximum at 3-4 neurons.

### 6.4.2 Recurrent back propagation networks for classification

Throughout the following exercises we apply the above results using networks with 4 hidden neurons and restricted to 20 patterns per facies. Now, the remaining parameters to tune are those of the recurrent network (Figure 6.1). Here the inputs to the RBPANN are the logs plus the lagged output from the BPANN. The problem is to select the number of lagged predictions  $h$  and the number of lagged logs  $x$  and we performed the exercise shown in Figure 6.7. The bottom graph shows the hit rate versus error for the initial predictions by the BPANN with five input logs, four hidden units and one output (5-4-1). Notice that in this case we are aiming at predicting Facies 2 located between Facies 1 and 3 (Figure 6.4) and with only 50% of the contrasts relative to the adjacent layers compared with that of the experiments shown in Figure 6.6. The error histograms are given for two different bins; a coarse bin of 0.25 to illustrate the

overall features and a fine bin of 0.02 to reveal the details of the distribution. In the fine-grain bin the two peaks at error 0 and 1 represent the absolute true and absolute false predictions, respectively. Whereas the peak around 0 contains about 85 % (2550) of the data, the peak at 1 contains only 7% (223). Using a threshold of 0.5 the overall hit rate is 90%. The aim is to eliminate the isolated error peak at 1, and to move the distribution of errors as close to zero as possible.

By introducing the recurrent back propagation network RBPANN of configuration 8-4-1 with inputs  $h(z-1), h(z), h(z+1)$  from the BPANN and  $x(z)$  from the logs we increase the overall hit rate from 90 to 93%. By further increasing the lag vector by adding the prediction  $h(z-2)$  to the input layer, the overall hit rate of the resulting 9-4-1 RBPANN increased to 96%. In the attempt to add the lagged logs  $x(z-1)$ , no improvement in the overall hit rate (95%) was achieved by the significant extra cost of using the 14-4-1 network.

Results of testing the individual RBPANNs for the three logfacies (Figure 6.4) are shown in Figure 6.8. The outputs  $h$  of the BPANNs are reasonably successful but with some scatter and misclassifications. Further enhancement of the hit rate after employing the RBPANN is clearly seen, although a few points are still incorrectly positioned and need to be fixed.

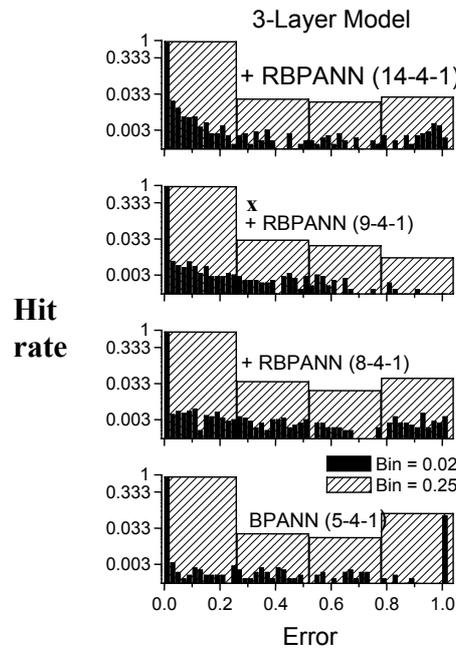


Figure 6.7: Comparison of hit-rate for layer 2 versus error distribution for BPANN and RBPANN for various embedding dimension and input combination

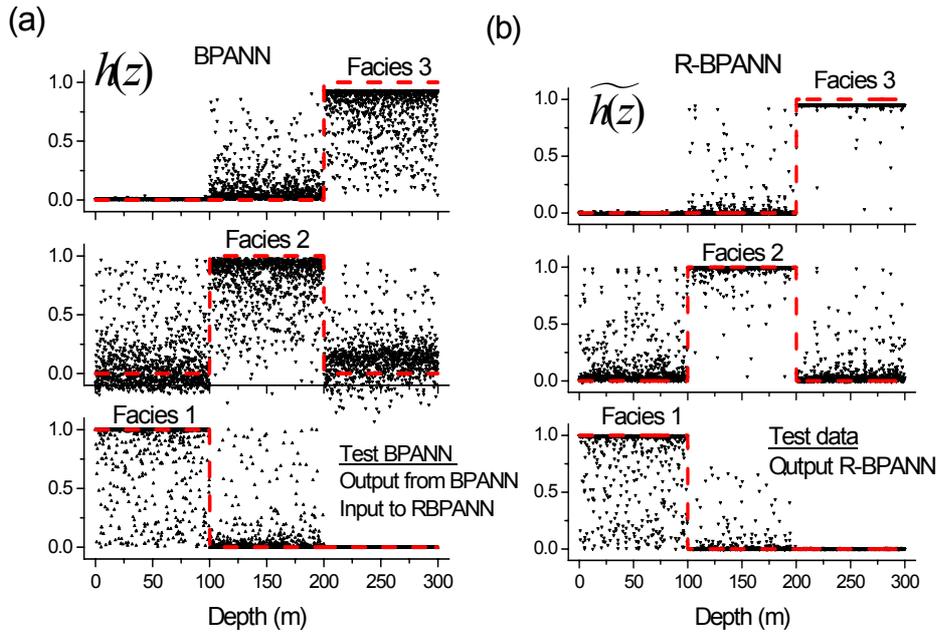


Figure 6.8: Test of network performance on the 3-layer model: (a) Output  $h(z)$  from BPANN and (b) output  $\tilde{h}(z)$  from RBPANN using nine inputs

### 6.4.3 Ensemble network and stacked generalisation

The simple way to grow an ensemble BPANN is to vary the set of initial random weights; i.e. a set of networks can be created by varying the initial random weights from which each network is trained, whilst holding the training data constant. This is a popular approach since in many real situations training samples are few because they are expensive or difficult to obtain.

Shortage of training data is also the motivation for the bagging approach. In bagging (Breiman, 1996) a training set containing  $N$  cases is perturbed by sampling with replacement (bootstrap)  $N$  times from the training set. This procedure can be repeated several times to create a number of different, although overlapping, data sets. A similar method is to use disjoint or mutually exclusive training sets; i.e. sampling without replacement (Sharkey et al., 1996). There is then no overlap between the data used to train different networks and diversity thus guaranteed. Shortage of data, on the other hand, may lead to reduced size of each data set and hence this may lead to deterioration in performance.

Since we may generate any number of patterns in the synthetic example, both disjoint training patterns and randomising weights are applied in this exercise. Also in the case of real well logs sufficient data are normally available

to select disjoint sets. Logfacies may span a significant number of samples, or data may be selected from a number of different wells. On the other hand, if shortage of data is the problem, a combination of bagging and disjoint sampling may be applied.

A number  $M=9$  disjoint training sets were created by sampling from the  $M$  next neighbour log samples (Figure 6.4), each containing 60 samples, or 20 per facies. The total training data thus consists of 540 patterns selected from a total of 3000 points. The  $M$  networks of the architecture Figure 6.1 were trained on the  $M$  disjoint data sets and redundantly combined into the committee machine Figure 6.2, with one CM for each facies. The resulting mis-hit rate for each component network in the committee and the ensemble average, for each CM, can be appreciated in Figure 6.9. Compared to the hit rate of individual component networks the ensemble classification is dramatically improved with mis-hit rates well below 1 %. The committee classification and the final output from the modular network (Figure 6.2) are shown in Figure 6.10. Again, the RBPANN at the final output significantly enhances the performance except for a few points. From a mis-hit rate of 6-9 % by the initial BPANN we have by application of the recurrent network RBPANN and the use of the ensembles effectively reduced the mis-classification rate of the MNN to less than 0.1% (Table 6.2).

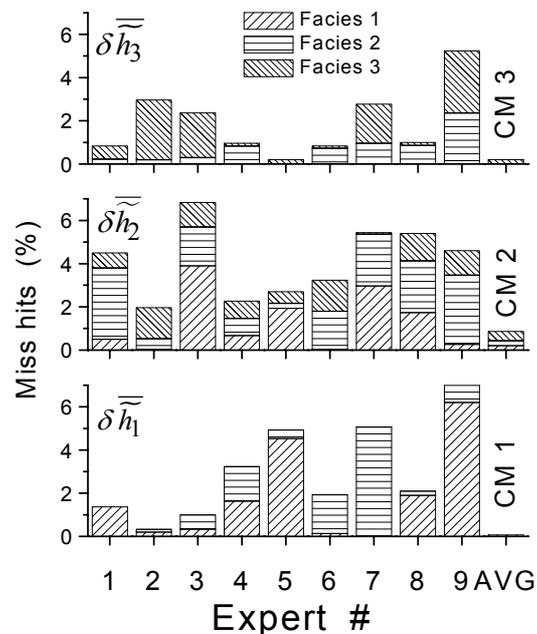


Figure 6.9: Distribution of miss-hits for the three committees CM1,CM2 and CM3 and their component networks for predicting the log-facies

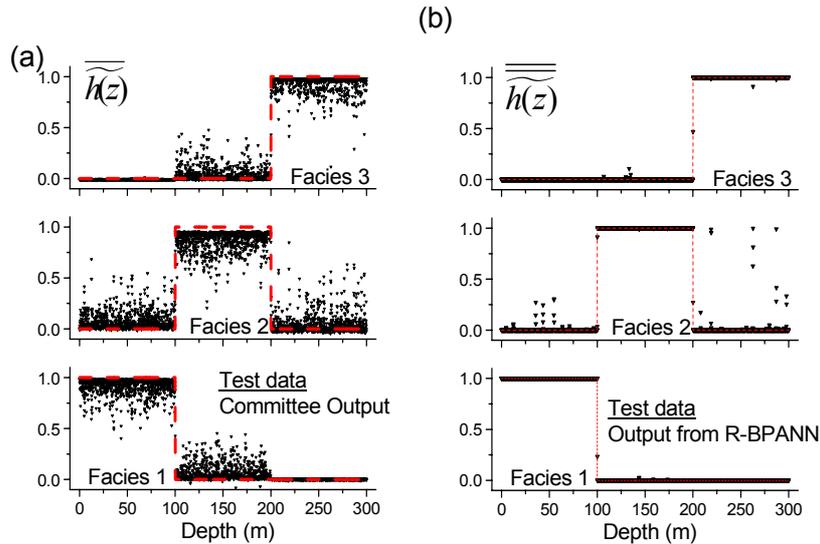


Figure 6.10: Test of network performance on the 3-layer model for the output  $\overline{h(z)}$  (a) from the committee consisting of 9 experts and the final output  $\overline{\overline{h(z)}}$  (b) after application of RBPANN

## 6.5 Classification of logfacies in a multi-layer model with thin layers

While still exploiting the synthetic data the idea here is to construct a multi-layer model based on the three log-facies, together with the corresponding set of networks already trained for their classification. In particular we want to evaluate the performance of the recurrent networks on various layer thicknesses and contrasts. A model consisting of 17 layers, with thicknesses varying from 3 to 55 meters (30-550 samples), was produced by splitting the initial three-layer model (Figure 6.4) into finer sections as shown in Figure 6.11. The networks trained on synthetic data were applied to the multilayer model. The classification of the layers by the CMs is shown in Figure 6.12a and the corresponding output from the MNN is shown in Figure 6.12b. Again, the three logfacies are well separated and correctly clustered with only a few mis-hits dominated, as expected, by the misclassification of the Facies 2 (0.5%).

The summary in Table 6.2 confirms that fine layering is more of a challenge to the networks than the coarse layer model, with slightly more mis-hits for the 17-layer model than for the 3-layer model. However, for any practical application the results for the multi-layer model are quite satisfactory. We now apply the above techniques to real data.

Facies	BPANN 3-layer	RBPANN 3-layer	CM 3-layer	MNN 3-layer	CM Multi layer	MNN Multi layer
Facies 1	5.73 (172)	4.93 (148)	0.07 (2)	0.0 (0)	0.26 (8)	0.30 (9)
Facies 2	8.63 (259)	2.70 (81)	0.87 (26)	0.2 (7)	0.93 (28)	0.5 (15)
Facies 3	5.90 (177)	1.00 (30)	0.2 (6)	0.03 (1)	0.2 (6)	0.13 (4)
Total	6.74 (608)	2.89 (259)	0.38 (34)	0.09 (8)	0.46 (42)	0.31 (28)

Table 6.2: Mis-classifications (in % of 3000 samples) for the various classification steps for the 3-layer model (Figure 6.4) and the multi-layer model (Figure 6.11). Number of mis-hits in parentheses.

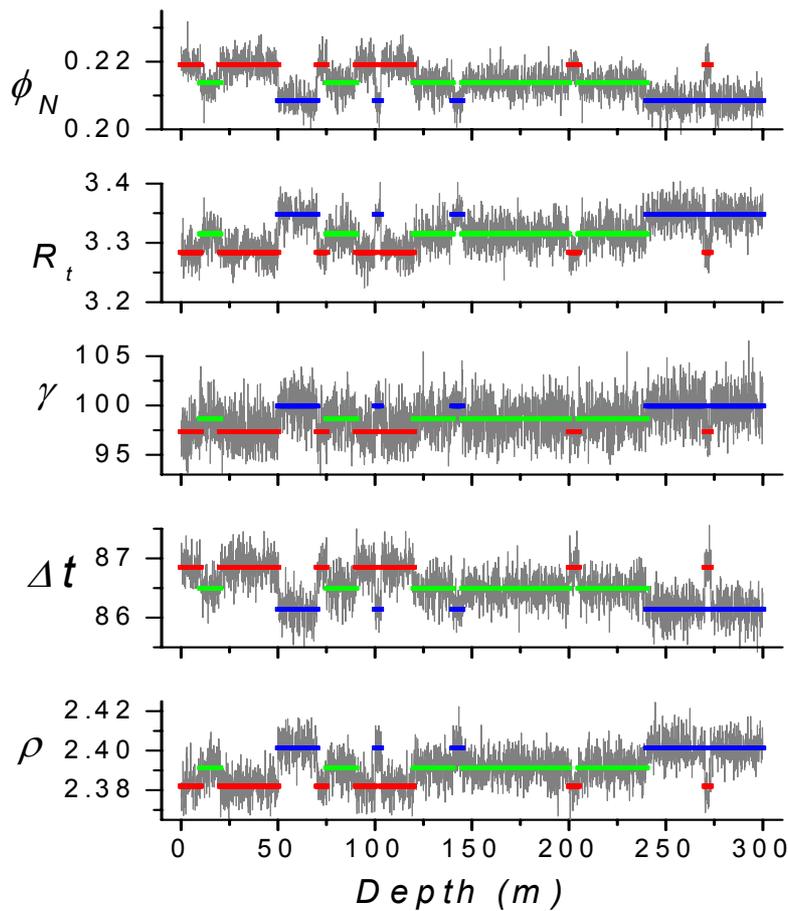


Figure 6.11: Synthetic logs for a multi-layer model (different colours denote different layers)

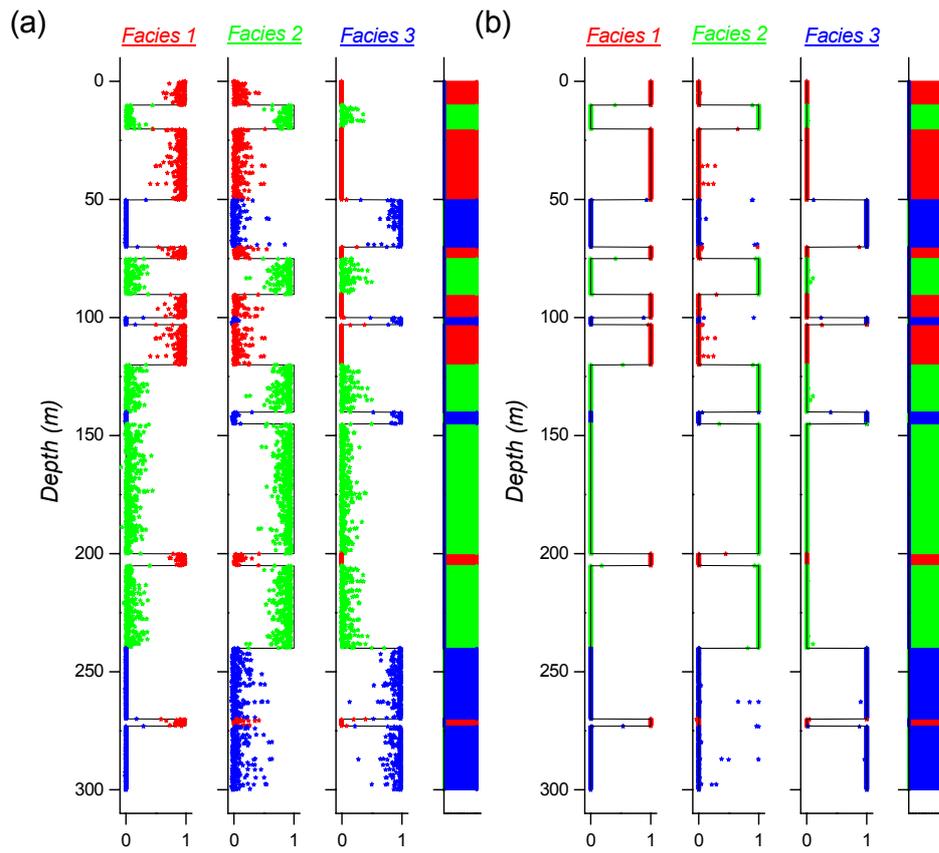


Figure 6.12: Predictions for the 17-layer model (a) output from the CMs (b) consecutive application of the RBPANN on CM output

## 6.6 Real data

We consider the Ness Formation of the Oseberg Field in the North Sea, confined to the stratigraphic interval between the Tarbert and the Etive-Rannoch formations of the Brent group (Figure 6.13). It is characterised by a relatively low proportion of good quality reservoir rocks. It is much more heterogeneous than the Oseberg formation and comprises sandstones, mudrocks and coal beds accumulated in deltaic plain or coastal plain environments. Fluvial channel sandstones of variable thicknesses and orientations and having laterally restricted geometries make up the reservoir geology. The formation is interpreted to reflect deposition within a fluvio-lacustrine, upper delta plain environment where the fluvial channel systems represent the main conduits for sediment transport and deposition (Ryseth, 1989). The distinction between reservoir and nonreservoir lithologies grouped the formation into two main facies assemblages. Various types of fluvial channel sandstones are contained within assemblage 1 (channel fill) and the surrounding delta plain deposits including coal beds, lake, levee and crevasse deposits are grouped into assemblage 2 (flood plain).

Studies of the alluvial architecture of the Ness Formation (Ryseth et al., 1998) have shown that the stacking density of channel sandstones varies with the stratigraphic level; i.e. with relatively thick and laterally persistent bodies near the base of the formation. Above this lower fluvial reservoir, the remaining lower Ness is dominated by coal-bearing floodplain deposits with isolated fluvial sandstone bodies of both simple and multi-storey character. In the upper part of the Ness Formation, numerous fluvial, multi-storey sandstone bodies interfinger with contemporary floodplain deposits, and the whole succession culminates in a coal-bearing unit underlying the Tarbert Formation.

The four main lithofacies within the formation identified by geologists on the cores: (1) channel sands, (2) crevasses, (3) lake and (4) coals have detectable differences in the log response. Facies (1) is relatively clean poorly to well-sorted sand characterised by low values of gamma, density and velocity, the resistivity and neutron porosity are more variable, depending on the type of pore fluid. Average porosity and permeability of the channel sands are 26% and 1700 mD, respectively. Thickness of the channel deposits varies in the range 2-25 m with a maximum in the frequency distribution around 5 m.

Facies (2) is a material consisting of fine-grained sand inter-bedded with clay-rich material. The log responses of the crevasse are more variable than in the channel sands, depending on the clay content and pore fluid. In general, however, the log readings lie between those of the channel sands and the shaley lake facies but with considerable overlap (Figure 6.14). The reservoir quality of the crevasses is less favourable and the transition between the adjacent sand channels and the floodplain lakes are gradual in properties. Studies on the

thickness distribution in the area are not available, but the wells included in this study indicate a layer thickness in the range 2-10 m.

The floodplain lake (3) is the dominating facies within the Ness Formation as seen from Figure 6.13 and Figure 6.15. The log characteristics are those of Jurassic mudrocks; i.e. high gamma, density and velocity, and generally low resistivity.

Coal layers (4) are in general characterised by low density and sonic velocity with distinct peak at high values of the neutron porosity. However, some of the coal streaks apparent from the cores contain mixtures of other material and hence have a different response in the log readings as apparent from the wide range log values picked from layers nominally interpreted to be coal. The implications of the latter are evident in lower section of Figure 6.13 where distinct coal beds around 2675 m are not detected by the logs. Most of the coal beds occur in the lower half of the Ness Formation and also in the thin topmost part.

The distribution of log values for the four lithofacies within the Ness formation is shown in Figure 6.14. Gamma ray and density logs are the most significant in lithofacies identification. In gamma ray the cluster centres for channel sand, crevasse and lake are well separated, while those for coal and crevasses are overlapping. The latter is remedied by the density and sonic where the cluster centres for coal and crevasses are located in opposite ends of the range. In addition the response to coal is also quite distinct in neutron porosity where its cluster centre appears fairly isolated in the upper porosity range. In general, however, the log responses to coal covers a wide range of values, supporting our worries that layers which, by visual inspection are interpreted as coal, may be contaminated by other rock components. Except for the high-porosity channel sands where the resistivity is a strong function of the pore fluid type, the other facies are clustered in the lower end of the resistivity scale. For the water filled sands the resistivity cluster is at the lower end, while for the hydrocarbon bearing sand clustering is confined to the high-resistivity range.

A comparison between zonation based on the neural networks and cores for four wells is shown in Figure 6.15, and the classification performance is summarised in Table 6.3. Training patterns are mainly extracted from well 30/6-C-12, while the other three wells are essentially unknown to the network. Therefore, the mis-hits for the training well 30/6-C-12 are at minimum (4.7%) whereas a maximum of 10 % occurs in one of the unknown test wells 30/9-B-21. Facies-wise the maximum average mis-hits of 13.6 % occur for lake whereas the smallest average mis-hit rates are for the coal and crevasse of 2.8 and 3.0 %, respectively.

The network was capable of identifying all four lithofacies although there was overlapping between clusters. The network fails to identify thin lithofacies

less than 0.5 m thick, for example it missed the crevasse in well 30/9-B-21 at a depth of 2625m. The reason is there are only 4 samples within an interval of 0.1m from this facies i.e. the thickness of the facies is about 0.4 m, so the logging tools (for example resistivity, gamma ray and neutron porosity) cannot obtain an actual response from this bed and were more affected by the side beds that is the channel sand above and below this facies. The network has a limitation at few depths in identifying the correct boundaries of the facies also for example in well 30/9-B-21 (at 2667m), and in well 30/9-1 (at 2724m), because the log response is not unique at the boundaries. The reason is that different tools have different resolution, resulting in formation property averages representing different formation volumes. The network missed to identify coal layers in well 30/9-1 at a depth of 2703m and the reason is that the lithofacies identification from cores are done by visual inspection but at this depth there was not a coal response in the logs that is the unique feature of coal with low density, high resistivity and high neutron porosity. For this reason coal was misinterpreted to be a crevasse in well 30/9-B-21 at a depth of 2638m. Due to the overlapping clusters of crevasse and lake the network misinterpreted a lake to be a crevasse at the depth of 2708m in well 30/9-1. But overall the network identified the four lithofacies with an error of less than 10% in the four wells, which is significant.

It is of interest to compare the performance of our classification of real lithofacies with those of similar studies published in the current literature. Based on discriminant factor analysis Busch et al. (1987) classified 7 facies with an overall mis-hit rate of 25%. Using a similar technique Jian et al. (1994) classified 8 facies with an overall miss-hit rate of 23%. In the comparative study of Wong et al. (1995) 4 facies were classified with the same miss-hit rate of only 5% both for discriminant factor analysis and the neural network approach. Using a recurrent neural network Yang et al. (1996) obtained an overall mis-hit rate for 6 carbonate facies of 17%, whereas the mis-hit rate using conventional clustering analysis exceeded 40% for the same data set. By a hybrid neural network and expert system approach Chang et al. (2000) identified 4 facies with a mis-hit rate of 13%.

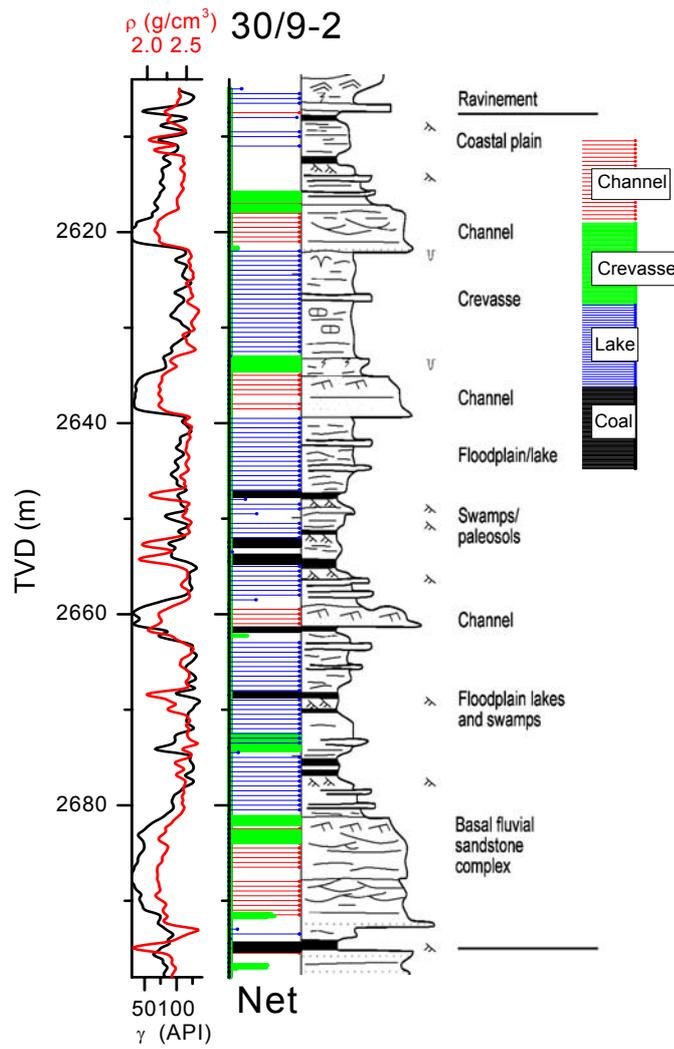


Figure 6.13: Detailed comparison of logfacies from present study (left) and lithofacies (right) after Ryseth et al. (1998).

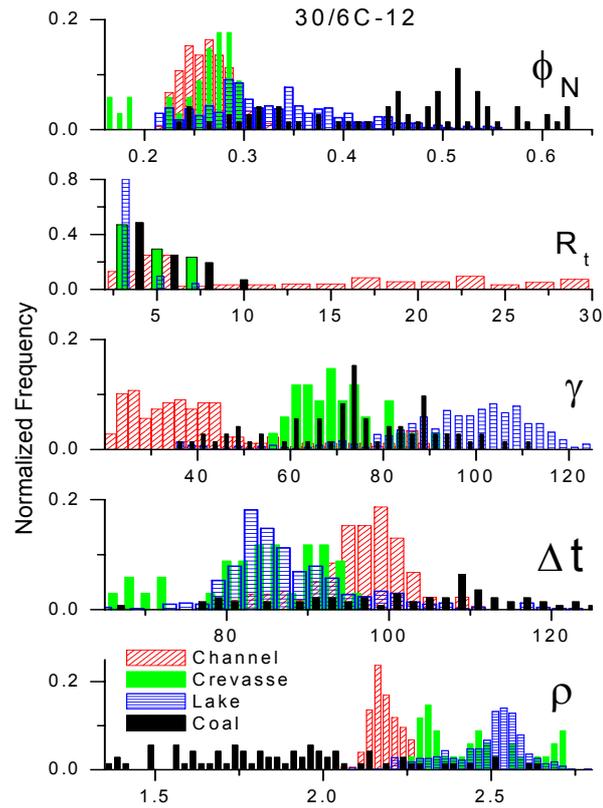


Figure 6.14: Distribution of log values for the four facies within the Ness formation of the well 30/6-C-12.

Lithofacies	30/6-C-12 Training well	30/9-1	30/9- 2	30/9-B-21	Average
Channel sand	3.6	1.3	9.1	12.3	6.6
Crevasse	1.1	4.3	4.0	2.5	3.0
Lake	11.3	16	16.5	11.3	13.6
Coal	2.8	3	2.7	6.7	3.8
Average	4.7	6.2	8.1	10.0	

Table 6.3: Performance of the MNN as mis-classification rates for the four facies of the Ness Formation (in % of total samples used).

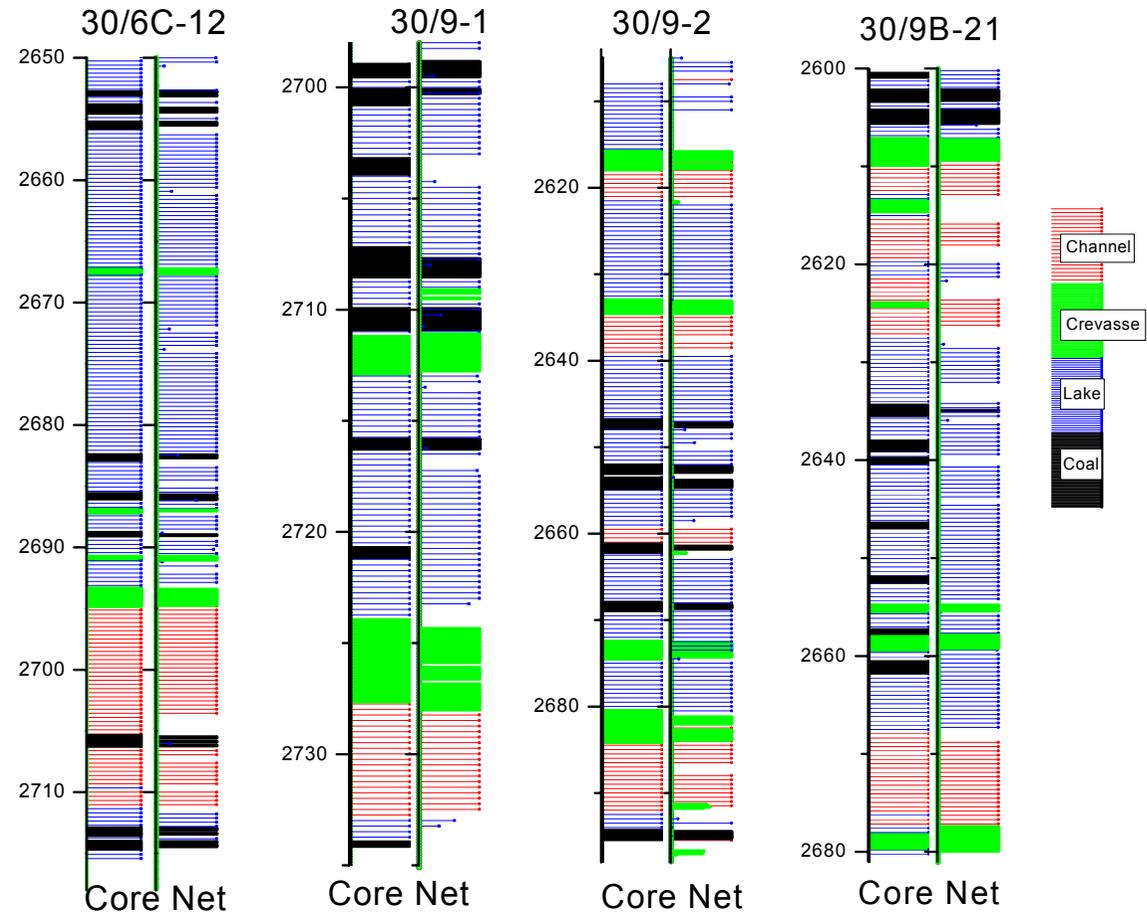


Figure 6.15: Comparison of log facies from the neural nets with the manual zonation based on cores.

## 6.7 Conclusions

In this study we have combined modern neural network techniques in an attempt to improve generalisation of the BPANN approach for objective numerical clustering and identification of facies from well logs. The method which is based on combining back-propagation neural networks in ensembles and modular systems, where the multi-class classification problem of facies identification has effectively been reduced to a number of two-class problems is very effective in pattern classification. The basic property of the facies that it consists of several sequential points along the well bore has been exploited using a recurrent BPANN adopted from time-series analysis. Ensembles of neural networks trained on disjoint sets of patterns using a soft over-training approach are essential components of the technique to ensure diversity and improve generalisation of the stack. We have optimised the individual building blocks of a modular system by using synthetic logs and applied the networks to facies zonation of the Ness Formation of the Brent Group in a selection of North Sea wells.

By using synthetic logs from a realistic model with small layer contrast and moderate noise level we find an excellent classification performance of slightly less than 100 % hit-rates. By introduction of fine-layering in the model we have shown that the performance is only slightly reduced, demonstrating excellent performance of the RBPANN for layer enhancement, also in case of thin layers.

Classification from real data is more challenging since the facies in the present study were initially defined by visual inspection of cores, and thus not fully compatible with the readings of the logging tools which detect different physical properties and with coarser spatial sampling. Application to the four facies of the Ness Formation reveals an average hit-rate well above 90% in wells unknown to the network. Compared with similar classification studies published our results reveal slightly to significantly better performance.

## Chapter 7

# Application of neural networks on measurement while drilling (MWD) logs

### 7.1 Introduction

Reservoir characterization while drilling is very important, as there are several decisions to be taken immediately while drilling without delaying rig time. So in this situation having a complete information of reservoir properties will ease the process of decision-making. In this chapter we have utilised our prior experiences of training networks for porosity, permeability and partial fluid saturation with wireline data and then generated networks based on MWD data, which can be used in real time.

For economical and technical reasons cores are normally not collected in highly deviated and horizontal wells. Cores for calibrating the petrophysical analysis of MWD data are thus not commonly available. On the other hand, a pilot well is often drilled where both wireline and MWD data have been acquired, allowing for a calibration of the MWD networks based on the formation properties established from the wireline data and calibrated neural nets for the actual field. In this study the patterns for training the MWD networks for porosity and permeability are based on the networks established for wireline data by Helle et al. (2001).

The neural networks for porosity, permeability, fluid identification and partial saturation using MWD logs have been trained and tested on data mainly from the Oseberg field in the North Sea and are shown to be sufficiently accurate to satisfy most practical needs. A major advantage is that the neural network approach requires no a priori knowledge of the grain material and pore fluid, and thus can be applied during drilling without prior petrophysical evaluation. Once the networks have been established the application requires a minimum of

computing time. However, the main drawback of the method is the amount of effort required to select a representative collection of training facts, which is common to most models relying on real data, and the time needed to train and test the network.

## 7.2 Sources of error in the data

While training MWD networks we use MWD, wireline and core data. Thus apart from the differences between core and log data as discussed before there are differences in MWD and wireline measurements also which may induce errors in porosity, permeability and water saturation prediction by neural network while using MWD networks. For example:

There is a difference in the environment during wireline logging and MWD. MWD is recorded while drilling therefore either mud filtrate invasion has not taken place or it is very less. Therefore the MWD logs read mainly the properties of uninvaded formation, whereas it is hard to eliminate the invasion effect during wireline logging even after environmental corrections.

The wireline and the MWD data may have same or different horizontal and vertical resolution depending upon the source receiver spacing and the frequency of the tool.

The horizontal depth of investigation is different for wireline and MWD data. The wireline data is recorded with low frequency therefore the signal has a greater depth of investigation. The MWD data is recorded with a high frequency signal therefore the tool measures closer to borehole.

The mechanical differences between the wireline and MWD density tools are significant. The MWD system uses a full gauge stabiliser for mud exclusion, which is completely effective only in smooth gauge holes whereas the wireline tools are padded and can make good contact with the borehole wall. Thus the MWD tools measure an average density around the borehole as opposed to the single track that the wireline system measures. These effects introduce important differences in the two measurements.

The effect of well bore condition on MWD neutron porosity measurement generally is greater than on a wireline compensated neutron log porosity measurement. This greater borehole dependence is because the MWD neutron porosity tool is nominally centered while the wireline tool is eccentric in the borehole.

### 7.3 Porosity prediction

Commonly neural networks are used to predict petrophysical properties by calibrating log data with respect to core data, as the cores have the measured values of the properties and the logs contain its geophysical information. But cores are rarely collected in horizontal wells. In this study we devised a technique of predicting porosity and permeability using MWD logs in horizontal and highly deviated wells. The calibration of the MWD networks rely on the formation properties established from the wireline data and the calibrated neural nets for the actual field. Thus training patterns for the MWD network are the output from wireline network of (Helle et al., 2001) in wells where both MWD and wireline data were acquired. This network uses density, resistivity and sonic logs in its input layer and the majority of the porosity values are based on grain density laboratory measurements and bulk density from wireline data (Lucas, 1998) which were carefully selected to obtain a range of values appropriate for most sediments in the Viking Graben (Bhatt, 1998) for use in a basin-scale fluid flow analysis. This network has been discussed in section 3.6 of the report.

The network for predicting porosity using MWD data consists of 3 input neurons corresponding to the density, neutron porosity and resistivity logs, 10 neurons in the hidden layer and one output neuron (Figure 2.4). In MWD networks we used neutron porosity logs in the absence of sonic logs as used in wireline networks. The optimal architecture of the network is designed using synthetic data as discussed in section 3.3. The training patterns were selected from water and hydrocarbon bearing intervals of the reservoir and as well as nonreservoir sections. The selected training patterns cover the porosity range from 0.02-0.32. We trained  $K=20$  redundant networks with the same input data, but with randomised initial weights. Out of those 20 networks we selected 9 networks based on the 'select and test approach' of Sharkey et al. 2001. The slightly different outputs from these networks are then combined using the ensemble average and OLC approach.

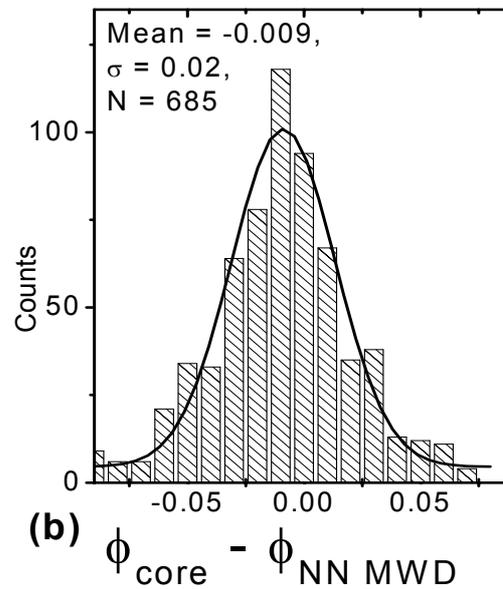
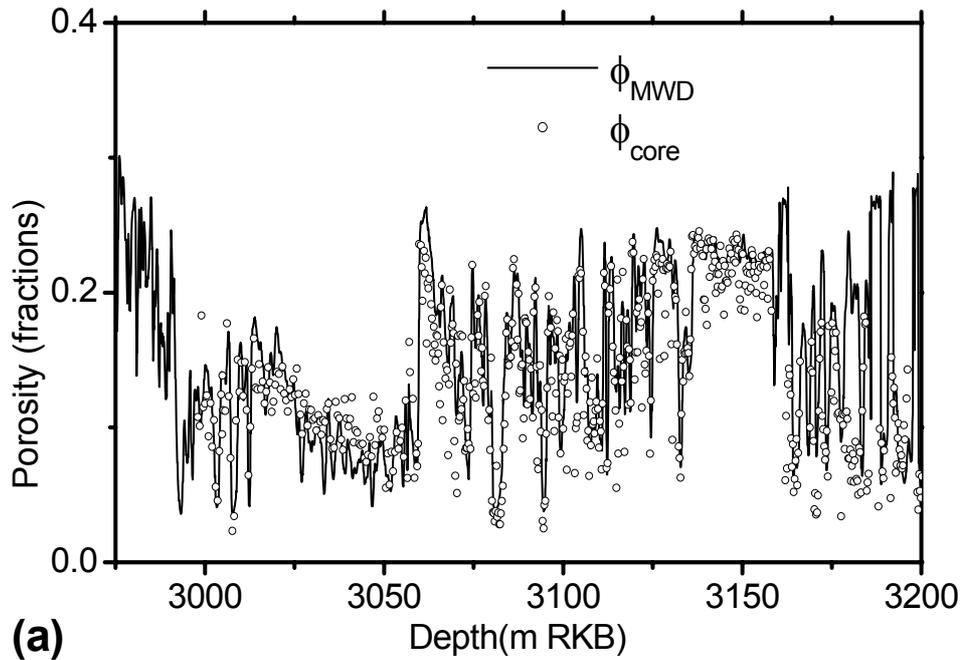


Figure 7.1: a) Comparison of porosity prediction using MWD and core data in well 30/9-14, b) error distribution.

The comparison of the porosity predicted by MWD network and core porosity in well 30/9-14 is shown in Figure 7.1a. The network gives good results. The difference between the core measured He porosity and the network predicted porosity gives a mean error of 0.009 and a standard deviation of 0.02 which is certainly low error. The results are comparable with the porosity predicted by wireline network using CM approach as shown in Figure 3.13 and 3.14 which is quite good because normally the results on MWD data are poorer than that of wireline data. The reason is that the log quality of MWD logs is not as good as that of wireline logs as they are more exposed to noise while drilling. The errors are now mainly due to the combination of different measuring techniques such as cores, wireline and MWD measurements.

## 7.4 Permeability prediction

In the absence of a sonic log the MWD permeability network has only 3 neurons in the input layer corresponding to density, neutron porosity and gamma ray logs unlike the wireline network (discussed in section 4.6), which consisted of 4 neurons in the input layer. Based on the experience from permeability synthetic data discussed in section 4.3, the optimal architecture of the network is 3 neurons in the input layer, 10 neurons in the hidden layer and 1 neuron in the output layer. The training patterns cover a logarithmic permeability range of micro Darcy to about 7 Darcy (-6.6 to 3.82 in logarithmic scale). As for the porosity network, the patterns for training the MWD networks are based on the predicted values established from the single network for wireline data by Helle et al. (2001).

The problems and errors involved in combining air permeability data at room conditions with log measurements at down hole conditions as discussed before in section 4.2 hold for this network also. Another problem with the permeability is its wide range of variation; i.e. by a few orders of magnitude within the reservoir. In order to improve the resolution of the network we split the range into several ( $N=3$ ) sub-ranges. We train ( $K=9$ ) networks for each sub range and combine them by simple averaging, so that we now get  $N=3$  experts for the three ranges of permeability. A gating network that can predict approximate values, in turn, will trigger the output from the appropriate expert where the individual experts are trained to accurately predict within a specified range. The architecture of permeability CM is shown in Figure 4.6. A comparison of permeability predicted by wireline network, MWD network and core measured permeabilities is shown in Figure 7.2. The difference between the logarithm of core permeability and the predicted permeability gives a mean error of 0.27 and a standard deviation of 0.33. The results obtained after testing MWD CM on real data in well 30/6-C-15 (Figure 7.2a) are comparable with the results from wireline CM as shown in 4.13 and 4.14. The errors are low enough and are mainly due to the differences in the spatial resolution, measurement conditions of

log and core data and the different types of fluid permeabilities measured by log and core data and due to the difference in MWD and wireline logs.

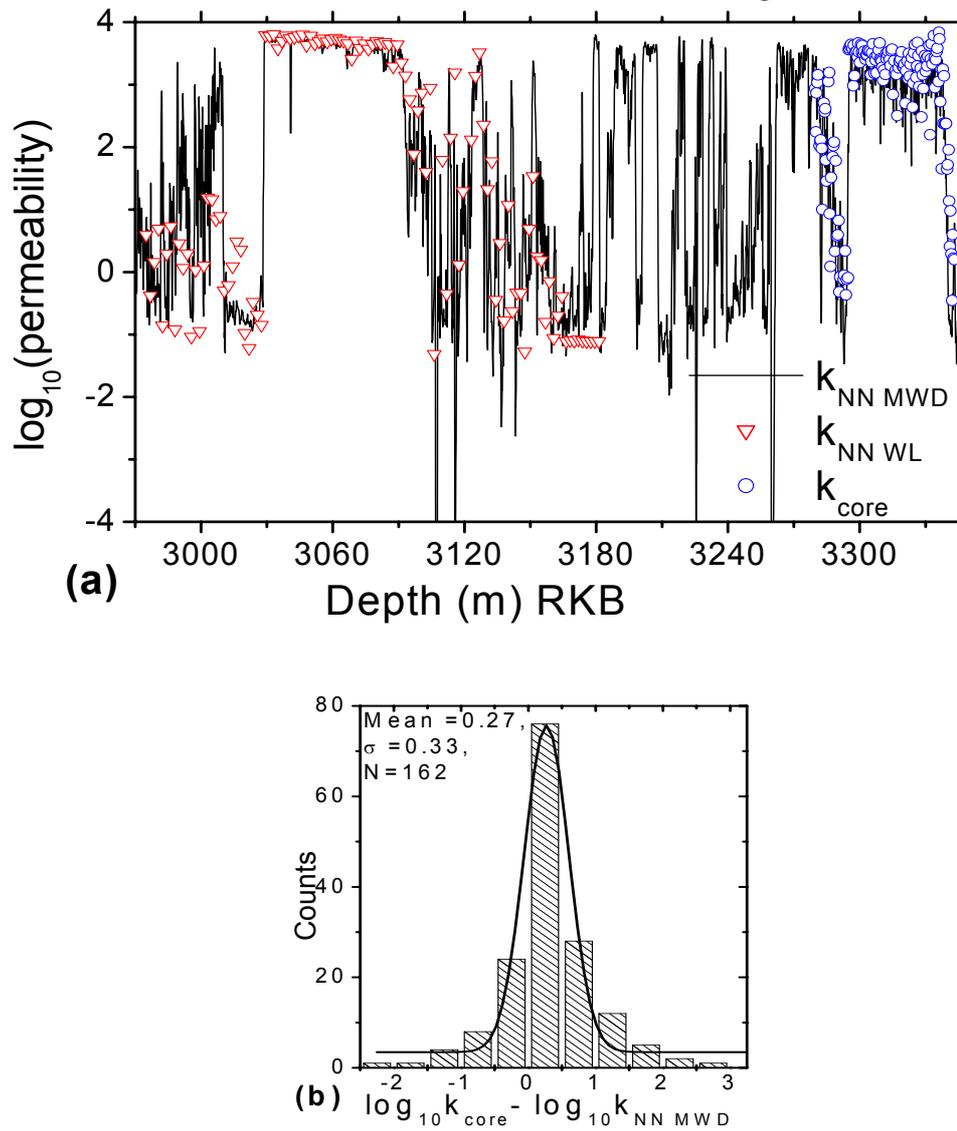


Figure 7.2: a) Comparison of permeability prediction using MWD, wireline(WL) and core data in well 30/6-C-15, b) error distribution.

## 7.5 Water saturation prediction

Based on the experience from synthetic data of partial fluid saturation prediction the basic architecture of individual network for predicting water saturation is as shown in Figure 2.4. It consists of 3 neurons in the input layer corresponding to density, neutron porosity and resistivity logs, 4 neurons in the hidden layer and 1 neuron in the output layer. The patterns are the water saturation values from CPI logs. The water saturation values in the CPI logs are based on the individual interpretation of a petrophysicist using the Indonesian equation (5.8) (Poupon et al., 1971). A CM approach has been applied for predicting water saturation using MWD logs. We trained ( $K=9$ ) experts and their output has been combined by the OLC method reducing the variance and the bias from the individual networks. The architecture of CM is as shown in Figure 3.11.

The MWD network trained by OLC method has been tested on several wells, Figure 7.3a shows a comparison between the water saturation predicted by the MWD network using the OLC approach and the corresponding  $S_w$  values from CPI logs in well 30/9-B-6. There are 7000 data points on which the prediction has been made. The data from this well is unknown to the network while training. There is an oil water contact (OWC) in this well at a depth of around 5880m and a transition zone from oil to water of around 20m thickness. As shown in Figure 7.3b the difference between the  $S_w$  values predicted by neural network and those from CPI logs give a mean error of 0.03 and a standard deviation of 0.03. Thus, as shown in Figure 7.3.c, there is an underprediction of  $S_w$  values in comparison with CPI data, the reason being that in the training dataset the  $S_w$  values are about 2-3 saturation units lower for the combination of logs corresponding to this well. Moreover, even by experienced petrophysicists it is not trivial to quantify exactly the  $S_w$  values because of the uncertainty in log data and the saturation parameters. Thus we believe that in view of the existing uncertainties the prediction of  $S_w$  by neural network is good. As shown in Figure 7.3c there is a good correlation between the  $S_w$  predicted by neural network and the one from CPI logs, in spite of the underprediction.

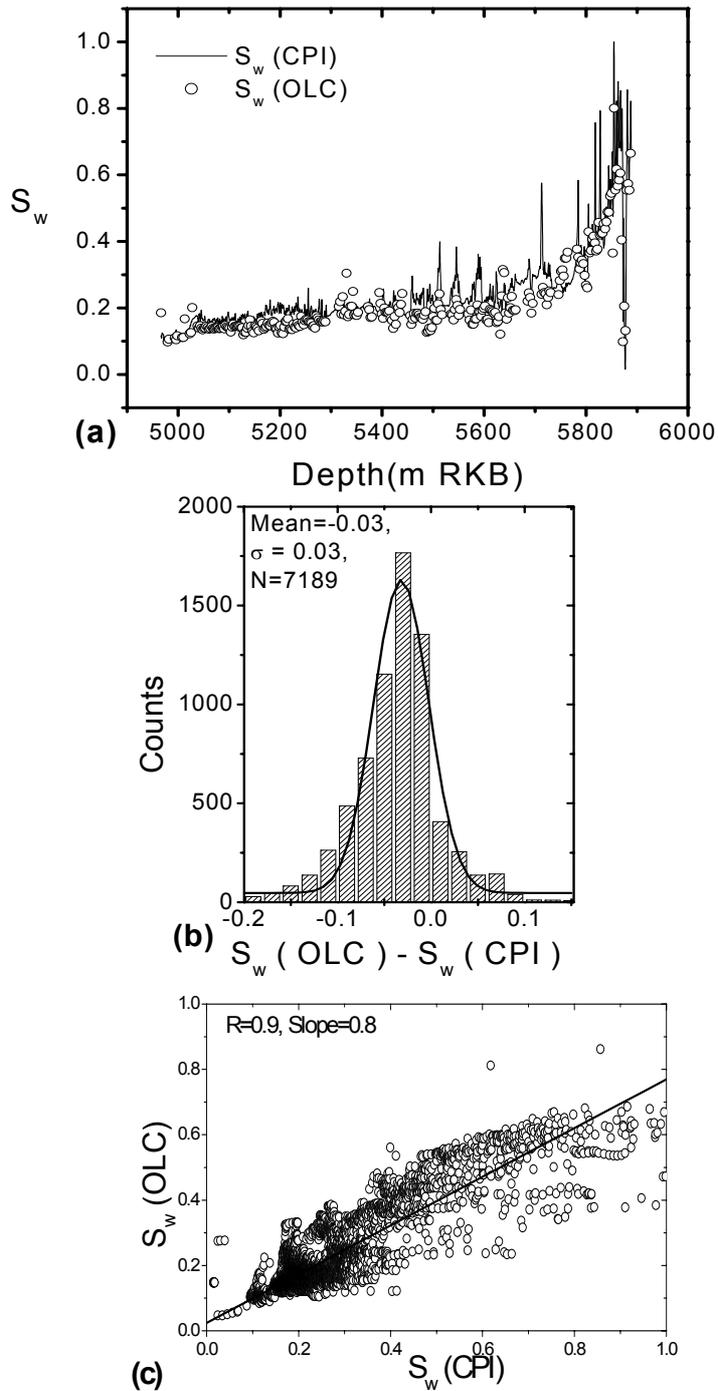


Figure 7. 3: a) Comparison of  $S_w$  by OLC approach and  $S_w$  from CPI logs in well 30/9-B-6, b) error distribution, c) cross plot of  $S_w$  from OLC approach and  $S_w$  from CPI logs.

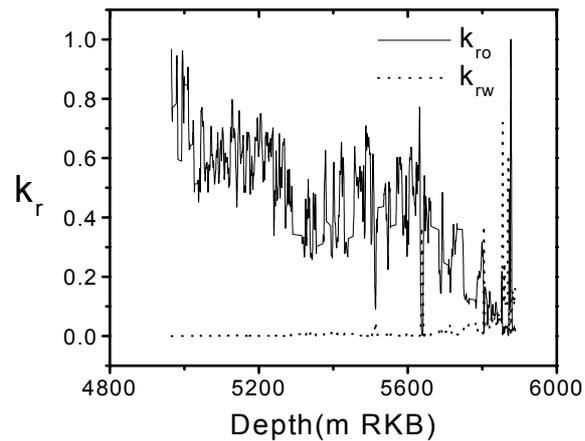


Figure 7.4: Normalised relative permeability logs of well 30/9-B-6.

## 7.6 Conclusions

The parallel architecture of the neural network in the committee machine approach combined with optimal linear combination and gating networks, significantly enhances the power of the neural network techniques for application in petroleum geoscience. With this approach prediction of porosity, permeability and water saturation from MWD data is feasible and thus can be made while drilling. The standard deviation errors in predicting porosity, permeability and water saturation from MWD data are of the same order as for wireline logs. As for wireline networks it is also important to consider the behavior of MWD networks in different well and formation fluids. A comparison of net to gross predictions by porosity and/or permeability networks with the net to gross results from standard CPI evaluations will represent a useful enhancement of the present study.



## Chapter 8

### Conclusions and future work

From the present study we have concluded that the neural network approach for predicting reservoir properties has many advantages over the conventional methods including semi-analytical expressions and empirical formulas based on linear regression method. The neural network provides a pragmatic solution to the problem of converting well logs to reservoir properties. The main drawback is the effort required in selecting representative training data that, on the other hand, is a common problem for all modelling methods relying upon real data. A committee machine is the best option for predicting reservoir properties from well logs because the optimisation of the network has no unique solution so minimising the error by a number of redundant networks is one of the solutions. The committee machine having modular neural networks is also beneficial because then the main task is distributed among a number of various small tasks and thus the accuracy in the results is maintained by keeping the architecture simple. The predictions made by neural networks on a test well will sometimes be too high or too low, the reason being that the network has not been exposed to this data during training.

However it is important to design the optimum architecture of the network regarding the number of hidden neurons. The accuracy of the network is less sensitive to the number of training samples. In most cases, the more patterns the better will be the accuracy. However a representative training data is essential for the network to approximate the function with sufficient accuracy. But the network is much more sensitive to the number of hidden neurons. With fewer neurons the network may not be capable of approximating the function fully while the larger number of them will make the network memorise the training data and therefore the network will generalise poorly. Selection of a proper training algorithm is also important in order to achieve the minimum error in reduced time. The Levenberg-Marquardt back-propagation algorithm has proven excellent performance in all applications in this study.

It is essential to preprocess the log data to form a good training dataset. Pre-processing may include depth shifting the cores to match log data and also depth shifting different log curves so that each curve reads the same formation interval. The core and the log data should represent similar reservoir conditions. Log data is always collected at in situ conditions while core data normally is

obtained at laboratory condition. Therefore, it is necessary to bring the core data also to be at in situ conditions by applying overburden corrections on the samples. It is important to use high resolution log data in order to match the resolution of core data otherwise the curves generated by neural network are much smoother than the core data. In lithofacies prediction also high-resolution log data is crucial in identifying all the thin facies, which can be seen on core data but hard to see on logs.

For the porosity neural network with three inputs, i.e. sonic, density, resistivity, the optimal number of hidden neurons has been determined to be in the range 6-10, with a sufficient number of training patterns of about 150. The network is sensitive to the fluid property. The unconstrained OLC approach of Hashem (1996), with zero intercept term based on least squares, is the most suitable ensemble approach for the porosity CM and the accuracy is mainly limited by the accuracy of the training patterns and the accuracy of the log data themselves.

Due to the dynamic range of permeability, varying from 0.1mDarcy to several Darcies within a reservoir section we find that splitting the whole range into a number of sub ranges increases the resolution and accuracy. Also due to the large dynamic range we normalised the logarithmic permeability values in order to bring the permeability values within the range of activation function. With the four inputs; i.e. sonic, density, gamma, neutron porosity, we find that optimal number of hidden units of the permeability neural network is confined to the range 8-12 where the variance and bias are at their minima. In general, the errors steadily decrease with the number of training facts. A practical lower limit has been set to 300, or twice the size of the training set required for the porosity network due to the increased complexity of the background relationships with the log readings. The networks should be trained by validation approach and the output should be combined by simple ensemble averaging instead of the OLC approach, which further enhances the noise in this case.

For predicting the fluids water, oil and gas, and their partial saturation the inputs are density, sonic, resistivity and neutron porosity logs. The output of three saturations i.e. oil, water and gas saturations from a single MLP (4-10-3) reveals the same accuracy as those of three individual MLPs with one output (4-4-1), the latter has the advantage of simplicity in terms of number of neurons, fewer training patterns and faster training. The optimal number of training patterns should be in excess of 100 to ensure negligible errors in case of data with moderate noise. A committee neural network for each fluid type is the preferred solution, with each network consisting of a number of individually trained MLPs connected in parallel and redundantly combined using OLC approach.

The problem of identification of lithofacies is solved by reducing the multiclass classification problem to a two-class classification problem using the

modular neural network system. The CM architecture is based on combining BPANN with a recurrent BPANN. Ensembles of neural networks are trained on disjoint sets of patterns using a soft overtraining approach for ensuring diversity and improving the generalisation ability of the stack.

It is important to check the sensitivity of the networks to different borehole fluids (water based or oil based mud) since the logging tools are sensitive to the mud properties. This could be one of the scopes for future studies. Also, comparing the net to gross results from porosity and/or permeability predictions, and comparing them with net to gross results from standard petrophysical interpretations may constitute a further enhancement of the present study.

While identifying lithofacies by neural network the facies thickness should be greater than a depth sample to be resolved and identified as a facies. In the present study we trained porosity, permeability and fluid saturation networks from log data using patterns from core data. In this approach we have taken the reservoir as a whole assuming that the influence of lithofacies architecture on reservoir properties and flow performance is embedded in log data. This approach is called non-genetic approach. In the genetic approach the reservoir is subdivided into flow units defined as a volume of rock subdivided according to geological and petrophysical properties that influence the flow of fluid through it and then each flow unit is characterised separately. Thus in this approach the genetic flow units are firstly identified and then the reservoir properties are predicted. Recent studies linking the genetic approach to reservoir engineering have shown that reservoir performance is more sensitive to lithofacies architecture than to details of petrophysical properties. In future work we may try the genetic approach, which emphasises the lithofacies prediction, or identification of flow units, rather than variation in petrophysical properties. This approach may help to characterise the inter-well region by mapping lithofacies from area to area. In future lithofacies identification can also be done in two steps. The first step can be identifying the clusters within the log data using an unsupervised network. In the second step a supervised network can be trained for labelling the clusters. The ambiguities due to overlapping of the facies can be resolved by employing recurrent back propagation networks. The ambiguities in the sequence of lithofacies can be avoided by providing a priori knowledge on the local stratigraphy to the gating network.

In future we plan to use seismic and log data for prediction of rock properties by using neural networks. Seismic inversion is an attempt to predict rock properties e.g. porosity, sand thickness, fluid content, hydrocarbon saturation etc. from seismic data. Under the most favourable conditions the seismic method is restricted to the estimation of only four rock physics parameters; i.e. P- and S-wave velocity, density and anisotropy. When inverting seismic data for rock properties such as e.g. porosity, we implicitly assume a relationship between the property and one or more of the fundamental seismic

parameters. The goal of attribute inversion is to visualise seismic patterns pertaining to a specific geologic interval. Inversion requires some form of constraint and need to be calibrated by tying the result to real or simulated well data. An unsupervised neural network can be used to perform this task by clustering seismic waveforms around a mapped horizon. Input to the neural network can be a set of seismic amplitudes. The number of clusters and the time gate relative to the mapped horizon can be defined by user. Each cluster is characterised by a waveform shaped class centre. The network firstly learns how to classify the seismic waveforms. This training is done on a representative selection of seismic waveforms. In unsupervised approach the aim is to find structure in the data themselves without imposing a priori conclusion. Unsupervised learning can be used for data clustering. Popular networks that use unsupervised learning are Kohonen feature maps (Kohonen, 1989). However, as we know the supervised approach requires a representative training data set. To validate the inversion method the network can be applied to the well data.

## References

- Abarbanel H. D. I., Brown R., Sidorowich J.J. and Tsimring L. S. 1993. The analysis of observed chaotic data in physical systems. *Reviews of Modern Physics* **65**, 1331-1392.
- Archie G.E. 1942. The electrical resistivity log as an aid in determining some reservoir characteristics. *Trans.Americ.Mineral.Met.* **146**, 54-62.
- Anand, R., Mehrotra, K., Mohan, C.K. & Ranka, S. 1995, Efficient classification for multiclass problem using modular neural networks. *IEEE Transactions on Neural Networks*, **6**, 117-124.
- Balan B., Mohaghegh S., Ameri S. 1995. State of the art in permeability determination from well log data part 1- a comparative study, model development. SPE 30978.
- Baum E., and Haussler D., 1989. What size net gives valid generalisation? In Touretzky, D., Ed., *Advances in neural information processing systems 1*: Morgan Kaufmann, 81-90.
- Bear J.,1988. *Dynamics of fluids in porous media*. Dover publications, inc. New York.
- Berryman J.G. 1980. Confirmation of Biot's theory. *Applied Physical Letters* **37**, 382-384.
- Bhatt A. and Helle H.B. 2001a. Committee neural networks for porosity and permeability prediction from well logs. *Geophysical Prospecting*, Nov 2002, 645-660. (Expanded abstract presented in EAGE/SEG Research workshop on reservoir rocks Pau, France, 2001)
- Bhatt A. and Helle H.B. 2001b. Determination of facies from well logs using modular neural networks. *Petroleum Geoscience*, **8** (3), 217-229.( Expanded abstract presented in EAGE 64<sup>th</sup> conference and technical exhibitions-Florence, Italy, 2002).
- Bhatt A. 1998. Porosity, permeability and TOC prediction from well logs using a neural network approach. M.Sc. thesis, Dep. Petroleum Technology and Applied Geophysics, NTNU, Trondheim.

- Brigaud F., Vasseur G. and Caillet G. 1992. Thermal state in the north Viking Graben(North Sea) determined from exploration well data. *Geophysics* **57**, 69-88.
- Breiman L. 1996. Bagging predictors. *Machine Learning* **26**, 123-140.
- Busch J.M., Fortney W.G. and Berry L.N. 1987. Determination of lithology from well logs by statistical analysis. *SPE Formation Evaluation* **2**, 412-418.
- Carcione J.M., Gurevich B. and Cavallini F. 2000. A generalized Biot-Gassmann model for the acoustic properties of shaley sandstones. *Geophysical Prospecting* **48**, 539-557.
- Chang H.C., Kopaska-Merkel D.C. Chen H.C. and Durrans S. R. 2000. Lithofacies identification using multiple adaptive resonance theory neural networks and group decision expert system. *Computers & Geoscience* **26**, 591-601.
- Cho S., Jang M. and Chang S. 1997. Virtual sample generation using a population of networks. *Neural Processing Letters* **5**, 83-89.
- Davey, N., Hunt, S.P. & Frank, R.J 1999, Time Series Prediction and Neural Networks. *EANN '99 Fifth International Conference on Engineering Applications of Neural Networks*, Warsaw Poland, 13-15 September, 1999.
- Delfiner P., Peyret O. and Serra O. 1987. Automatic determination of lithofacies from well logs. *SPE Formation Evaluation* **2**, 303-310.
- Dullien F.A.L. 1991. One and two phase flow in porous media and pore structure. In: *Physics of Granular Media*. (eds D.Bideau and J. Dodds), pp. 173-214. Science Publishers Inc.
- Fang J.H., Chen H.C., Shultz A.W. and Mahmoud W. 1992. Computer-aided well correlation. *AAPG Bull.* **76**, 307-317.
- Fischer W.D. 1958. On grouping for maximum homogeneity. *Journal of the American Statistical Association* **53**, 789-798.
- Freund Y. and Schapire S. 1996. Experiments with a new boosting algorithm. In: *Proceedings of the thirteenth international conference on machine learning*. 149-156.
- Geman S., Bienenstock E., Doursat R.1992. Neural networks and the bias/variance dilemma. *Neural Computation* **4**, 1-58.

- Gill D., Shamrony A. and Fligelman H. 1993. Numerical zonation of log suites and logfacies recognition by multivariate clustering. *AAPG Bulletin* **77**, 1781-1791.
- Hagan M.T., Demuth H.B. and Beale M. 1996. *Neural network design*. PWS Publishing Co.
- Hashem S. 1997. Optimal Linear Combination of Neural Networks. *Neural Networks* **10**, 599-614.
- Haykin S. 1999. *Neural networks: A comprehensive foundation*. Prentice Hall.
- Helle H.B., Bhatt A. and Ursin B. 2001. Porosity and permeability prediction from wireline logs using artificial neural networks: a North Sea case study. *Geophysical Prospecting* **49**, 431-444. (Expanded abstract presented in EAGE 62<sup>nd</sup> conference and technical exhibitions-Helsinki, Finland, 2001).
- Helle H.B. and Bhatt A. 2001. Fluid saturation from well logs using committee neural networks. *Petroleum Geoscience* **8**, May 2002. (Expanded abstract presented in EAGE 64<sup>th</sup> conference and technical exhibitions-Florence, Italy, 2002).
- Hornik K.M., Stinchcombe M. and White H. 1989. Multilayer feedforward networks are universal approximators. *Neural networks* **2(5)**, 359-366.
- Huang Z., Shimeld J., Williamson M. and Katsube J. 1996. Permeability prediction with artificial neural network modelling in the Venture gas field, offshore eastern Canada. *Geophysics* **61** 422-436.
- Huang Z. and Williamson M.A. 1997. Determination of porosity and permeability in reservoir intervals by artificial neural network modelling, offshore eastern Canada. *Petroleum Geoscience* **3**, 245-258.
- Jacobs R.A. 1995. Methods for combining experts probability assessments. *Neural computation* **7**, 867-888.
- Jian, F.X., Chork, C.Y., Taggart, I.J., McKay, D.M. & Bartlett, R.M. 1994, A genetic approach to the prediction of petrophysical properties. *Journal of Petroleum Geology*, **17**, 71-88.
- Jordan M.I. and Jacobs R.A. 1994. Hierarchical mixtures of experts and the em algorithm. *Neural Computation* **6(2)**, 181-214.
- Klimentos T. 1991. The effect of porosity-permeability-clay content on the velocity of Compressional waves. *Geophysics* **56**, 1930-1939.

- Kohonen T. 1989. *Self-organization and associate memory*, Springer-Verlag.
- Krooss B.M., Burkhardt M. and Schlömer S. 1998. Permeability and petrophysical properties of mudrocks from Haltenbanken area offshore Norway, Report 501398, Institut für Erdöl und Organische Geochemie, Jülich.
- Larionov W.W. 1969. *Borehole radiometry*. Nedra Moscow.
- Lawrence J. 1994. *Introduction to neural networks: Design, theory and applications*. California Scientific Software Press.
- Lucas A. 1998. An assessment of linear regression and neural network methods of porosity prediction from well logs. M.Sc. thesis, PRIS, University of Reading.
- Malki H.A., Baldwin J.L., Kwari M.A. 1996. Estimating permeability by the use of neural networks in thinly bedded shaly gas sands. *SPE Computer Applications* April, 58-62.
- Mavko G. and Nur A. 1997. Effect of a percolation threshold in the Kozeny-Carman relation. *Geophysics* **62**, 1480-1482.
- Masters T. 1993. *Practical neural network recipes in C++*. Academic Press.
- McCulloch W. and Pitts W. 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical biophysics* **5**, 115-133.
- Mohaghegh S. (ed) 2000. *Application of virtual intelligence to petroleum engineering*. *Computers & Geoscience* **26** (Special Issue).
- Naftaly U., Intrator N. and Horn D. 1997. Optimal ensemble averaging of neural networks. *Computation in Neural Systems* **8**, 283-296.
- Nelson P.H. 1994. Permeability-porosity relationships in sedimentary rocks. *The Log Analyst* **3**, 38-62.
- Nikravesh M., Amnizadeh F. and Zadeh L.A. (eds) 2001. *Soft computing and earth science*. *Journal of Petroleum Science & Engineering*, **29** (Special Issue)
- Parmanto B., Munro P.W. and Doyle H.R. 1996. Reducing variance of committee predictions with resampling techniques. *Connection Science* **8**, 405-425.
- Patterson D.W. 1996. *Artificial Neural Networks: theory and applications*. Prentice Hall, 477p.

- Poulton M.M. 2001. *Computational neural networks for geophysical data processing, Handbook of Geophysical Exploration*, Vol. 30, (K. Heilbig and S. Treitel eds.), Pergamon.
- Poupon A., Hoyle W.R. and Schmidt A.W. 1971. Log analysis in formations with complex lithologies. *Jour.Petroleum Tech.*, 995-1005
- Rogers S.J., Chen H.C., Kopaska-Merkel D.C., Fang J.H. 1995. Predicting permeability from porosity using artificial neural networks. *AAPG Bulletin* **79**, 12, 1786-1797.
- Rogers S.J., Fang J.H., Farr C.L. and Stanley D.A. 1992. Determination of lithology from well logs using a neural network. *AAPG Bulletin* **76**, 731-739.
- Rose W. and Bruce W.A. 1949. Evaluation of capillary character in petroleum reservoir rock. *Petroleum transactions AIME*, 127-142.
- Rosenblatt F. 1958. The perceptron: A probabilistic model for information storage and organisation in the brain. *Psychological review* **65**, 386-408.
- Ryseth A., Fjellbirkeland H., Osmundsen I.K., Skålnes Å. and Zachariassen E. 1998. High-resolution stratigraphy and seismic attribute mapping of a fluvial reservoir: Middle Jurassic Ness Formation, Oseberg Field. *AAPG Bulletin* **82**, 1627-1651.
- Ryseth A. 1989. Correlation of depositional patterns in the Ness formation Oseberg area. In: *Correlation in hydrocarbon exploration*. (eds. J.D. Collinson). 313-326.
- Saggaf M.M. and Nebrija Ed L. 2000. Estimation of lithologies and depositional facies from wire-line logs. *AAPG Bulletin* **84**, 1633-1646.
- Sharkey A. (ed) 1999. *Combining artificial neural nets – Ensemble and modular multi-net systems. Perspectives in Neural Computing*. Springer .
- Sharkey A., Sharkey N.E. and Chandroth G.O. 1996. Neural nets and diversity. *Neural Computing and Application* **4**, 218-227.
- Sharkey A., Sharkey N., Gerecke U. and Chandroth G.O. 2000. The “test and select” approach to ensemble combination. In: Kittler, J.& Roli, F. (eds) *Multiple Classifier Systems* pp. 30-44. *Lecture Notes in Computer Science*. Springer.
- Schapire R.E. 1990. The strength of weak learnability. *Machine Learning* **5**, 197-227.

- Schlumberger 1989. *Log interpretation. Principles and applications*. Schlumberger Educational Services Houston.
- Søndenå E., Bratteli F., Normann H.P. and Kolltveit K. The effect of reservoir conditions on saturation exponent and capillary pressure curve for water wet samples. In: *Advances in Core Evaluation* (eds P.F. Worthington) pp. 411-426, 1990 Gordon and Breach science publishers.
- Tiab D. and Donaldsson E.C. 1996. *Petrophysics. Theory and practice of measuring reservoir rock and fluid properties*. Gulf Publishing Co.
- Vernik L. 1997. Predicting porosity from acoustic velocities in siliciclastics: A new look, *Geophysics* **62**, 1, 118-128.
- Ward J.H.Jr. 1963. Hierarchical grouping to optimise an objective function: *Journal of the American Statistical Association* **58**, 236-244
- Wendt W.A., Sakurai S., Nelson P. H. 1986. Permeability prediction from well logs using multiple regression. In: *Reservoir Characterization* (eds L.W. Lake and H.B.Carroll Jr.), pp. 181-221. Academic Press.
- White H. 1992. *Artificial Neural Networks: Approximation and learning theory*. Oxford: Basil Blackwell.
- Widrow B., Hoff M.E. 1960. Adapting switching circuits. *IRE WESCON Convention record*, 96-104.
- Wishart D. 1969. An algorithm for hierarchical classifications. *Biometrics* **25**, 165-170.
- Wolpert D.H. 1992. Stacked generalisation. *Neural Networks* **5**, 241-259.
- Wolff M. and Pelissier-Combescure J. 1982. FACIOLOG: automatic electrofacies determination. *SPWLA Annual Logging Symposium*, Paper FF, 6-9.
- Wong P.M., Jang M., Cho S. and Gedeon T.D. 2000. Multiple permeability predictions using an observational learning algorithm. *Computers & Geoscience* **26**, 907-913.
- Wong P.M., Jian F.X. and Taggart I.J. 1995. A critical comparison of neural networks and discriminant analysis in lithofacies, porosity and permeability predictions. *Journal of Petroleum Geology* **18**, 191-206.

- Woodhouse R. 1998. Accurate reservoir water saturations from oil-mud cores: Questions and answers from Prudhoe bay and beyond. *The Log Analyst* May-June.
- Worthington P.F. 1991. Reservoir characterization at the mesoscopic scale,. In: *Reservoir Characterization II* (ed. L.W.Lake et al.) pp.123-165. Academic Press.
- Wyllie M.R.J. and Rose W.D. 1950. Some theoretical considerations related to the quantitative evaluation of the physical characteristics of reservoir rock from electrical log data. *Journal of Petroleum Technology*, 189.
- Wyllie M.R.J., Gregory A.R. and Gardner L.W. 1956. Elastic wave velocities in heterogeneous and porous media. *Geophysics* **21**, 41-70.
- Yang H., Chen H.C. and Fang J.H. 1996. Determination of carbonate lithofacies using hierarchical neural networks. *Intelligent Engineering Systems Through Artificial Neural Networks* **6**, 481-486.
- Zhang Y., Salisch H.A. and McPherson J.G. 1999. Application of neural networks to identify lithofacies from well logs. *Exploration Geophysics* **30**, 45-49.
- Zhang Y., Salisch H.A. and Arns C. 2000. Permeability evaluation in a glauconite-rich formation in the Carnarvon Basin Western Australia. *Geophysics* **65**, 46-53.