2000

Using knowledge-based neural networks in the prediction of permeability

William K. Zollinger
The University of Montana

USING KNOWLEDGE-BASED NEURAL NETWORKS IN THE PREDICTION OF PERMEABILITY

by

William K. Zollinger III

B. A. Oberlin College, 1991

presented in partial fulfillment of the requirements

for the degree of

Master of Science

The University of Montana

2000

Approved by:

Chairperson

Dean, Graduate School

5 - 11 - 2000

Date

Zollinger, William K., M.S., May 2000 Computer Science

Using Knowledge-Based Neural Networks in the Prediction of Permeability (38 pp.)

Director: David W. Opitz

Permeability prediction is one of the most important tasks in petroleum reservoir characterization; however, traditional laboratory determination of this property is time-consuming and expensive. Therefore, it is desirable to find a method of determining permeability which is less computationally expensive. Traditionally, empirically derived formulas or regression methods have been used; recently, machine learning techniques have been applied to this problem, most notably artificial neural networks (ANN's). We present research into this problem using another type of neural network, knowledge-based neural networks, or KNN's. KNN's combine the generalization power of traditional ANN's with some form of background knowledge, such as a mathematical formula (in this case, one for permeability). Using this method of permeability prediction, it is possible to obtain results which are superior to those of both the mathematical formulas alone and traditional ANN's. We present results for two types of data: numerical core data provided by Nance Petroleum, and computer images of core thin-sections. For the numerical core data, the KNN results were indeed superior to the formulas and to traditional ANN's; for the image data, the KNN results were generally inferior. However, most of the results point to the strong predictive power of this technique when applied to the problem of predicting permeability.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction and Motivation of the Problem

Much research has been done on the problem of predicting permeability—loosely defined as the ability of fluid to flow through the pore space—of oil-bearing rock from other factors. Permeability determination by physical means is a fairly complex and expensive process; therefore, any method which predicts permeability from other, more easily and cheaply determined quantities is worth investigating. Numerous empirical formulas have been discovered by scientists over the years; regression methods have also been used. In recent years, machine learning techniques have been brought to bear on this problem; the usual way is to train a neural network on a set of data, then apply this trained neural network to a test set. However, there is a fairly new subset of neural network research which focuses on giving the network some background information or knowledge first, then allowing the network to learn from there. Such networks are called "knowledge-based" neural networks, or KNN's. Given imperfect knowledge (such as a mathematical formula), a KNN should theoretically perform better than either the theory alone or a traditional neural network without background knowledge.

For this project, we have used a popular neural network software package (the Stuttgart Neural Network Simulator, or SNNS), and modified the networks produced so as to provide them with background knowledge. The first and greater part of this thesis details experiments performed with core data (principally permeability, porosity, and water saturation); the second part discusses work performed as a part of the Petrographic Image Analysis Project at the University of Montana, in which images of rock thin sections are analyzed to determine petrophysical properties. The results (particularly for the numerical core data) are encouraging: KNNs performed better in the majority of cases than traditional neural networks, and much better than the background knowledge alone. In the experiments for the PIA project, KNN's did not perform particularly well, but there are several possible ways to improve their predictive power in this area. Overall, however, knowledge-based neural networks represent a significant step in the improvement of permeability prediction.

Chapter 2: Background Information

This research is part of the petroleum reservoir characterization project established by the Montana Organization for Research in Energy (MORE) and the Montana University system in 1993 and funded by a grant from the U. S. Department of Energy. The project is intended to develop techniques to improve reservoir characterization through the analysis of seismic data, computer visualization of existing data, and new techniques for analyzing core data and samples. The original data was obtained from the Rabbit Hills oil field in Blaine County, north central Montana. Additional data was acquired from Nance Petroleum concerning the North Bainville oil field in Roosevelt County, northeastern Montana. More recently, the project has expanded to include the Petrographic Image Analysis (PIA) project at the University of Montana. This part of the project seeks to use new methods of image analysis to extract useful information from computer images of core thin sections.

The petrophysical property of greatest importance to this research is permeability, which is a measurement of the ease with which liquid is able to flow through porous rock. Permeability is of great interest to petroleum extraction, since a low-permeability field will be very difficult to pump the oil out of, regardless of how much oil is present in the reservoir. Ordinarily, the measurement of permeability involves expensive and time-consuming analysis of core samples in the laboratory. Therefore, it would be extremely desirable to have a method of permeability determinationwhich did not depend on this kind of analysis. Classically, empirical methods have been used to derive equations that give the relationships between permeability and more easily measured quantities, such as porosity (the percentage of empty space inside the rock) and water saturation (the percentage of fluid in the pore space of the rock that is water). The problem with static equations is that the exact empirical relationship varies considerably across geological formations (Balan, Mohaghegh, & Ameri, 1995b). An alternative technique is to derive adaptable formulas based on data analysis. Multiple regression methods have traditionally been used to derive permeability relationships (Wendt et al., 1985; Yao and Holditch, 1993); these methods perform well when given all data, but do not generalize well to unseen examples.

Therefore, machine learning techniques have been brought to bear on the problem, particularly the use of artificial neural networks. In rough terms, neural networks are composed of many interconnected simple units, each of which take one or more real-valued inputs (possibly the outputs of other units) and produce one real-valued output (possibly to be used as an input to another unit (Mitchell, 1996; Nelson and Illingworth, 1991)). The functions at each unit which compute the outputs are called *activation functions*. The two activation functions used in this study are the *identity* (more properly *identity plus bias*) function and the *sigmoid* function. The identity function produces an output which is equal to the sum of the products of each input unit and the weight of the edge between input and output unit: $o = b + \sum_j w_j i_j$, where o is the output value, i_j is the input value for the jth input unit, w_j is the weight of the edge between the jth input unit and the output unit, and b is the *bias* assigned to the output unit. The sigmoid function is as follows: $o = \sigma(\sum_j w_j i_j)$, where $\sigma(y) = \frac{1}{1+e^{-y}}$. This has the effect of "squashing" the output to values between 0 and 1 (Zell et al., 1995). Using these activation functions, one can model many different mathematical formulas with an artificial neural network.

Neural networks go through a process of *training*, in which data is fed into the network (thus such networks are called *feed-forward* networks), and the results are propagated backwards through the network (using an algorithm such as *backpropagation* (Rumelhart, Hinton, & Williams, 1986)), which then adjusts its weights and biases to try to improve the accuracy of its predictions. This process is repeated many times, until some stopping criterion is met (length of training or accuracy are most common). The trained neural network can then be tested on new data.

Artificial neural networks have been used to good effect in many petroleum reservoir characterization studies (Olson et al., 1997; Mohaghegh et al., 1994a; Mohaghegh and Ameri, 1995; Al-Kaabi and Lee, 1993; Juniardi and Ershaghi, 1993; Zhou et al., 1993; Mohaghegh et al., 1994b; Mohaghegh, 1995; Osborne, 1992; Wiener et al., 1995; Rogers et al., 1995), but they are sensitive to the quality and amount of the training data. Traditional artificial neural networks need a large amount of data to learn accurately, however, and data in the particular domain under study is expensive to obtain and rare. One way to address this problem is to incorporate background knowledge of the domain in the learning process. In this study, we use *knowledge-based neural networks*, or KNN's (Opitz & Shavlik, 1996). These networks make use of some sort of background knowledge, such as one of the empirically derived formulas mentioned above, as a starting point. Because the background knowledge supplied to the network is specific to the problem at hand, KNN's have an advantage over traditional neural networks, which are general tools. Since background knowledge makes up for lack of data, KNN's have a "head start" over traditional ANN's (which only analyze the given data) when used on the same data set. Given imperfect background knowledge and the generalization ability of the neural network, KNN's have been shown to perform better than either the background knowledge alone or the traditional neural network. In this study, we supply KNN's with empirical formulas for permeability derived from other data sets, and use these networks to predict permeability from new oil-field data.

Chapter 3: Knowledge-Based Neural Network Analysis of Core Data

3.1 Introduction

This part of the study tests the efficacy of knowledge-based neural networks (KNN's) in predicting permeability of oil-bearing rock from core data. The data set used was provided by Nance Petroleum in May of 1997. The rock is from the North Bainville field in Roosevelt County, Montana. The results given by the KNN's are then compared with results from traditional artificial neural networks (ANN's), as well as with the results from the formulas alone.

The data included, at each depth (in increments of one foot): permeability, porosity, oil saturation, water saturation, grain density, and lithological information on rock type, color, and structure (e.g., crystalline or sucrosic). Three classical empirical formulas were used to provide background knowledge to different neural networks: the Coates formula $K^{1/2} = 100\frac{\phi^2(1-S_{wirr})}{S_{wirr}}$ (Schlumberger Ltd., 1987), the Timur formula $K = 0.136\frac{\phi^{4.4}}{S_{wi}^2}$ (Timur, 1968), and the Tixier formula $K^{1/2} = 250\frac{\phi^3}{S_{wi}}$ (Tixier, 1949). All these formulas present permeability as a function of porosity and irreducible water saturation (Balan, Mohaghegh, & Ameri, 1995a). Our data did not include irreducible water saturation, only ordinary water saturation. However, since values for ordinary water saturation are more easily obtained than those for irreducible water saturation, this parameter was used and its utility tested in our networks.

Giving neural networks background information corresponding to a mathematical formula is mainly a question of determining the proper structure of the network, then assigning initial weight values to links and biases of nodes. In order to allow the networks to start from the formulas, then learn from there (and possibly improve the original knowledge), the networks are composed of two parts: the lower part has weights and biases corresponding to the formula used, while the upper part has weights and biases close to zero, so that its effect will be negligible at first, but could grow if the data required it (see Figure 1). In general, simple three-layer feedforward networks were used--one layer of inputs, one layer of hidden units, and one layer of outputs. The inputs to the network depended on the formula used; for the Coates formula, they were porosity, water saturation, and inverse water saturation ($1-S_w$), while for the other two formulas (and the traditional neural network used for comparison), they were only porosity and water saturation. In addition to these four networks, four other networks were used which had all given data (except depth) as inputs; for the KNNs, the weights and biases corresponding to quantities not present in the formulas were assigned numbers close to zero. The single output for all networks was permeability.

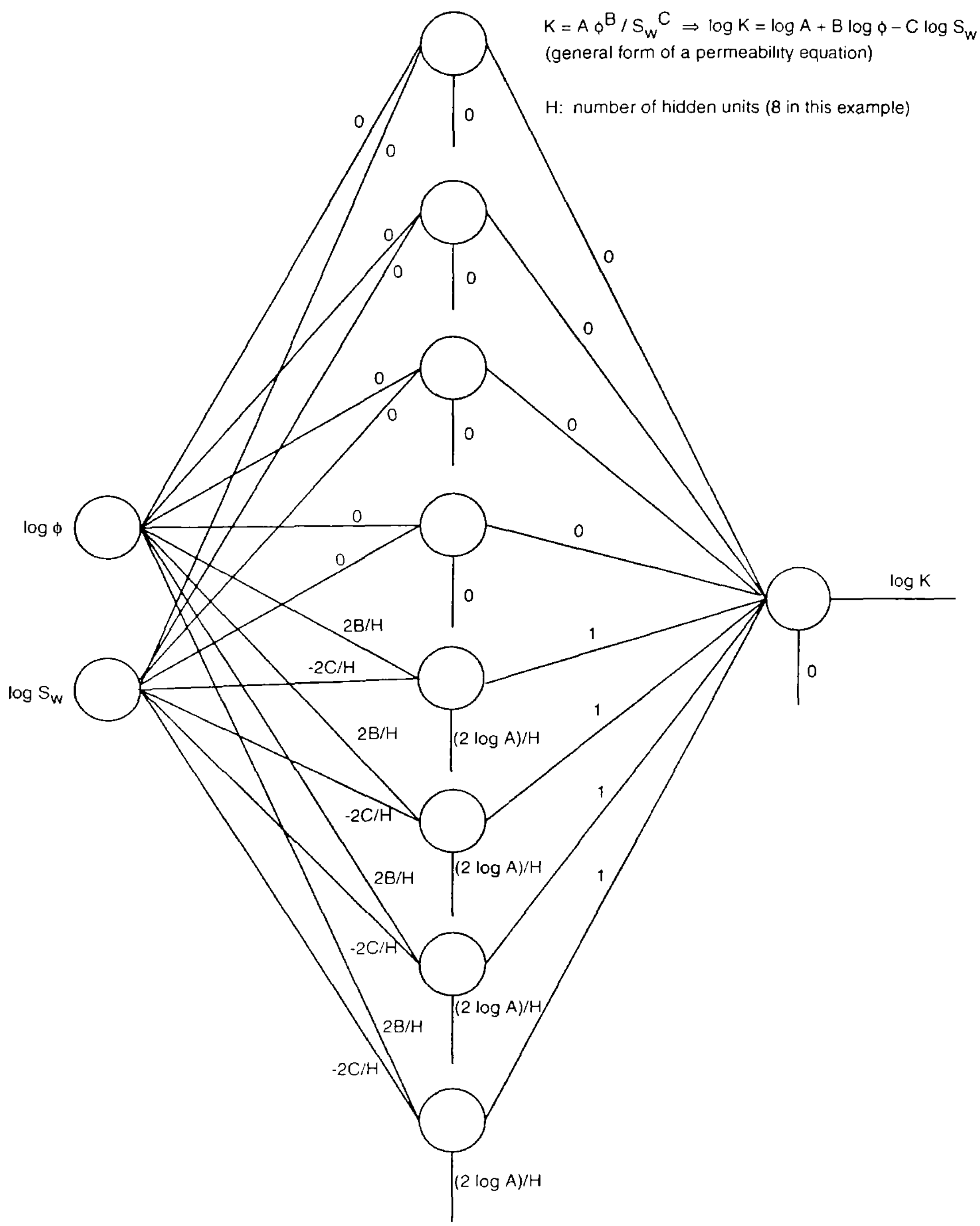

Figure 1. Structure of a typical KNN used in this study.

3.2 Methodology

It is desirable, when running neural network experiments, to have many trials with the same parameters and then average the results; to this end, scripts were written (in SNNS's scripting language, batchman) that would load in the proper networks and data sets, then run the experiments 100 times for each set of parameters. The learning rate chosen was 0.001, and the number of hidden units used for the original set of experiments was four and eight (for the set of experiments where all data was used as inputs, only eight hidden units were used). Each network was trained both for 200 epochs and for 1000 epochs. The Nance data is on three wells (Crusch 2-10, Granley 14-10, and Granley State 4-15X); each was divided into two sections based on the rock layer they occupied (Red River, Winnipegosis, and Nisku formations). This made for six individual data sets; in the experiments, each network was trained on all but one of the sets and tested on the remaining set (all six sets were used once and only once as test sets during each experiment). The final results present the sum-squared error obtained by each network during training; these results were obtained from the SNNS result files by the use of two programs, written in C, to collect results, calculate the error, then compute averages and build the final results tables.

3.3 Results

In the following tables, "all inputs" denotes the inclusion of the following quantities as inputs to the neural networks: porosity, water saturation, oil saturation, grain density, rock type, rock color, and rock structure (crystalline or sucrosic). Experiments were run with all inputs to determine if other properties of the rock to which we had access (but which were not present in the formulas) would have any effect on the permeability. The phrase "original inputs" refers to the inclusion of only porosity and water saturation, as required by all three formulas. The phrase "empirical formulas alone" denotes a learning rate of 0.0, meaning that the data was tested against the indicated formula without learning; we wanted to test the predictive power of the KNN's against that of the original empirical formulas.

Table 1. Test error for KNN's based on the Coates formula. Original, unscaled inputs were used.

		Epochs	
Well	**Hiddens**	**200**	**1000**
cr210nis	4	0.517599	0.510523
	8	0.515135	0.514341
cr210red	4	0.374281	0.351099
	8	0.380321	0.350542
g1410red	4	0.569293	0.555267
	8	0.569733	0.554259
g1410win	4	1.317951	1.428904
	8	1.244084	1.410777
gs415nis	4	0.875389	0.942927
	8	0.873537	0.904255
gs415win	4	0.560646	0.610062
	8	0.558409	0.583339
Total	4	**0.697487**	**0.720940**
	8	**0.683316**	**0.710852**

Table 2. Test error for KNN's based on the Timur formula. Original, unscaled inputs were used.

		Epochs	
Well	**Hiddens**	**200**	**1000**
cr210nis	4	0.525727	0.528437
	8	0.583140	0.534875
cr210red	4	0.325893	0.317928
	8	0.334190	0.319173
g1410red	4	0.545705	0.533969
	8	0.556118	0.534830
g1410win	4	1.391640	1.452781
	8	1.352305	1.444617
gs415nis	4	0.896031	0.932046
	8	0.878504	0.925016
gs415win	4	0.563888	0.592025
	8	0.559334	0.595899
Total	4	**0.695847**	**0.708720**
	8	**0.693358**	**0.707855**

Table 3. Test error for KNN's based on the Tixier formula. Original, unscaled inputs were used.

		Epochs	
Well	**Hiddens**	**200**	**1000**
cr210nis	4	0.517065	0.512972
	8	0.508014	0.536497
cr210red	4	0.346406	0.326468
	8	0.357481	0.324668
g1410red	4	0.562191	0.539605
	8	0.561982	0.540962
g1410win	4	1.342839	1.437655
	8	1.242489	1.414065
gs415nis	4	0.882430	0.980496
	8	0.891262	0.915787
gs415win	4	0.560835	0.626678
	8	0.549397	0.610641
Total	4	**0.694446**	**0.716623**
	8	**0.675051**	**0.706226**

Table 4. Test error for traditional ANN's. Original, unscaled inputs were used.

		Epochs	
Well	**Hiddens**	**200**	**1000**
cr210nis	4	1.162254	1.158321
	8	1.187146	1.168704
cr210red	4	0.654156	0.642242
	8	0.654898	0.644038
g1410red	4	0.936313	0.919763
	8	0.934333	0.920972
g1410win	4	1.558948	1.551680
	8	1.562663	1.552976
gs415nis	4	1.383610	1.376585
	8	1.378594	1.375139
gs415win	4	0.803614	0.791259
	8	0.808847	0.794206
Total	4	**1.022243**	**1.010232**
	8	**1.023661**	**1.011942**

Table 5. Test error for KNN's based on the Coates formula. All inputs were used; they were not scaled.

	Epochs	
Well	**200**	**1000**
cr210nis	1.120057	3.321761
cr210red	0.437006	1.376393
g1410red	0.566376	1.315502
g1410win	1.293642	1.725247
gs415nis	0.992422	1.186664
gs415win	0.757795	1.772915
Total	**0.758822**	**1.530589**

Table 6. Test error for KNN's based on the Timur formula. All inputs were used; they were not scaled.

	Epochs	
Well	**200**	**1000**
cr210nis	1.457017	5.881400
cr210red	0.406961	0.853509
g1410red	0.610849	0.830128
g1410win	1.410573	1.508362
gs415nis	0.959181	0.997905
gs415win	0.906152	2.705476
Total	**0.819524**	**1.395942**

Table 7. Test error for KNN's based on the Tixier formula. All inputs were used; they were not scaled.

	Epochs	
Well	**200**	**1000**
cr210nis	1.057386	1.589254
cr210red	0.442296	1.256468
g1410red	0.587100	1.349423
g1410win	1.214117	1.426103
gs415nis	1.022631	1.586967
gs415win	0.759051	4.205468
Total	**0.750866**	**1.734002**

Table 8. Test error for traditional ANN's. All inputs were used; they were not scaled.

	Epochs	
Well	**200**	**1000**
cr210nis	1.188699	1.140115
cr210red	0.651981	0.632374
g1410red	0.933165	0.917783
g1410win	1.562276	1.545858
gs415nis	1.377113	1.377811
gs415win	0.805820	0.772495
Total	**1.022081**	**1.003233**

Table 9. Test error for empirical formulas alone, with no learning.

Well	Coates formula	Timur formula	Tixier formula
cr210nis	16.2134	66.6478	44.3128
cr210red	2.7299	37.1461	9.4474
g1410red	9.4049	52.6482	23.7075
g1410win	11.4027	65.1455	34.6634
gs415nis	6.3327	52.1229	26.5631
gs415win	3.6693	30.5713	9.1957
Total	**7.6204**	**49.3814**	**21.8181**

Table 10. Test error for KNN's vs. ANN's and mathematical formulas (summary table).

Experiment	Test error
KNN (Coates formula)	0.683316
KNN (Timur formula)	0.693358
KNN (Tixier formula)	0.675051
Traditional ANN	1.023661
Coates formula alone	7.6204
Timur formula alone	49.3814
Tixier formula alone	21.8181

As these results show, the KNN's outperform both the traditional ANN's and the formulas alone, with the networks based on the Tixier formula generally more accurate; the KNN's with original inputs perform generally better than those with full inputs, suggesting that the major contributors to the rock's permeability are in fact porosity and water saturation, although other factors may contribute to a lesser degree. Particularly in the case of the networks with full inputs, 200 was a considerably better number of training epochs than 1000; this probably indicates overtraining (learning the training data so well that the power to generalize to unseen examples is worsened). The sum-squared error is still fairly high in an absolute sense; this may be because of the use of the normal water saturation in place of the irreducible water saturation in the formulas the KNN's were based on. However, the KNN's still represent a significant gain over traditional methods of permeability determination for this data set.

3.4 Effect of Scaling Inputs

Next, experiments were performed to test the effect of scaling the input values to numbers between 0 and 1. There is evidence that scaling the inputs increases the predictive power of a neural network, so we used the same KNN's as in the first round of experiments, but the inputs were scaled. In all following tables, "normal" refers to original inputs, and "full" refers to full inputs.

Table 11. Test error for KNN's based on the Coates formula; scaled inputs were used.

Well	Normal	Full
cr210nis	0.487937	0.969751
cr210red	0.385140	0.449516
g1410red	0.569889	0.651619
g1410win	1.233399	1.070765
gs415nis	0.883921	1.091021
gs415win	0.556399	0.823247
Total	**0.681772**	**0.755054**

Table 12. Test error for KNN's based on the Timur formula; scaled inputs were used.

Well	Normal	Full
cr210nis	0.502030	1.015916
cr210red	0.358630	0.449823
g1410red	0.561904	0.683893
g1410win	1.243603	1.092363
gs415nis	0.889512	1.071892
gs415win	0.547504	0.813436
Total	**0.674947**	**0.769654**

Table 13. Test error for KNN's based on the Tixier formula; scaled inputs were used.

Well	Normal	Full
cr210nis	0.502097	1.005222
cr210red	0.359447	0.439400
g1410red	0.563863	0.651678
g1410win	1.243662	1.090952
gs415nis	0.891225	1.070754
gs415win	0.550348	0.803199
Total	**0.676295**	**0.754237**

Table 14. Test error for traditional ANN's; scaled inputs were used.

Well	Result
cr210nis	1.207626
cr210red	0.657510
g1410red	0.940574
g1410win	1.568605
gs415nis	1.389131
gs415win	0.811549
Total	**1.029350**

In the experiments with full inputs, the scaled-input trials produced better results, particularly in the case of the Tixier formula. However, since the results of the full-input experiments were worse than those of the original-input experiments (where scaling did not make much of a difference), the obvious conclusion is that scaling the inputs does not generally improve predictive power for this data set.

3.5 Testing Structure

Next, experiments were run to test the effects of different types of hidden units and output units, as well as the effects of different network structures. The first set of experiments in this round were run with networks structured like the original KNN's (half the hidden units sigmoids, half of them identity units), but with all weights and biases near 0. There were two networks used with different input unit structures: 3 input units (to parallel the structure of a network based on the Coates formula) and 2 input units (to parallel the structure of a network based on the Timur or Tixier formulas).

Table 15. Test error for zero-weight and -bias KNN's based on the Coates formula; unscaled inputs were used.

Well	Normal	Full
cr210nis	0.496170	0.852087
cr210red	0.358878	0.446844
g1410red	0.563371	0.558470
g1410win	1.242030	1.097265
gs415nis	0.900729	1.021621
gs415win	0.546894	0.688385
Total	**0.675663**	**0.702023**

Table 16. Test error for zero-weight and -bias KNN's based on the Timur or Tixier formula; unscaled inputs were used.

Well	Normal	Full
cr210nis	0.500230	0.861558
cr210red	0.357924	0.447078
g1410red	0.563297	0.563387
g1410win	1.241422	1.099358
gs415nis	0.899430	1.016314
gs415win	0.547607	0.685716
Total	**0.675441**	**0.703813**

These results are very close to the results involving the original KNN's with non-zero weights and biases (although the discrepancy is more pronounced in the case of full inputs); this suggests that the structure of the network plays a strong role in its predictive power.

3.6 Combining Structure and Scaling

Next, the effects of both scaling inputs and network structure were examined simultaneously. The type of network used in this set of experiments was a KNN with zero weights and biases as in the last set of experiments, but with scaled inputs.

Table 17. Test error for zero-weight and -bias KNN's; inputs were scaled between 0 and 1.

Well	Result
cr210nis	1.046467
cr210red	0.490098
g1410red	0.656036
g1410win	1.098159
gs415nis	1.133576
gs415win	0.675194
Total	**0.757722**

These results are very close to the results from the same experiments run without scaled inputs (although the discrepancy is more pronounced with the KNN-structured networks). This suggests, once again, that scaling the inputs is not a particularly useful technique with this type of data.

3.7 Testing Substructure

The next experiments in this round were run with two types of networks, each of which had all weights and biases set to near 0: 1) a network with 8 hidden units with identity activation functions and an output unit with the identity activation function (referred to in the tables as "id"); and 2) a perceptron (no hidden units) with the identity output function (referred to in the tables as "perc"). These experiments were run for the same two types of input unit structures as in the last round of experiments.

Table 18. Test error for zero-weight and -bias KNN's based on the Coates formula. The output unit was an identity unit, and the inputs were unscaled.

Well	Structure	Normal	Full
cr210nis	id	0.496353	0.858109
	perc	0.497586	0.856579
cr210red	id	0.357888	0.449024
	perc	0.356223	0.458319
g1410red	id	0.564509	0.556696
	perc	0.568089	0.552081
g1410win	id	1.244318	1.096953
	perc	1.249490	1.096186
gs415nis	id	0.904771	1.024515
	perc	0.920684	1.029684
gs415win	id	0.546548	0.693415
	perc	0.546426	0.695798
Total	id	**0.676538**	**0.702875**
	perc	**0.679557**	**0.703800**

Table 19. Test error for zero-weight and -bias KNN's based on the Timur or Tixier formula. The output unit was an identity unit, and the inputs were unscaled.

Well	Structure	Normal	Full
cr210nis	id	0.499252	0.866597
	idsig	0.852855	0.691827
	perc	0.495625	0.861211
cr210red	id	0.357358	0.449082
	idsig	0.581598	0.482367
	perc	0.356923	0.455986
g1410red	id	0.564268	0.560645
	idsig	0.903831	0.818219
	perc	0.568593	0.554368
g1410win	id	1.239091	1.097574
	idsig	1.517539	1.434998
	perc	1.230604	1.092373
gs415nis	id	0.903567	1.019275
	idsig	1.158628	1.055668
	perc	0.920851	1.025725
gs415win	id	0.547187	0.690729
	idsig	0.712559	0.606021
	perc	0.546042	0.695019
Total	id	**0.675362**	**0.703961**
	idsig	**0.949241**	**0.854848**
	perc	**0.675903**	**0.703073**

The results for identity hidden units and for the perceptron were comparable to the results for the half identity, half sigmoid hidden unit experiments; there is no significant difference between them. The results for the sigmoid hidden units were generally worse. This suggests that at least some of the hidden units should be identity units.

The next experiments were run to test the effect of sigmoid output functions. There were two types of networks in this round: 1) networks structured like the original KNN's (half the hidden units sigmoids, half of them identity units), but with zero weights and biases and a sigmoid output unit (referred to as "strucsig"); and 2) a perceptron with zero weights and biases and a sigmoid output unit (referred to as "sigperc").

Table 20. Test error for zero-weight and -bias KNN and perceptron with sigmoid output unit; unscaled inputs were used.

Well	Structure	Result
cr210nis	strucsig	1.181631
	sigperc	1.058964
cr210red	strucsig	0.649768
	sigperc	0.589540
g1410red	strucsig	0.931467
	sigperc	0.860880
g1410win	strucsig	1.555992
	sigperc	1.478983
gs415nis	strucsig	1.372084
	sigperc	1.322151
gs415win	strucsig	0.800440
	sigperc	0.704378
Total	strucsig	**0.986996**
	sigperc	**0.945240**

These results were significantly worse than those for the zero-weight and -bias KNN's and the perceptron with identity output. This confirms the idea that the ideal output unit is an identity unit; sigmoid output units (and exclusively sigmoid hidden units) make the networks perform worse overall.

The final round of experiments was performed, once again, with networks that have identity outputs: one network was a perceptron (referred to as "idnormperc"), and the other was a network with sigmoid hidden units and identity output (referred to as "idnormsig"). In this set of experiments, the input data is scaled between 0 and 1.

Table 21. Test error for ANN with sigmoid hidden units and perceptron; both type of network had an identity output unit. Inputs were scaled.

Well	Structure	Result
cr210nis	idnormperc	0.944111
	idnormsig	1.181835
cr210red	idnormperc	0.516337
	idnormsig	0.656294
gl410red	idnormperc	0.664077
	idnormsig	0.934118
gl410win	idnormperc	1.134884
	idnormsig	1.575938
gs415nis	idnormperc	1.172060
	idnormsig	1.366635
gs415win	idnormperc	0.600592
	idnormsig	0.823740
Total	idnormperc	**0.763457**
	idnormsig	**1.027545**

Once again, these results are worse than the identical experiments for unscaled inputs, thus confirming that scaling the inputs between 0 and 1 is not necessary for this data.

Putting the results of all these experiments together, we arrive at the ideal structure for a neural network to be used in analysis of this core data. The network should be a KNN (or at least have the KNN structure as defined above) with at least some of the hidden units identity units; the output unit should also be an identity unit. The networks based on the Tixier formula seem to do slightly better generally. Furthermore, the ideal input format is also obtained from these results; the inputs should be porosity and water saturation (and, in the case of the Coates formula, inverse water saturation), and should be left at their normal values, not scaled between 0 and 1. When this prescription is followed, one obtains the most accurate prediction of permeability among all methods, and improvement over formula use alone is especially dramatic.

Chapter 4: Knowledge-Based Neural Network Analysis of Data Acquired from Thin-Section Images

This chapter details research performed as part of the Petrographic Image Analysis project at the University of Montana. One of the goals of this project is to be able to calculate physical properties of oil-bearing rock from images made of thin sections taken from that rock. Various image transformations are performed on the original thin-section images in an effort to extract meaningful petrophysical data.

For these experiments, we were given 28 color slides made from thin sections taken at three different depth levels from oil-bearing rock originating in the Rabbit Hills oil field in north-central Montana. The core samples were prepared by first removing all residual fluid from the pore spaces, then injecting them with a blue epoxy to make it easier to identify pore space in the slides. Since none of the natural colors in the rock were any shade of blue, the distinction was fairly easy to make. Thin sections were then cut from the rock, and slides taken of them using a Carl Zeiss polarizing microscope with Zeiss objectives at 1.25x and a 35mm camera with Kodak Ektachrome Tungsten film at ASA 160. Each thin section's image area was approximately 8.25 mm across and 5.5 mm high; when made into slides, the image area was approximately 3.5 cm by 2.3 cm.

These slides were scanned into Adobe Photoshop using a Microtek ScanMaker 35T at an optical resolution of 1600 dpi (interpolated to 3000 dpi).

In order to utilize a knowledge-based neural network for these experiments, an equation for permeability was needed that would be useful for data gleaned from these thin-section images. Following Mowers and Budd (Mowers & Budd, 1996), we used a two-dimensional form of the classical Kozeny-Carman equation $k = \frac{\theta^3}{c(1-\theta)^2(4PP/\pi PA)^2}$, where k represents permeability, θ represents porosity, PP represents total pore perimeter, PA represents total pore area, and c is a proportionality constant generally assumed to be 5. In order to use this equation, the porosity, pore perimeter, and pore area had to be determined. In order to do this, the original images had to be cleaned up so that the pore and grain space could be more accurately measured.

After scanning the slides, the color images were in JPEG format. The ultimate goal for these images was to reduce them to black and white images, where the pore spaces would be white and the grain spaces would be black. This was done using Adobe Photoshop, the industry standard image processing software, on a Power Macintosh G3. The first step in this process was to create grayscale images of the slides; this was done with the Photoshop function "Select Color Range". A color in the image was selected (blue, to correspond with pore space), and the tolerance was adjusted manually until the resulting selection was felt to be as close as possible to what appeared to be the true pore space. This selection was saved as a separate channel, and then the original channels were removed, leaving a grayscale image (see Figure 2).

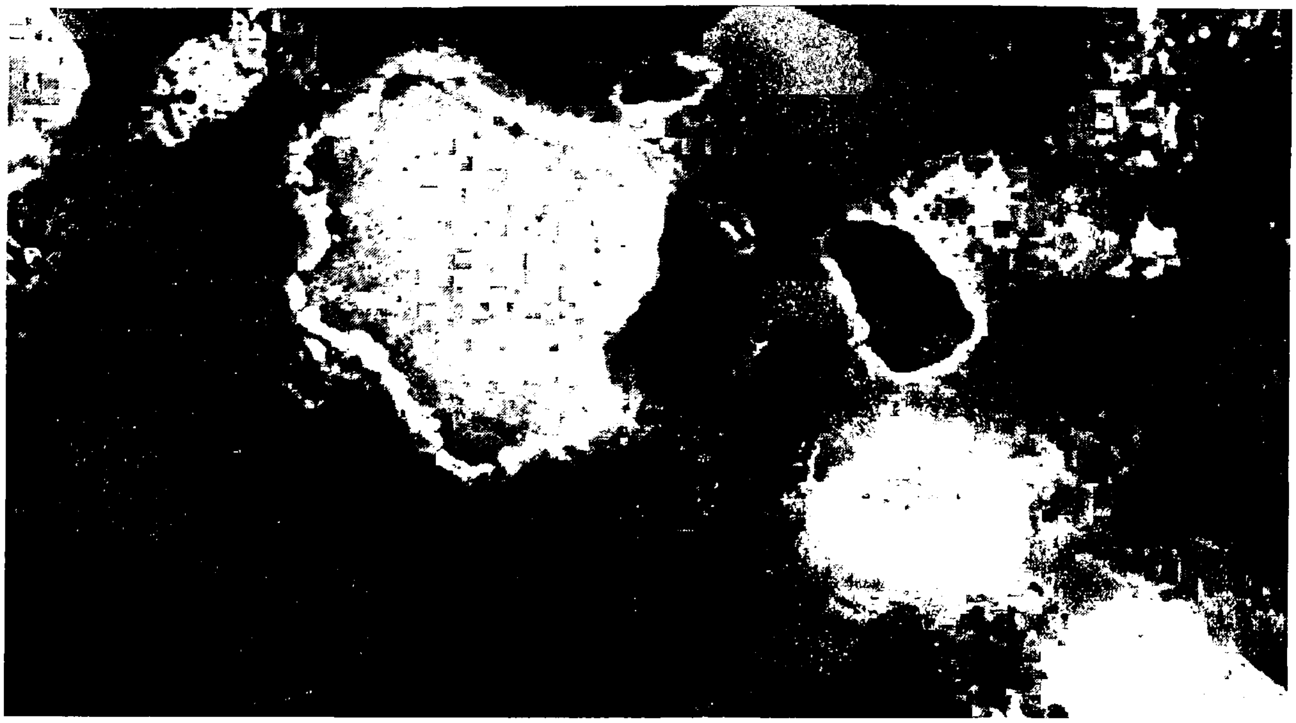

Figure 2. Grayscale representation of example pore space.

Finally, the grayscale image was converted to a black and white image using Photoshop's "Adjust Threshold" function, in which pixels above a certain value were colored white, while pixels below that value were colored black. Again, this threshold value was adjusted manually until the most acceptable depiction of pore space was achieved (see Figure 3).

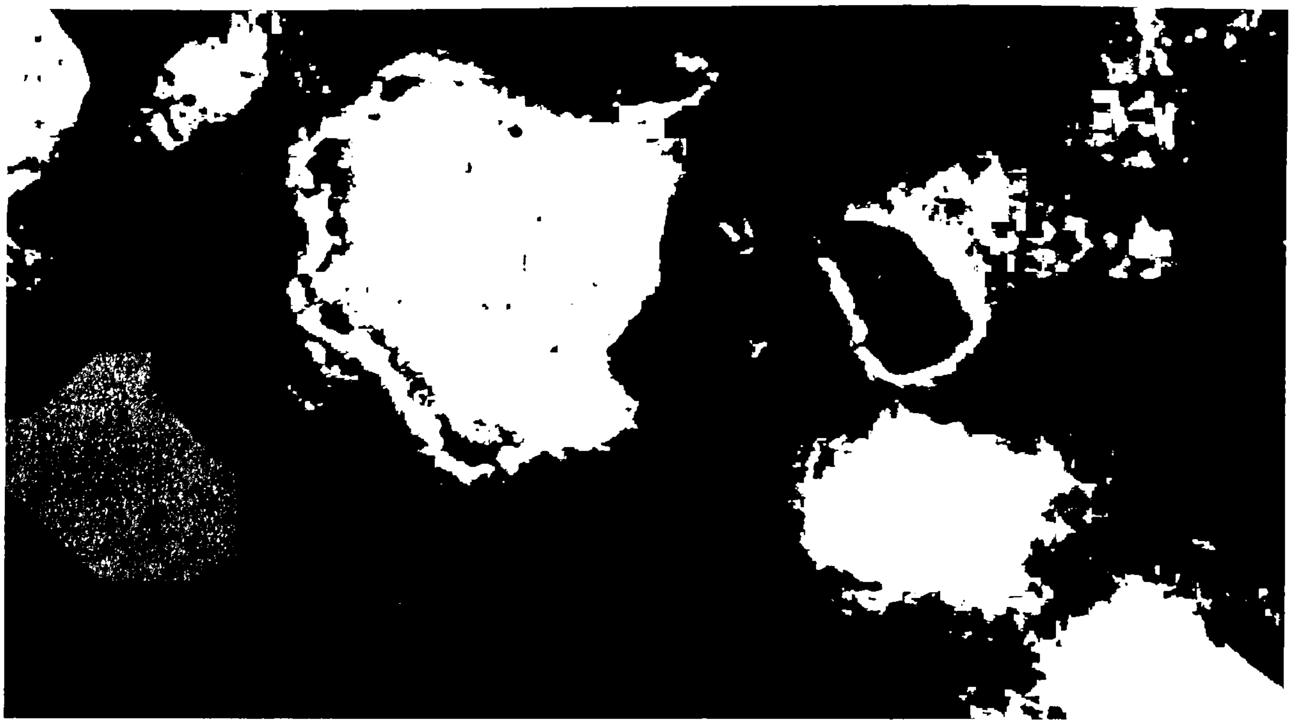

Figure 3. Black and white representation of pore space.

In order to use the information in these images in the formula the KNN's would be based on, the relevant quantities had to be extracted from them: total pore perimeter, total pore area, and porosity. For the first two, we used a Java class written by Dr. Ray Ford at the University of Montana for the PIA project, called AREAOPNS. This class has the ability (along with many others), to examine an image and find all areas of equal color value, then calculate the areas and perimeters of each, as well as giving totals for both. However, this class only has the ability to work with TIFF images and ERDAS images; since TIFFs take up so much space, we elected to convert the images to ERDAS format. Since the images were by this time only black (for grain space) and white (for pore space), it was a relatively simple matter, once we had them in an acceptable format, to use AREAOPNS to calculate the total pore perimeter and area. Once these quantities were acquired, porosity was simple to calculate, being merely the percentage of pore space in the image. Since we knew the total area of the image, and we now had the total pore area, we could easily calculate the porosity of each image. These three quantities were used as inputs to the neural network.

Experiments were run on each image; the results were then averaged over each depth value. In addition to the KNN experiments, experiments were performed using ANN's and experiments using the original formula (which was accomplished by using a KNN with the learning rate set to zero). For the KNN and ANN experiments, the inputs were then scaled to values between zero and one, to determine if this would have any effect on the accuracy of the networks' permeability predictions.

Table 22. Results, KNN's with original inputs

	Epochs	
Depth (meters)	**200**	**1000**
4134	test error = 5.177378	test error = 6.880134
4135	test error = 6.200812	test error = 6.356312
4137	test error = 0.872343	test error = 0.616543

Table 23. Results, KNN's with original, scaled inputs

	Epochs	
Depth (meters)	**200**	**1000**
4134	test error = 5.423697	test error = 6.915390
4135	test error = 5.896326	test error = 6.203959
4137	test error = 0.252743	test error = 0.179526

Table 24. Results, ANN's with original inputs

	Epochs	
Depth (meters)	**200**	**1000**
4134	test error = 3.878230	test error = 3.904831
4135	test error = 2.544261	test error = 2.471070
4137	test error = 0.855400	test error = 0.783412

Table 25. Results, ANN's with original, scaled inputs

	Epochs	
Depth (meters)	**200**	**1000**
4134	test error = 3.846381	test error = 3.897002
4135	test error = 2.355366	test error = 2.568803
4137	test error = 0.819637	test error = 0.738818

Table 26. Results, formula

Depth (meters)	Result
4134	test error = 4.779297
4135	test error = 0.716201
4137	test error = 0.282900

As these results show, the KNN's performed significantly worse in this case than the ANN's, and worse even than the original formula. There are several possible reasons for this. First, as Mowers and Budd warn, the value of 5 used for the proportionality constant may not in fact be accurate in all cases. "[T]here is no a priori reason why one value for this empirical constant would be applicable to all types of reservoir rocks." (Mowers & Budd, 1996, p. 313) The value of this constant depends on two other factors: the Kozeny constant, which has been shown to vary for different pore shapes (Carman, 1956), and the tortuosity, a geometric quantity which certainly should vary for each pore system. However, since tortuosity is extremely difficult to determine, a value of 5 was chosen for these experiments, and this almost certainly had an impact on the predictive power of the KNN's.

Second, due to the nature of the transformations the images went through between thin-section creation and the neural network experiments, some information was certainly lost. The process of obtaining the black and white final images for input to AREAOPNS was a somewhat subjective one, as the tolerances and sample pore colors were adjusted manually. Since there were so many steps between the original thin sections of rock and the black and white images, it would be unreasonable to expect that we were working with a perfect representation of the pore space.

Finally, although the form of the Kozeny-Carman equation used was specifically tailored for use with two-dimensional images, it is not straightforward to take techniques designed for a three-dimensional model and apply them to a two-dimensional one. The pore-system model is inherently three dimensional; porosity is really a percentage of pore volume to total volume, and permeability measures the ability of a fluid to flow in a three-dimensional manner through the rock. Steroscopic techniques are brought to bear in order to justify applying this model to a two-dimensional projection, but it is an imperfect approximation.

All this aside, it should be noted that the traditional neural networks performed significantly better than the original formula. This suggests that machine learning techniques do have advantages over traditional methods, even with this data set and these approximate methods. If better values for the proportionality constant were determined for this data set, and better approximations of the pore space were obtained, it may be that KNN's would outperform other methods.

Chapter 5: Conclusion

As we have seen, knowledge-based neural networks can be used as an effective method of predicting permeability in oil-bearing rock. When given an adequate data set (in terms of size and accuracy of representation of the overall field), KNN's outperform not only the original empirically-derived formulas used as background knowledge to the networks, but also more traditional artificial neural networks whose weights and biases have not been calculated to represent some mathematical formula. Using KNN's to analyze data gleaned from thin-section images is still in its infancy, however, and their predictive power in this area has not been fully developed.

There are several areas of potential further study. First, in order to use KNN's in petrographic image analysis, a better method is obviously needed to extract the relevant petrophysical properties from the images. The method used in the research described above was quite subjective; manual adjustments were necessary at most stages of the process. If a way could be found to automatically, reliably, and accurately acquire the needed data from the images, this would undoubtedly increase the ability of the KNN's to predict permeability: after all, any method is dependent on the quality of the data used.

Secondly, the actual permeability equation used to inform the KNN's should undergo more scrutiny. As mentioned in Chapter 4, the proportionality constant probably should not have a value of 5; in fact, it should probably be dependent on the particular field or formation under study. Mowers and Budd used a simple iterative process to ultimately arrive at a proper value of the constant for their particular study; something similar may be indicated here. With a more accurate value of this constant as part of the equation, the accuracy of the KNN's prediction would likely increase.

Lastly, there is the problem of essentially using a two-dimensional projection of a three-dimensional model. Since porosity, permeability, and similar quantities are properties of three-dimensional rock, using a two-dimensional image (and an equation derived for the two-dimensional case) is probably an oversimplification. The overall project was created to take advantage of, among other things, 3-D seismic data; if this data could be used, and if an image analysis procedure to deal with three-dimensional images were devised, this would probably provide a more accurate picture of the structure of the actual rock. The data extracted from this process would allow the KNN's to more accurately predict permeability as well as other petrophysical properties.

All this said, however, knowledge-based neural networks have proven to be an effective way to analyze petrographic data; when used on numerical core data, the results are extremely gratifying. As more study is undertaken in this area, there is great potential for them to be a useful tool to the petroleum industry, as well as to other areas of study. Given the proper data and background knowledge, their applicability is potentially very wide-ranging.

References

1. Al-Kaabi, A., Lee, W. J.: "Using Artificial Neural Nets to Identify the Well Test Interpretation Model", SPE Formation Evaluation, September 1993 (p. 233-240).

2. Balan, B., Mohaghegh, S., Ameri, S.: "State-Of-The-Art in Permeability Determination From Well Log Data: Part 1—A Comparative Study, Model Development", SPE 30978, SPE Eastern Regional Conference & Exhibition, Morgantown, West Virginia, September 1995a.

3. Juniardi, I. J., Ershaghi, I.: "Complexities of Using Neural Networks In Well Test Analysis of Faulted Reservoir", SPE 26106, Proceedings, SPE Western Regional Meeting, Anchorage, Alaska, March 1993.

4. Mitchell, T. M.: *Machine Learning*, McGraw Hill, September 1996.

5. Mohaghegh, S.: "Neural Network: What Can It Do for Petroleum Engineers", Journal of Petroleum Technology, January 1995 (p. 42).

6. Mohaghegh, S., Arefi, R., Ameri, S., Hefner, M. H.: "A Methodological Approach for Reservoir Heterogeneity Characterization Using Artificial Neural Networks", SPE 28394, Proceedings, SPE Annual Technical Conference and Exhibition, New Orleans, Louisiana, 1994a.

7. Mohaghegh, S., Arefi, R., Ameri, S., Rose, D.: "Design and Development of an Artificial Neural Network for Estimation of Formation Permeability", SPE 28237, Proceedings, SPE Computer Conference, Dallas, Texas, 1994b.

8. Mohaghegh, S., Ameri, S.: "Artificial Neural Network As A Valuable Tool For Petroleum Engineers", SPE 29220, unsolicited paper, Society of Petroleum Engineers, Inc., Richardson, Texas, 1995.

9. Mohaghegh, S., Balan, B., Ameri, S.: "State-Of-The-Art in Permeability Determination From Well Log Data: Part 2—Verifiable, Accurate Permeability Predictions, the Touch-Stone of All Models", SPE 30979, SPE Eastern Regional Conference & Exhibition, Morgantown, West Virginia, September 1995b.

10. Mowers, T. T., Budd, D. A.: "Quantification of Porosity and Permeability Reduction Due to Calcite Cementation Using Computer-Assisted Petrographic Image Analysis Techniques", AAPG Bulletin, v. 80, no. 3, March 1996 (p. 309-322).

11. Nelson, M. M., Illingworth, W. T.: *A Practical Guide to Neural Nets*, Addison-Wesley Publishing Company, Inc., 1991.

12. Olson, T. M., Babcock, J. A., Prasad, K. V. K., Boughton, S. D., Wagner, P. D., Franklin, M. H., Thompson, K. A.: "Reservoir Characterization of the Giant Hugoton Gas Field, Kansas", AAPG Bulletin, v. 81, no. 11, November 1997 (p. 1785-1803).

13. Opitz, D. W., Shavlik, J. W.: "Actively Searching for an Effective Neural Network Ensemble", Connection Science, v. 8, nos. 3 & 4, 1996 (p. 337-353).

14. Osborne, D. A.: "Permeability Estimation Using A Neural Network: A Case Study From The Roberts Unit, Wasson Field, Yoakum County, Texas", AAPG South West Section Transactions, 1992 (p. 125-132).

15. Rogers, S. J., Chen, H. C., Kopaska-Merkel, D. C., Fang, J. H.: "Predicting Permeability From Porosity Using Artificial Neural Networks", AAPG Bulletin, V. 79, No. 12, 1995 (p. 1786-1797).

16. Schlumberger Ltd.: "Log Interpretation Charts", Houston, Texas, 1987.

17. Timur, A.: "An Investigation of Permeability, Porosity, and Residual Water Saturation Relationship for Sandstone Reservoirs", The Log Analyst, V. 9, No. 4, July-August 1968 (p. 8).

18. Tixier, M. P.: "Evaluation of Permeability From Electric-Log Resistivity Gradients", Oil & Gas Journal, June 1949 (p. 113).

19. Wendt, W. A., Sakurai, S., Nelson, P. H.: "Permeability Prediction From Well Logs Using Multiple Regression:, *Reservoir Characterization*, Lake, L. W., and Caroll, H. B., Jr., Editors, Academic Press, New York, New York, 1985.

20. Wiener, J., Rogers, J., Moll, R.: "Predict Permeability From Wireline Logs Using Neural Networks", Petroleum Engineer International , May 1995 (p. 18-24).

21. Yao, C. Y., Holditch, S. A.: "Estimating Permeability Profiles Using Core and Log Data", SPE 26921, Proceedings, SPE Eastern Regional Conference, Pittsburgh, Pennsylvania, 1993.

22. Zell, A., Mamier, G., Vogt, M., Mache, N., Hübner, R., Döring, S., Herrmann, K.-U., Soyez, T., Schmalzl, M., Sommer, T., Hatzigeorgiou, A., Posselt, D., Schreiner, T., Kett, B., Clemente, G., Wieland, J.: *Stuttgart Neural Network Simulator User Manual, Version 4.1*, University of Stuttgart Institute for Parallel and Distributed High Performance Systems, June 1995.

23. Zhou, C. D., Wu, X., Cheng, J.: "Determining Reservoir Properties in Reservoir Studies Using a Fuzzy Neural Network", SPE 26430, Proceedings, SPE Annual Meeting, Houston, Texas, October 1993.