The Pennsylvania State University

The Graduate School

Department of Energy and Mineral Engineering

DEVELOPMENT AND TESTING OF AN ARTIFICIAL NEURAL NETWORK BASED HISTORY MATCHING PROTOCOL TO CHARACTERIZE RESERVOIR PROPERTIES

A Dissertation in

Petroleum and Natural Gas Engineering

by

Prasanna Chidambaram

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2009

The dissertation of Prasanna Chidambaram was reviewed and approved* by the following:

Turgay Ertekin
Graduate Program Chair of Petroleum and Natural Gas Engineering
Professor of Petroleum and Natural Gas Engineering
George E. Trimble Chair in Earth and Mineral Sciences
Dissertation Advisor
Chair of Committee

Michael Adewumi
Professor of Petroleum and Natural Gas Engineering
Quentin E. and Louise L. Wood Faculty Fellow in Petroleum and Natural Gas Engineering

Robert Watson
Associate Professor Emeritus of Petroleum and Natural Gas Engineering and Geo-Environmental Engineering

Sridhar Anandakrishnan
Associate Professor of Geosciences

*Signatures are on file in the Graduate School

ABSTRACT

History matching is one of the more critical steps in the reservoir performance predictions. It is during this step the reservoir parameters used in the model are adjusted until the reservoir model mimics actual reservoir behavior. A lot of times, there is no reliable way to measure some of the reservoir parameters required to build the model. This leads to gross approximation of these properties over the whole reservoir. Often these parameters are adjusted and readjusted until a good history match is obtained. Once a good history match is obtained, greater confidence can be placed on predictions made by the model.

The most common method of history matching is to make numerous simulation runs with each run using a different set of model parameters. The model parameters are varied in small steps in a trial-and-error fashion between each run until the observed production data matches with simulation model production data. This process is computationally intensive and time consuming. The number of simulation runs required to obtain a good history match also depends on the initial estimates of the parameters.

In this study, artificial neural networks are used to build a neuro-simulation tool for predicting properties like porosity, permeability, net pay thickness and two-phase relative permeability curves. Network uses cumulative production and pressure data as input to predict the history match parameters. Predictions made by the tool developed will give a good history match or the least serve as a good starting point to perform history match. The main advantage of using this tool is that the tool will not require an initial guess value for the parameters. This will remove the guess work involved in estimating some of the unknown parameters. The proposed artificial neural network will also reduce the actual number of simulation runs required to obtain a good history match when good estimate of model parameters are not available. The parameters predicted should provide a good history match or the least serve as good estimate for parameters that can be fine tuned to improve history match.

A commercial reservoir simulator was used to generate synthetic data necessary to train and validate the artificial neural network. The neural network developed can be used to predict reservoir properties of black oil reservoirs. Two separate networks were developed. A prediction network that will predict reservoir properties and the other, a network designer, that will provide design parameters required to build the prediction network.

The results of this study show that the prediction network as designed by the network designer is capable of predicting reservoir parameters within acceptable margins of error. It will also considerably reduce the number of simulation runs required to achieve a good history match, thus reducing the computational resources and time required for history matching process.

The developed tool was implemented to real field data from Perry reservoir located in Brayton Fields, west of Corpus Christi, Texas. Neuro-simulation tool was able to obtain a good history match with field production data. With just 50 simulation runs made to generate the training data for the network, it was able to predict the properties of the reservoir without any need for an initial estimate of the parameters.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A: area (ft^2)

Bo: oil formation volume factor (STB/RB)

Bg: gas formation volume factor (scf/RB)

Bw: water formation volume factor (STB/RB)

Gp: cumulative gas production (scf)

h: thickness (ft)

k: absolute permeability (md)

kr: relative permeability (fraction)

krgro: relative permeability of gas at residual oil saturation (fraction)

kroirw: relative permeability of oil at irreducible water saturation (fraction)

krocritg: relative permeability of oil at critical gas saturation (fraction)

krwro: relative permeability of water at residual oil saturation (fraction)

N: exponent used in computing two phase relative permeability (no units)

Np: cumulative oil production (STB)

P: pressure (psia)

Rs: solution gas-oil ration (scf/STB)

Sgcrit: critical gas saturation (fraction)

Sor: residual oil saturation (fraction)

Sorw: residual oil saturation in a two phase oil-water system (fraction)

Sorg: residual oil saturation in a two phase oil-gas system (fraction)

Swirr: irreducible water saturation (fraction)

T: temperature (^oF)

t: time (days)

ν: fluid velocity (cm/s)

Wp: cumulative water production (STB)

ϕ: porosity (%)

μ: viscosity (cp)

Abbreviations:

ANN: artificial neural network

FFBPN: feedforward backpropagation network

GLR: gas-liquid ratio

GOR: gas-oil ratio

PVT: pressure-volume-temperature

WOR: water-oil ratio

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis advisor and mentor Prof. Turgay Ertekin. He has been very supportive and helpful during challenging times both in my research and in my personal life. I am grateful for his valuable time and support. I would also like to extend my gratitude to Dr. Michael Adewumi, Dr. Robert Watson and Dr. Sridhar Anandakrishnan for their interest in serving as committee members and providing valuable suggestions.

Special thanks go to Dr. Sarma V. Pisupati for providing financial support through my entire stay at PennState. I really enjoyed my association with him working as teaching assistant.

I am indebted to my friends Deepa Narayanan, Claudia Parada, Atul Rangarajan, Pradeep Indrakanti and Denis Pone who have motivated me at all times.

I am thankful to my parents whose constant support, thoughtful encouragement and belief in me has helped me to successfully complete my studies.

Chapter 1

INTRODUCTION

The modeling of hydrocarbon reservoirs is a complex, multi-disciplinary task. Generally, reservoir simulation is considered to be the most powerful predictive tool available to the reservoir engineer. Building a reservoir model is a complicated task. One of the key hurdles in building a good reservoir simulation model is the lack of reliable data required to build the model. Once the model is built it has to be validated before it can be deployed for reservoir performance forecasting. Validation of the reservoir model is an important step in reservoir simulation, during which model production data are compared with field production data. This validation step is called history matching. It involves performing numerous simulation runs with slight variations to the reservoir parameters until a good match with field production data are obtained. A history matched reservoir model can be used to forecast reservoir performance with some confidence. History matching is benefitted by the knowledge and judgment of the simulation engineer. History matching is more of an art than science. History matching is a computationally intensive task that is time consuming and consequently costly.

Artificial neural network has been effectively used in several petroleum engineering applications. They are information processing systems that mimic the working of biological neurons present in the human brain. They are neither as complex nor as powerful as the biological neurons. But they can be used to solve complicated problems similar to history matching. Neural network is a soft computing technique that requires some training with known data. Once they are trained they can be used to solve similar problems at a much faster rate than by hard computing techniques like reservoir simulation.

In this study, a neuro-simulation tool is developed that can help reduce the number of simulation runs required to achieve a good history match. The neuro-simulation tool consists of two separate networks, a prediction network that will predict reservoir properties and the other, a network designer, which will provide artificial neural network design parameters required to build the prediction network. An inverse approach is used to solve the history match problem. Reservoir production and pressure data are used as input to the neural network. That information is processed by the network and predicts the porosity, permeability, thickness and endpoint saturation and relative permeability values. The prediction network developed is used to predict history match parameters for black oil models.

Synthetic production data are generated using a commercial reservoir model. These data are used to train and validate the neural network. Several sample cases starting from a simple square homogeneous reservoir with 1 well up to a complicated reservoir with irregular boundaries with 20 regions and 20 wells are examined.

The proposed prediction tool along with network designer provides an efficient way to perform history matching using artificial intelligence.

Chapter 2

LITERATURE REVIEW

2.1 History Matching

The modeling of hydrocarbon reservoirs is a complex, multi-disciplinary task. Generally, reservoir simulation is considered to be the most powerful predictive tool available to the reservoir engineer. It considers much more geologic and reservoir data than any other reservoir-prediction technique. Consequently, reservoir simulation has a much greater data requirement than other techniques. Once a satisfactory model is developed, it is used for predicting performance under a range of operating and maintenance scenarios, for planning development strategies and for assisting production operations [Ertekin et al., 2001; Parish et al., 1993].

An important stage in the development of reservoir simulation model is the validation of the model itself. In this step, the numerical model is tested to see how closely a given model can reproduce certain aspects of observed or measured data. This validation process is called history matching [Parish et al., 1993]. Validation of simulation model is the primary objective of history matching. During history matching, properties at each grid block are set such that the simulated well pressures and production data are as close as possible to the measured values. A good history match is generally achieved with numerous simulation runs, each run evaluating a different set of model parameters. It can be time-consuming and costly [Dye et al., 1986]. Once the historical production data are matched, a much greater confidence can be placed in the predictions made with the model [Ertekin et al., 2001]

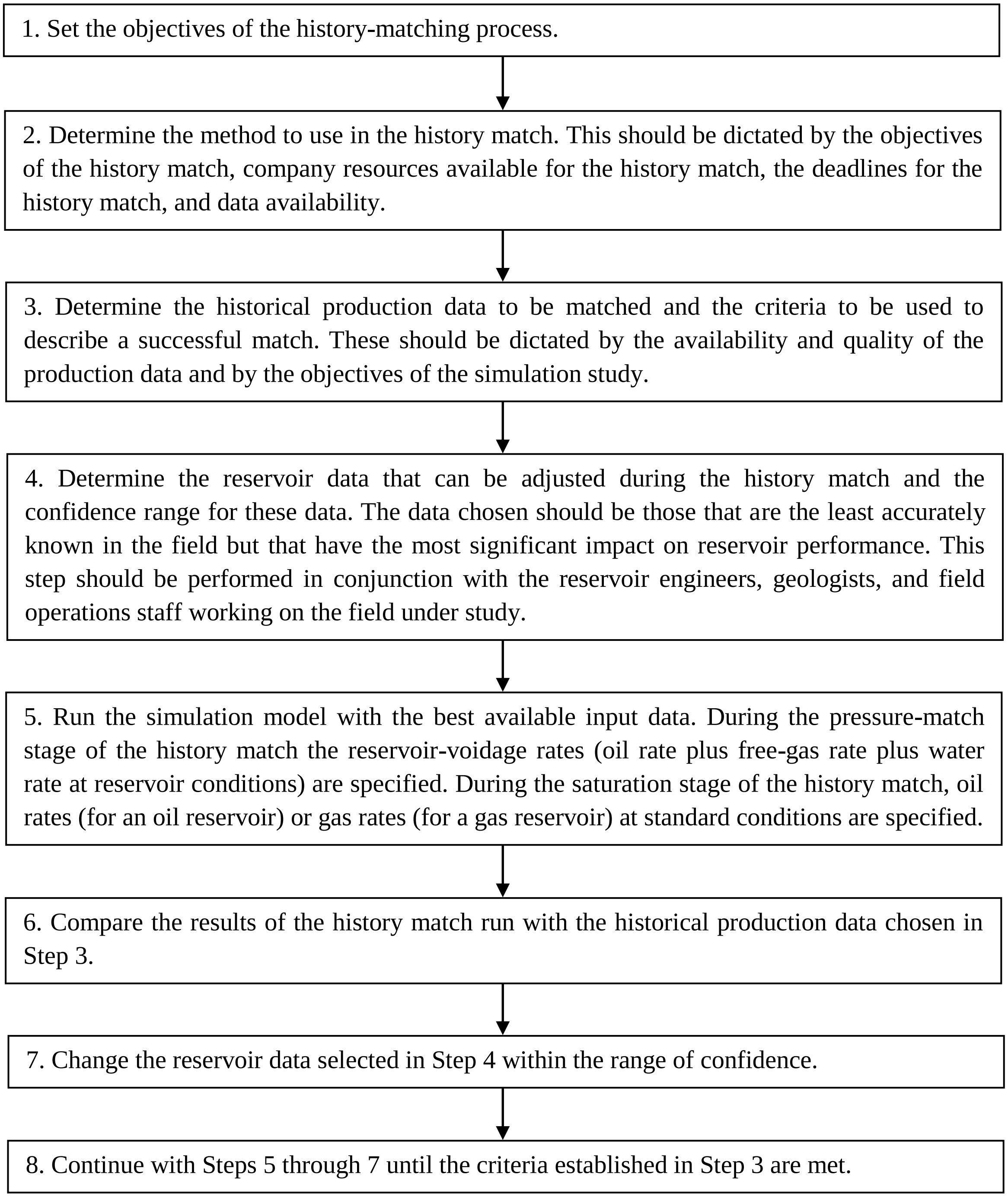
2.1.1 Objectives of History Matching

The primary objectives of history matching are to improve and to validate the reservoir simulation model. In general, the initial simulation input data does not produce results that match historical reservoir performance to a level that is acceptable for making accurate future forecasts. To improve the quality of the match, an iterative procedure involving adjusting initial input data are utilized. Figure **2-1** schematically shows the steps followed to adjust the initial simulation data systematically to provide an improved match [Ertekin et al., 2001].

Apart from the primary objectives, there are several beneficial byproducts to a successful history match. The very nature of history matching process may help understand the reservoir better with respect to level of aquifer support, paths of fluid migration, and areas of bypassed oil. This also means that a successful history match can identify opportunities to improve reservoir description and the data acquisition program. Finally, the history-matching process may identify unusual operating conditions. For example, if the water cut or GOR from an individual well appears to go against areal trends, problems (such as behind pipe communication) may be identified. On a field scale, areas of bypassed oil may be identified that can aid in an in-fill drilling program. Problems of this type are identified easily during a history match because the history-matching process forces engineers to look for areal and temporal trends in production data that may be overlooked otherwise [Ertekin et al., 2001].

2.1.2 Selection of History-Matching Method

Two approaches are commonly used for the history-matching process: manual and automatic history matching. Of the two methods, manual history matching is used more often [Ertekin et al., 2001].



1. Set the objectives of the history-matching process.

2. Determine the method to use in the history match. This should be dictated by the objectives of the history match, company resources available for the history match, the deadlines for the history match, and data availability.

3. Determine the historical production data to be matched and the criteria to be used to describe a successful match. These should be dictated by the availability and quality of the production data and by the objectives of the simulation study.

4. Determine the reservoir data that can be adjusted during the history match and the confidence range for these data. The data chosen should be those that are the least accurately known in the field but that have the most significant impact on reservoir performance. This step should be performed in conjunction with the reservoir engineers, geologists, and field operations staff working on the field under study.

5. Run the simulation model with the best available input data. During the pressure-match stage of the history match the reservoir-voidage rates (oil rate plus free-gas rate plus water rate at reservoir conditions) are specified. During the saturation stage of the history match, oil rates (for an oil reservoir) or gas rates (for a gas reservoir) at standard conditions are specified.

6. Compare the results of the history match run with the historical production data chosen in Step 3.

7. Change the reservoir data selected in Step 4 within the range of confidence.

8. Continue with Steps 5 through 7 until the criteria established in Step 3 are met.

Figure **2-1**: Overall iterative procedure for a history match [adapted from Ertekin et al, 2001].

In both methods, simulation model is run for the historical period considered and the results are compared with known field data. After comparison of the results, the simulation model parameters are modified in an effort to improve the match. In manual history matching, the parameters are modified by the simulation engineer. Thus, it requires knowledge of field under study, engineering judgment, and reservoir-engineering experience. While in automatic history matching, reservoir engineer is removed from picture and computer logic is used to modify the parameters. There are several approaches to automatic history matching, and each one attempts to minimize an error function. The error function is defined as a function of the difference between the observed reservoir performance and the simulated reservoir performance during the historical production period.

Both methods have their advantages and disadvantages. While manual history matching takes advantage of the engineering judgment and engineer's knowledge of the subject reservoir, it tends to be more time consuming. Automatic history matching is relatively faster but loses the advantage of knowledge learned during history matching. Thus selection of a specific method to use is defined by the objectives of the history match, the company resources devoted to the history match, and the deadlines of the simulation study.

2.1.3 Selection of Production Data to Specify and Match

In general, the selection of production data to specify depends on the stage of the history match and the hydrocarbons present in the reservoir. History-matching process generally is performed in a two-stage procedure. Mattax and Dalton refer to these two stages as gross and detailed, while Saleri and Toronyi refer to these stages as a pressure match and a saturation match. Regardless of the names, the objective of the first stage is to match average reservoir pressure and the objective of the second stage is to match individual well histories.

During the pressure-matching stage of the history match, the current reservoir-energy and reservoir-energy behavior with time are the most critical considerations. The most appropriate production data to specify during this stage of the history match are the historical well-voidage rates. The voidage rate is the sum of oil, free gas and water rates at reservoir conditions. During the saturation-matching stage of the history match, the production data to be specified are the oil production rates for an oil reservoir and the gas production rates for a gas reservoir. The selection of injection data to specify for injection wells is not as critical for production wells. In general, the specification of the historical surface injection rates is adequate for injection wells during all stages of the history match [Ertekin et al., 2001].

Selection of the production/injection data to be matched during a history match depends on the availability of production/injection data and the quality of the data. In general, the more data that can be matched during the history matching process, the more confidence that can be placed in the simulation model during the prediction stage of the study [Ertekin et al., 2001].

2.1.4 Adjusting Reservoir-Data to Match Historical Production

A fundamental concept of history matching is the concept of a “hierarchy of uncertainty.” The hierarchy of uncertainty is a ranking of model input data quality that lets the modeler determine which data are most and least reliable. When making changes to the model input, least reliable data should be changed first. Reliability of the data is determined when data are collected by evaluating the completeness and validity of the data [Fanchi, 1997, 2006; Raza, 1992; Saleri et al., 1992].

2.1.5 History Match Quality

There is no industry standard definition of what constitutes a successfully matched simulation model [Ertekin et al., 2001]. The definition of history match varies widely. A clear understanding of the study objectives should be used while defining what constitutes a successful history match. For example, pressure may be considered matched if the difference between calculated and observed pressures is within ±10% drawdown. The tolerance of ±10% is determined by estimating the uncertainty associated with measured field pressures and the required quality of study. A study demanding greater reliability in predictions may need to reduce the tolerance to ±5% or even less [Fanchi, 1997, 2006].

2.1.6 History-Match Limitations

The accuracy of the results is directly dependent upon the ability of the numeric model to emulate the reservoir. The first requirement of any simulation is a precise description of reservoir parameters over a large number of grid blocks. In most cases, field data does not permit such fine scale resolution of the parameters [Dye et al., 1986].

A vast majority of the reservoir remains unknown to the engineers and geologists working on the simulation study. Therefore, the initial data generally need to be adjusted, or tuned, for the simulation model to predict reservoir performance adequately [Ertekin et al., 2001]. The values obtained are not any 'true' average of the block properties of that reservoir, but are those which tend to compensate for the inaccuracies in the size and shape of logs in the model. Thus to some extent the parameters used are properties of the reservoir model and not properties of the reservoir itself [Dye et al., 1986]. Due to all these reason, no history-matching method, manual or automatic, guarantees a successful history match (one that meets all the history-matching objectives).

History matching process involves making numerous simulation runs of the reservoir with minor adjustments to input parameters until a satisfactory match is achieved. This makes history-matching time consuming and costly. In practice, a final match is often declared when the time or money allotted for the study is depleted [Fanchi, 1997, 2006].

The final history-matched model is not unique. In other words, several different history-matched models may provide equally acceptable matches to past reservoir performance but may yield significantly different future predictions [Ertekin et al., 2001].

2.2 Hard Computing-Reservoir Simulation

Reservoir simulation combines physics, mathematics, reservoir engineering, and computer programming to develop a tool for predicting hydrocarbon-reservoir performance under various operating conditions. The use of reservoir simulation as a predictive tool is becoming standard in the petroleum industry [Ertekin et al., 2001].

Reservoir modeling can be broadly classified into two as,

1. Black-oil simulation
2. Compositional simulation.

Black oil modeling is used for reservoir situations where fluid flow behavior is modeled using reservoir pressure and the effects of fluid phase composition on flow behavior do not need to be considered. The fluid is represented by a three component system (oil, water, gas) of constant composition.

Compositional simulator is used when an equation of state is required to describe reservoir fluid phase behavior or the compositional changes associated with depth. A compositional model is the right choice for studying condensates or volatile crude oils, gas injection programs, and secondary recovery studies. Knowledge of compositional behavior is also required for accurate planning and design of surface production facilities.

In both approaches, a form of mass balance equation is utilized. Temperature is assumed to be constant throughout the system. So, energy balance equations are not utilized. Advancements in technology and use of thermal recovery processes such as steam injection and in-situ combustion, has warranted the use of thermal simulators to model these operations. Thermal simulators use the compositional approach, where the energy-balance equation and the mass-balance equation are applied simultaneously.

2.3 Overview of Artificial Neural Networks (ANN)

Artificial neural network are information processing systems that are a rough approximation and simplified simulation of the biological neuron network system. In 1940's Warren McCulloch and Walter Pitts showed that networks of artificial neurons could, in principle, compute any arithmetic or logical function. Their work is often acknowledged as the origin of the neural network field. However, it was not until 1980's when ANN became popular due to the development of powerful computing systems [Hagan et al., 1996].

A biological neuron consists of three principle components: dendrites, cell body and axon as shown in Figure **2-2**. The tree-like structures are called dendrites. They are receptive networks of nerve fibers that carry electrical signals into the cell body. The cell body effectively sums and thresholds these incoming signals. The signal from the cell body is carried out to other neurons by the single long fiber called axon. The point where an axon of one cell and a dendrite of another cell are in contact is called synapses.

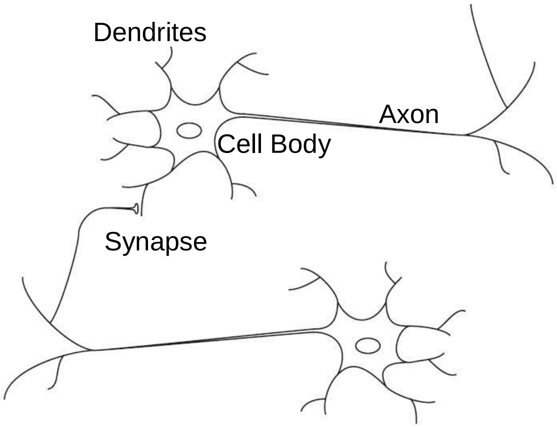


Figure **2-2**: Schematic Drawing of Biological Neurons [adapted from Hagan et al., 1996].

Artificial neural networks are neither as powerful as biological neurons in the brain nor are they as complex. They have two basic similarities between them, 1. Both are simple computational devices that are highly interconnected, 2. The connections between neurons determine the function of the network.

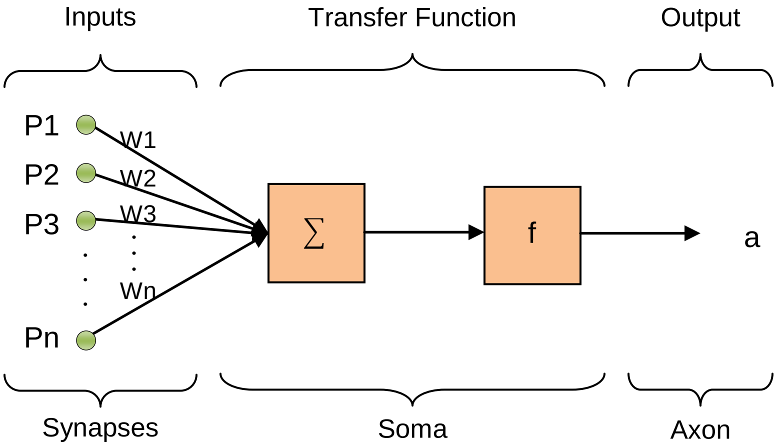
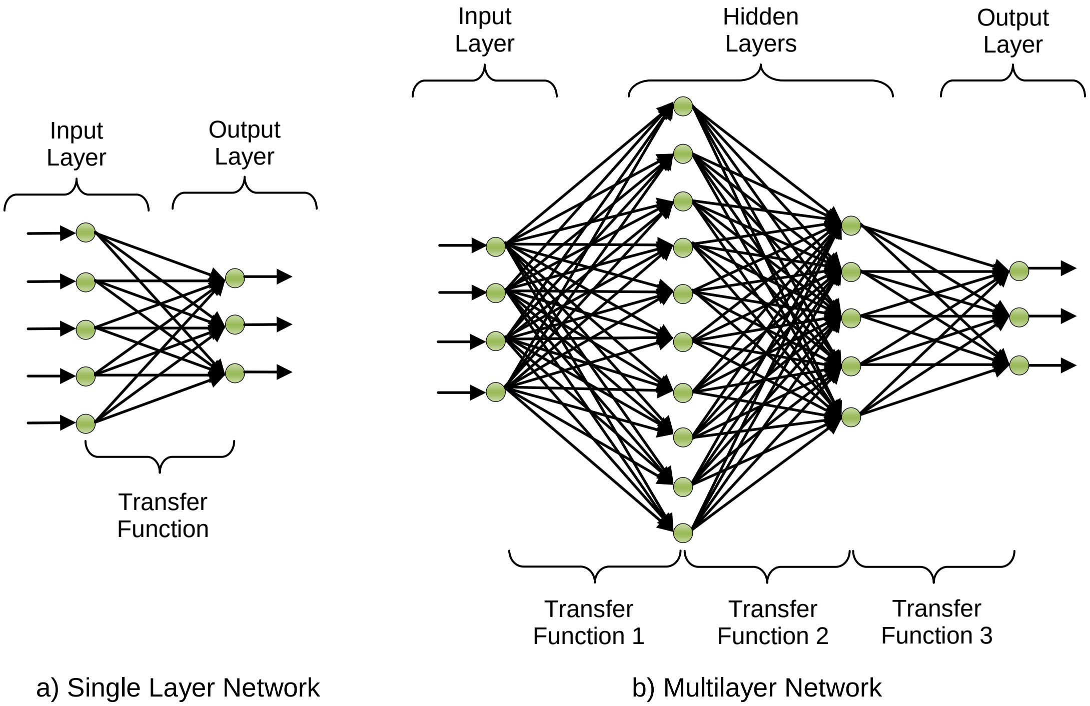


Figure **2-3**: Multiple Input Neuron.

A schematic of the multiple input neuron with n inputs is shown in Figure **2-3**. The individual inputs P1, P2, … Pn are each multiplied by corresponding elements W1, W2, … Wn to form W1*P1, W2*P2, … Wn*Pn. The weight corresponds to the strength of the synapses. The body of the neuron is represented by the summation of all W*P products and its modification by the transfer function ‘f’. The neuron’s output ‘a’ represents the electrical impulse carried through the axon [Hagan et al., 1996].

2.3.1 Artificial Neural Network Architecture

Typically the neurons in artificial neural network applications are arranged in layers. The arrangement of neurons into layers and the connections between them defines the network architecture. All the neurons in any particular layer perform similarly. Their behavior is conditioned by the transfer function and weights.



a) Single Layer Network b) Multilayer Network

Figure **2-4**: Classification of Network Architectures.

Networks are classified into two based on the number of layers they have, 1. Single layer, 2. Multilayer networks. Figure 2-4 shows typical architectures of a single layer and a multilayer network.

Single layer networks have an input layer and an output layer. While counting the number of layers in a network, input layer is not counted since its only task is to provide input data. Input layer doesn't perform any calculations. Multi-layer network has several layers which include an input layer, one or more hidden layers and an output layer. The hidden layers don't interact directly with the external surroundings of the net, hence the name hidden layers. The multilayer network shown in Figure **2-4** consists of three layers; two hidden layers and one output layer connected by three layers of weights.

2.3.1.1 Weights and Network Training

Weight corresponds to the strength of a synapse between 2 neurons. It is also referred to as synaptic weight. Initially, weights are set to either zero or a very small number. Those weights are changed using a learning rule during the iterative training process. A positive weight represents an excitatory stimulus while a negative weight corresponds to an inhibitory stimulus. A zero weight value indicates no connection or stimulus. The weight connections between layers of neurons are denoted as weight matrices 'W'. Typically, the matrix element w_{ij}, is used to denote the weight connecting the output of neuron *i* to the input of neuron *j* [Hagan et al., 1996].

Network training is performed by the use of learning algorithms. Network training process can be classified into three major types. They are, 1. Supervised learning, 2. Unsupervised learning and 3. Reinforced learning.

In supervised learning, pattern or input vector is provided with an associated target or output vector. The supervised learning can be thought of as learning with a 'teacher', in the form of a function that provides continuous feedback on the quality of solutions obtained thus far. Weights are updated continuously based on the feedback received about the quality of solutions obtained thus far. Supervised learning is used in pattern recognition and regression. Unsupervised learning uses only an input vector. The weights are modified so that similar input patterns are assigned to the same target [Fausett, 1994]. Unsupervised learning can be used for general estimation, estimation of statistical distribution and filtering. In reinforced learning, input data are not given, but generated by an agent's interactions with the environment. At each point in time t, the agent performs an action and the environment generates an observation and an instantaneous cost, according to some (usually unknown) dynamics. Reinforced learning is used in control problems, games and other sequential decision making tasks. In the present study supervised learning will be utilized with pattern or input and target or output generated using a commercial reservoir simulator.

2.3.1.2 Transfer Functions

Transfer function scales the response of an artificial neuron to an external stimulus and generates the neuron activation [Maren et al., 1990]. Transfer function can be either a linear or non-linear function. Any multilayer perceptron using a linear transfer function has an equivalent single-layer network; a non-linear function is therefore necessary to gain the advantages of a multi-layer network [Fausett, 1994 and Maren et al., 1990].

The output (a) of a linear transfer function is equal to its input (n):

$$a=n \tag{2.1}$$

The purelin transfer function is shown in the left of Figure **2-5**, while the output (a) versus input (p) characteristic of a single-input linear neuron with a bias is shown on the right of Figure **2-5**. This function commonly applied to the output layer since it allows the network to produce its output within the desired limits without having to de-normalize them.

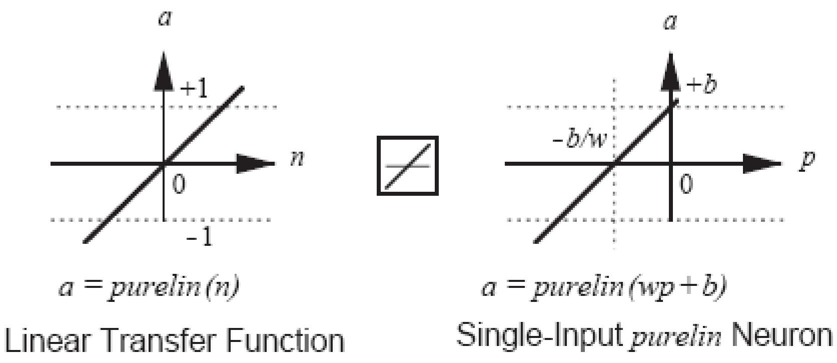


Figure **2-5**: Linear Transfer Function [reproduced from Hagan et al., 1996].

Log-sigmoid and hyperbolic tangent sigmoid are the transfer functions commonly used in multilayer networks using the back-propagation algorithm [Hagan et al., 1996]. Log-sigmoid transfer function takes the input (which may have any value between plus and minus infinity) and scales its output to range in between 0 and 1. The output (a) of a log-sigmoid transfer function is calculated according to the expression:

$$a = \frac{1}{1+e^{-n}} \tag{2.2}$$

Log-sigmoid transfer function is shown in the left of Figure **2-6**, while the output (a) versus input (p) characteristic of a single-input linear neuron with a bias is shown on the right of Figure **2-6**.

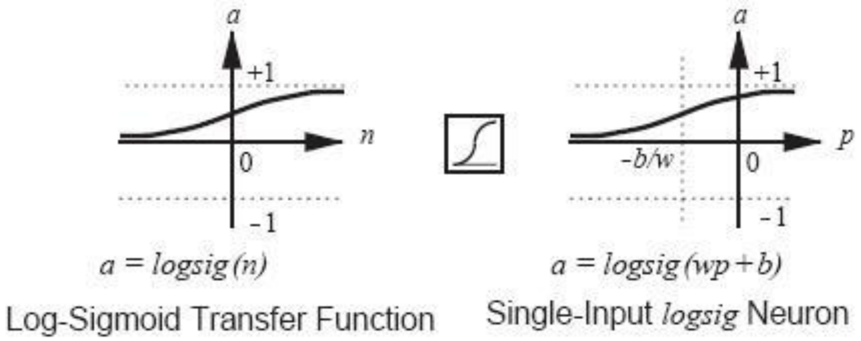


Figure **2-6**: Log-Sigmoid Transfer Function [reproduced from Hagan et al., 1996].

Hyperbolic tangent sigmoid transfer function has an advantage over log-sigmoid function of being able to deal directly with negative numbers. The output (*a*) of a hyperbolic tangent sigmoid transfer function is calculated according to the expression:

$$a = \frac{e^n - e^{-n}}{e^n + e^{-n}} \qquad 2.3$$

Hyperbolic tangent sigmoid transfer function is shown in the left of Figure **2-7**, while the output (*a*) versus input (*p*) characteristic of a single-input linear neuron with a bias is shown on the right of Figure **2-7**.

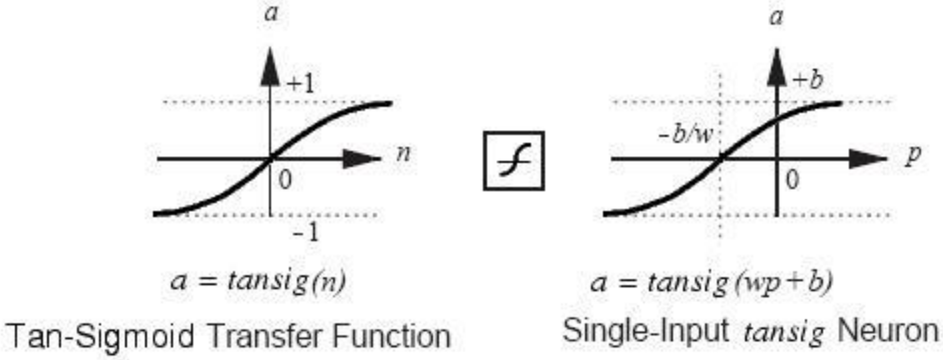
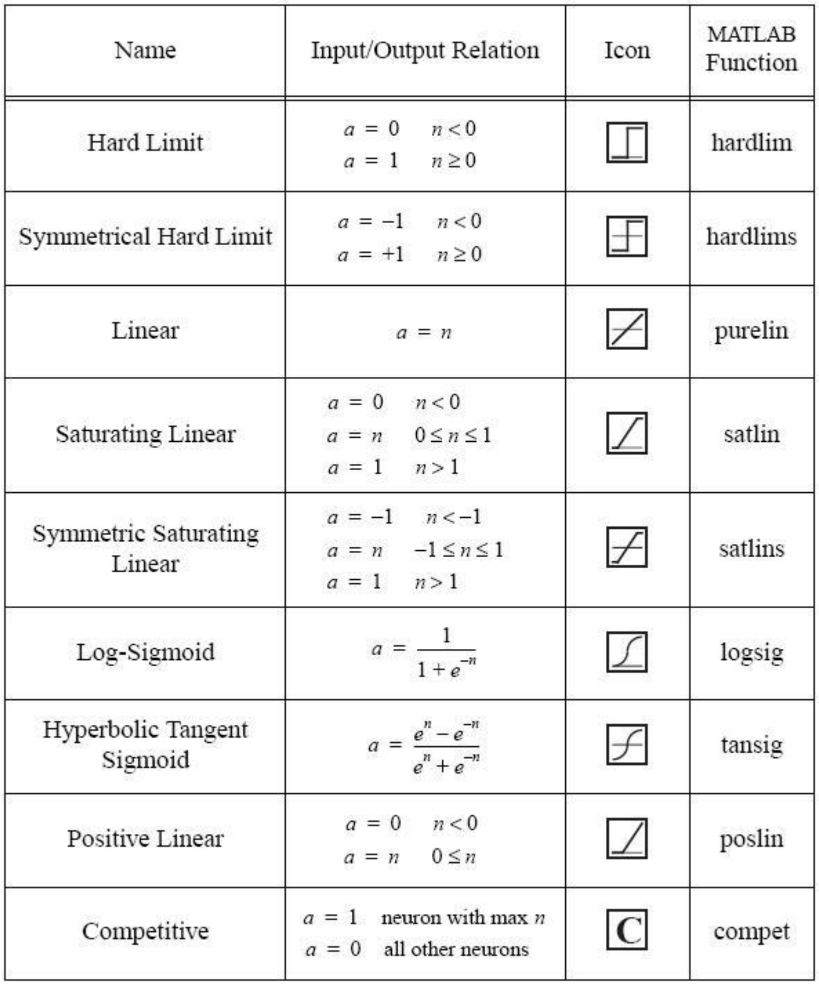


Figure **2-7**: Hyperbolic Tangent Sigmoid Transfer Function.

Table 2-1 shows a list of other commonly used transfer functions. However, it should be noted that in the present work multilayer networks with sigmoid functions are proved to be more appropriate for our problem. Previous works, as the one presented by Ramgulam (2006), have also shown that these types of architectures are more suitable for application to similar reservoir engineering problems.

Table **2-1**: Transfer Functions [reproduced from Hagan et al., 1996].



Name	Input/Output Relation	Icon	MATLAB Function
Hard Limit	$a = 0 \quad n < 0$ $a = 1 \quad n \geq 0$		hardlim
Symmetrical Hard Limit	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$		hardlims
Linear	$a = n$		purelin
Saturating Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n \leq 1$ $a = 1 \quad n > 1$		satlin
Symmetric Saturating Linear	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = 1 \quad n > 1$		satlins
Log-Sigmoid	$a = \frac{1}{1 + e^{-n}}$		logsig
Hyperbolic Tangent Sigmoid	$a = \frac{e^{n} - e^{-n}}{e^{n} + e^{-n}}$		tansig
Positive Linear	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$		poslin
Competitive	$a = 1$ neuron with max n $a = 0$ all other neurons	C	compet

2.3.2 Multilayer Feedforward Network with Backpropagation

Multilayer feedforward network with backpropagation is the most widely used network architecture [Maren et al., 1990, Patterson, 1995]. Feedforward networks are the simplest artificial neural networks. It has no feedback. This means there are no connections that loop. Backpropagation is a supervised learning method. It is an implementation of the delta rule. The term ‘backpropagation’ is an abbreviation for ‘backwards propagation of errors’. Feedfoward network are most benefited when working with backpropagation since it adds the component of feedback to the network.

Feedforward networks with backpropagation are easy to implement, trains faster than other types of networks and solves many types of problems correctly [Centilmen, 1999]. They operate in two steps. First is the feedforward step. During this step, input pattern is presented to the input layer and the information is transferred through hidden layer(s) to the output layer. Transfer functions process the information as they move from layer to layer. Second is the backpropagation step when, backpropagation is used to calculate the gradient of the error of the network with respect to the network's modifiable weights. During this step, networks response is compared to the desired output and the errors are propagated from the output layer to the inner layers. This error signal is used to adjust the network weights. Each intermediate layer receives a portion of the total error signal based roughly on the relative contribution of the unit made to the original output [Ali, 1994]. Thus after several iterations of this process, the error signal generated becomes small. At this stage the network is considered trained for the intended purpose. It must be able to make predictions from a novel set of inputs.

2.3.3 Convergence and Training Efficiency

Convergence problem refers to a situation where the total error of the current iteration is lower than the one from previous iteration. There can be several causes for convergence problems. The most common cause is the presence of several local minima on the error surface. This problem can be prevented by using a momentum parameter. Using a high momentum parameter can also help to increase speed of convergence. But care should be taken not to use a very high momentum since this may lead to overshooting the actual minimum thus making the network unstable. The optimization method used may not be guaranteed to converge when the system is far away from the local minimum.

Learning efficiency can be improved by taking the following measures, 1. Using high momentum parameter, 2. Using functional links, 3. Using faster learning functions. As discussed earlier use of high momentum parameter can improve the speed of learning. Use of functional links in the input and/or output layers usually helps the network interpret the data better and thus improves the learning efficiency. Training and learning functions are mathematical procedures used to automatically adjust the network's weights and biases. The training function dictates a global algorithm that affects all the weights and biases of a given network. Gradient decent is the most commonly used learning method. Other learning algorithms like conjugate gradient methods, the Levenberg-Marquardt algorithm (LM) can be used to make the learning faster. Levenberg-Marquardt algorithm is one of the fastest backpropagation algorithms, and is highly recommended as a first-choice supervised algorithm, although it does require more memory than other algorithms.

2.3.4 Application of Neuro-Simulation

Neuro-simulation is a technique that combines soft computing techniques with hard computing techniques. The main advantage of utilizing this technique is to reduce simulation times and also to reduce computational resource required. Hard computing is usually used to generate the necessary pattern/target data used in training. And soft computing is usually used to learn the internal relationship between the pattern and target. The trained net can be used to simulate and predict performance for scenarios that are different from the ones used in training.

Neuro-simulation can be used for a variety of petroleum engineering applications. Some work done in the past include exploring field development strategies in conjunction with various recovery schemes [Doraisamy et al., 1998], predicting natural gas production [Al-Fattah, 2001], development of proxy model for gas condensate reservoir exploitation [Ayala, 2004], characterization of carbon dioxide sequestration and coalbed methane projects [Gorucu et al., 2005], optimization of history matching [Ramgulam et al., 2007], screening and designing improved oil recovery methods [Parada, 2008] , optimizing design of cyclic pressure pulsing in naturally fractured reservoirs [Artun, 2008] just to name a few.

In the present study, Artificial Neural Network (ANN) and reservoir simulator are the soft computing and hard computing techniques used respectively. Reservoir simulator is used to generate reservoir production history for several sets of reservoir properties. This constitutes the pattern/target data that will be used while training the neural network. A successfully trained network can be used to predict the properties for reservoir that are different from the ones used during training.

Chapter 3

PROBLEM STATEMENT

History matching is one of the more critical steps in the reservoir performance predictions. It is during this step the reservoir parameters used in the model are adjusted until the reservoir model mimics actual reservoir behavior. A lot of times, there is no reliable way to measure some of the reservoir parameters required to build the model. Especially in the early life of the reservoir the limited set of parameters available are from a few well locations. This leads to gross approximation of these properties over the whole reservoir. At this point engineering judgment and experience of the engineer from working with similar or adjacent fields/reservoirs is also utilized in estimating some of the parameters. Often these parameters are adjusted and readjusted until a good history match is obtained.

The most common method of history matching is to make numerous simulation runs with each run using a different set of model parameters. The model parameters between each run are varied in small steps in a trial-and-error fashion until the observed production data matches with simulation model production data. This process is computationally intensive and time consuming. The number of simulation runs required to obtain a good history match also depends on the initial estimate of the parameters. When a good estimate of the parameters is available a good history match can be obtained with fewer simulation runs.

The main objective of this study is to develop an artificial neural network tool that will be able to predict the parameters required to build the reservoir model but not suffer from some of the problems discussed earlier. The proposed tool will use production and pressure data from the actual reservoir as input to predict the reservoir model parameters. The proposed tool will not require an initial guess value for the parameters. This will remove the guess work involved in estimating some of the unknown parameters. The proposed artificial neural network will also reduce the actual number of simulation runs required to obtain a good history match when good estimate of model parameters are not available. The parameters predicted should provide a good history match or the least serve as good estimate for parameters that can be fine tuned to improve history match.

The proposed artificial neural network tool will predict porosity, permeability and net pay thickness of each zone/region in the reservoir. It will also predict the endpoint saturations, relative permeability at endpoint saturations and exponent values for the oil-water and oil-gas two phase relative permeability curves.

There are no established guidelines for use of soft computing techniques like artificial neural network to history matching application. This study intends to establish guidelines and suggestions for the development of artificial neural network that can provide a good history match or act as good starting point for history matching procedure.

Chapter 4

RESERVOIR MODEL

Numerical reservoir simulation is an industry standard tool used by petroleum engineers to model hydrocarbon reservoirs. A successful history matched model is usually used to forecast performance of the reservoir. They are also used as a tool to design an optimum field development plan by simulating several possible scenarios. In our study, numeric reservoir simulation will be used to generate the necessary pattern/target data that will be used in the artificial neural network development.

4.1 Properties

Multi-phase black oil model was used to simulate the depletion of hydrocarbons in the reservoir. Synthetic production data were generated using CMG IMEX[1] (black oil simulator) for various sets of porosity (ϕ), permeability (k), net pay thickness (h), and relative permeability curves. Part of the generated data is used to train and the rest is used to validate the artificial neural network.

In order to build a reservoir model, it is essential to know the rock properties, fluid properties, formation structure, well design parameters, initial reservoir conditions and well control mechanism. To train and develop the proposed neural network, it is essential to create a set of reservoirs with different sets of porosity (ϕ), permeability (k), net pay thickness (h), and relative permeability curves, while keeping the other parameters uniform. In the initial study, only

[1] CMG is a commercial reservoir simulator developed by Computer Modeling Group Ltd. Calgary, Canada. IMEX is their black oil simulator.

porosity (ϕ), permeability (k) and net pay thickness (h) values are varied. Later, relative permeability curves are also varied. Relative permeability curves are varied by varying endpoint saturation (S_{orw}, S_{org}, S_{wirr}, S_{gcrit}), relative permeability at endpoint saturation (k_{rwro}, k_{roirw}, k_{rgro}, $k_{rocritg}$) and exponent (N) values.

4.1.1 Porosity (ϕ)

Porosity is the fraction of a porous medium that is void space. It is measured as a fraction and posses no units. Porosity of consolidated materials depends mainly on the degree of cementation. The porosity of unconsolidated materials depends on the packing of the grains, their shape, arrangement and size distribution. Graton and Fraser (1935) analyzed the porosity of various packing arrangements of uniform spheres. The least compact arrangement of uniform spheres is that of cubical packing with a porosity of 47.6%. Table **4-1** gives typical porosity values for various materials. In this study, porosity values ranging from 5% to 50% is used.

Table **4-1**: Typical Porosity Values of Natural Sedimentary Materials [adapted from Bear, 1972].

Sedimentary Material	Porosity Value (percent)	Sedimentary Material	Porosity Value (percent)
Peat Soil	60-80	Fine-to-medium mixed sand	30-35
Soils	50-60	Gravel	30-40
Clay	45-55	Gravel and sand	30-35
Silt	40-50	Sandstone	10-20
Medium-to-coarse mixed sand	35-40	Shale	1-10
Uniform sand	30-40	Limestone	1-10

4.1.2 Permeability (*k*)

Permeability is a measure of rock's ability to transmit fluids. It may also be defined as the measure of the connectivity of pore spaces. It is commonly measured in darcy (*d*) or millidarcy (*md*). Permeability has dimensions of L^2 where L is a unit of length. Permeability is a rock property and it depends on the rock type. Productive sandstone reservoirs usually have permeability in the range of 10 md to 1000 md [Fanchi, 2006]. In this study, permeability values ranging from 10 md to 2000 md is used.

4.1.3 Net Pay Thickness (*h*)

The reservoir rock thickness capable of producing commercial hydrocarbons within a specified interval is called net pay thickness. It is commonly measured in feet (*ft*). Thickness of reservoir can vary widely from reservoir to reservoir and also within the same reservoir. Net pay thickness of a reservoir can vary anything between a few feet to few hundred feet. In this study, net pay thickness values ranging from 10 ft to 1000 ft is used.

4.1.4 Relative Permeability Curves

When there is only one fluid flowing through the porous medium, permeability of the porous medium to the fluid is the absolute permeability of the medium. When two or more fluids flow simultaneously through the porous medium, flow path has to be shared by more than one fluid. Thus effective permeability of the porous medium to each fluid is less than absolute permeability of the porous medium. Relative permeability is a dimensionless measure of this effective permeability of each phase. Relative permeability value varies between zero and one.

Typically, relative permeability of each phase for a black oil model is calculated from two-phase relative permeability. Two-phase relative permeability curves, oil-water relative permeability and oil-gas relative permeability are constructed from endpoint saturations (S_{orw}, S_{org}, S_{wirr}, S_{gcrit}), relative permeability at endpoint saturation (k_{rwro}, k_{roirw}, k_{rgro}, $k_{rocritg}$) and exponent (N) values using the following equations.

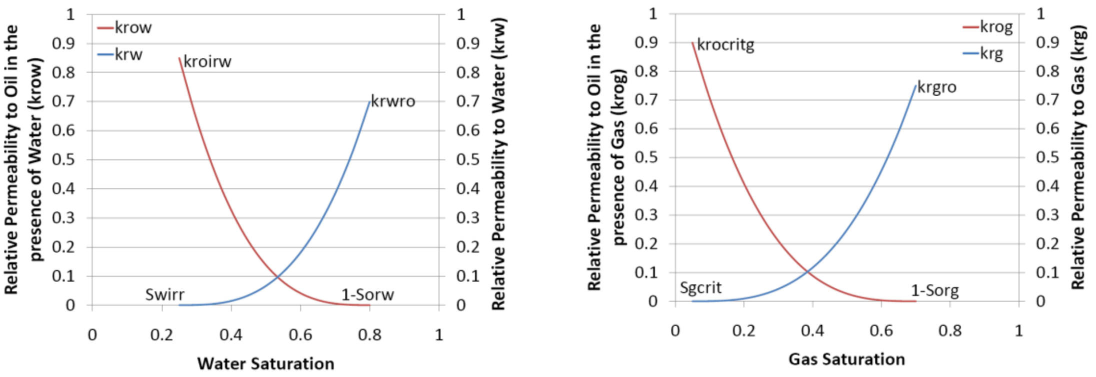
$$krw = krwro \times \left(\frac{Sw - Swirr}{1 - Swirr - Sorw}\right)^{N} \ldots\ldots\ldots\ldots\ldots\ldots 4.1$$

$$krow = kroirw \left(\frac{1 - Sw - Sorw}{1 - Swirr - Sorw}\right)^{N} \ldots\ldots\ldots\ldots\ldots\ldots 4.2$$

$$krg = krgro \left(\frac{Sg - Sgcrit}{1 - Sgcrit - Sorg}\right)^{N} \ldots\ldots\ldots\ldots\ldots\ldots 4.3$$

$$krog = krocritg \left(\frac{1 - Sg - Sorg}{1 - Sgcrit - Sorg}\right)^{N} \ldots\ldots\ldots\ldots\ldots\ldots 4.4$$

Figure **4-1** shows two-phase relative permeability curves generated using the above equations.



a) Oil-water two phase system

b) Oil-gas two phase system

Figure **4-1**: Two phase relative permeability curves.

4.1.4.1 Endpoint Saturations (S_{orw}, S_{org}, S_{wirr}, S_{gcrit})

When more than one fluid is present in a porous medium, the fraction of pore volume occupied by each fluid is called saturation. Oil remaining in the reservoir rock after the flushing or invasion process is called residual oil saturation (S_{or}). Residual oil saturation in a two phase oil-water system is represented by S_{orw} and in a two phase oil-gas system it is represented by S_{org}. The fraction of the pore volume occupied by water in a reservoir at maximum hydrocarbon saturation is called irreducible water saturation (S_{wirr}). The value of saturation of the specific gas phase at which the gas will first begin to flow as the saturation is increased is called critical gas saturation (S_{gcrit}). In this study, residual oil saturation and irreducible water saturation ranging from 0.10 to 0.40 along with critical gas saturation ranging from 0.00 to 0.05 are used.

4.1.4.2 Relative Permeability at Endpoint Saturations (k_{rwro}, k_{roirw}, k_{rgro}, $k_{rocritg}$)

Relative permeability at endpoint saturations represents the maximum relative permeability. These values are used to construct two-phase relative permeability curves. Relative permeability of water in an oil-water system at residual oil saturation (S_{orw}) is represented by k_{rwro}. Relative permeability of oil in an oil-water system at irreducible water saturation (S_{wirr}) is represented by k_{roirw}. Relative permeability of gas in an oil-gas system at residual oil saturation (S_{org}) is represented by k_{rgro}. Relative permeability of oil in an oil-gas system at critical gas saturation (S_{gcrit}) is represented by $k_{rocritg}$. Relative permeability value varies between zero and one. In this study, relative permeability at endpoint saturations ranging from 0.5 to 0.9 are used.

4.1.4.3 Exponent (N)

The curvature of the relative permeability curves is defined by exponent (N). In this study, exponent values ranging from 2 to 4 are used.

4.1.5 Synthetic Reservoir Data Generation Strategy

As discussed earlier, for the initial studies a set of reservoirs with different porosity, permeability and net pay thickness is created while keeping all the other properties and parameters same.

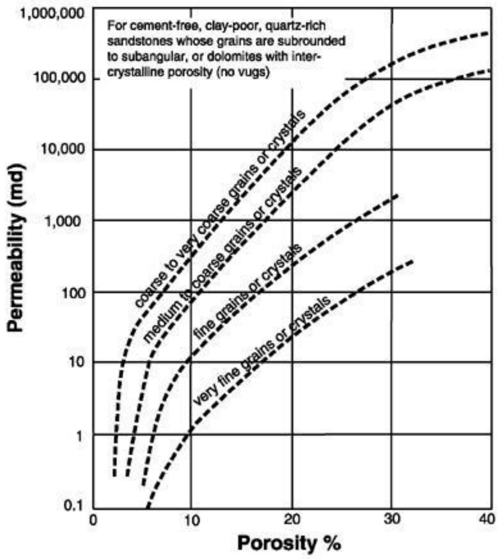


Figure **4-2**: Effect of grain size on permeability and porosity [adapted from Coalson et al., 1990].

To create several unique sets of these three properties, a random number generator is used. Those randomly generated values are then matched up into sets containing all three properties. After creating these sets, they are normalized to eliminate sets of properties that fall outside the typical permeability-porosity combinations. Cross plots of porosity and permeability of sand stone created by Coalson (1990), Hartman (2000) is used as reference in the process. Figure **4-2** shows cross plot of permeability versus porosity showing effect of grain size.

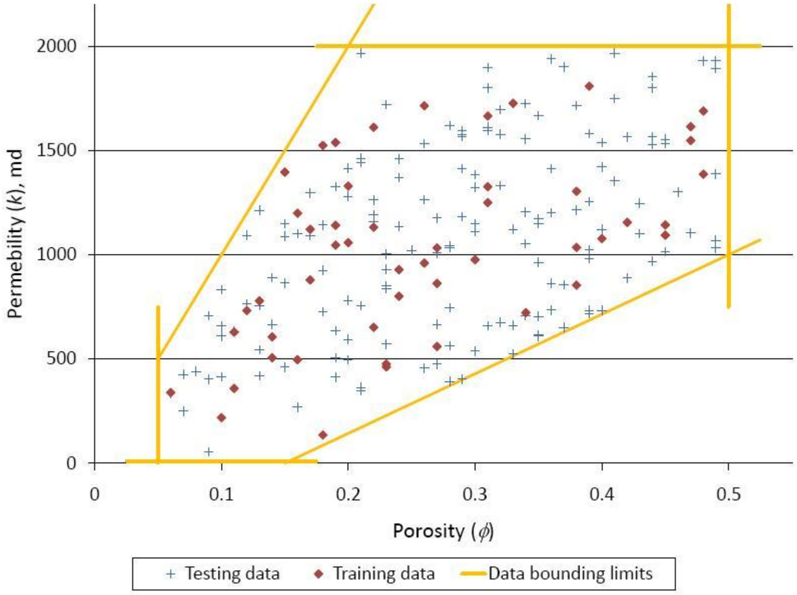


Figure **4-3**: Permeability versus porosity used for reservoir model creation.

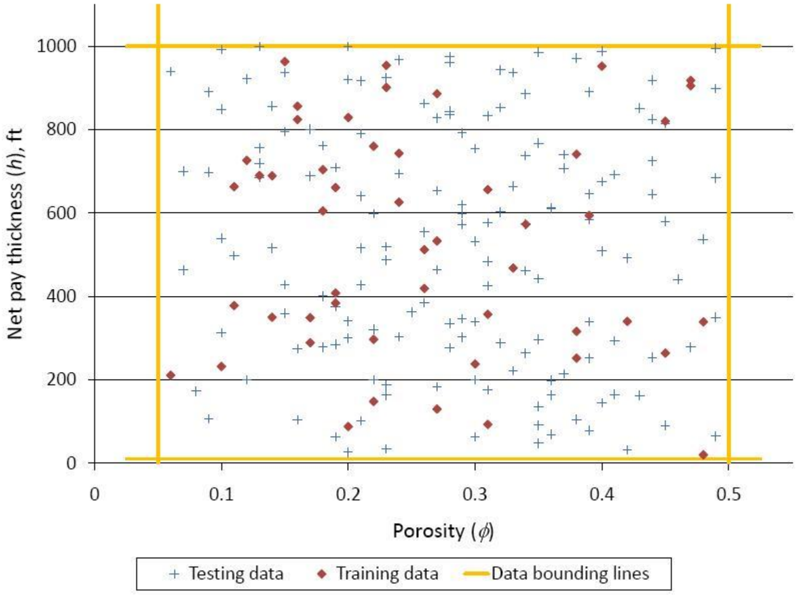


Figure **4-4**: Net pay thickness versus porosity used for reservoir model creation.

Finally, 200 unique sets of porosity, permeability and net pay thickness values are generated. These properties are be used to build reservoir models that are used to generate synthetic production data. Of them 50 data sets are used for training and 150 data sets are used for testing the artificial neural network. Figures **4-3**, **4-4** show plots of permeability and net pay thickness versus porosity values that are used in this study.

Initial reservoir conditions, rock properties and fluid properties required for building reservoir models are included in Appendix **A**. All these data are necessary to create reservoir model.

4.2 Case Studies

In order to build a reservoir model, the reservoir simulation engineer must know the rock properties, fluid properties, initial conditions, operating conditions, reservoir architecture, and drilling and completion information. All these properties except reservoir architecture and well information have been discussed in Section **4.1**. This section discusses the reservoir architecture and well information for each reservoir that is studied. The reservoir models that are studied can be broadly classified into three based on the shape of their boundaries. They are,

1. Multiphase reservoir with square boundaries
2. Multiphase reservoir with rectangular boundaries
3. Multiphase reservoir with uneven boundaries

Present study is initiated with a simple system that has one well and one region with homogeneous properties. Complexity of the model is increased progressively by increasing the number of wells and regions. Complex reservoirs with up to 20 wells and 20 regions are studied.

Based on the number of wells and regions present in a reservoir, they can be further classified into,

1. Reservoir with number of wells equal to number of regions
2. Reservoir with number of wells greater than number of regions
3. Reservoir with number of wells less than number of regions

The simplest system studied is a multiphase reservoir with square boundaries that has one well and one region with homogeneous properties and the most complex reservoir model studied is a multiphase reservoir with uneven boundaries that has 20 wells and 20 regions.

4.2.1 Case 1: Multiphase Reservoir with Square Boundaries

A multiphase black oil model with square boundary constitutes the first case studied. These reservoirs have square reservoir boundary that is discretized into 43 x 43 grid blocks. Each grid block measures 100 ft x 100 ft. The number of wells in the reservoir is varied from 1 to 20 and so are the regions with different set of porosity, permeability and net pay thickness. Thus, creating 400 different possible combinations based on number of wells and regions in the reservoir. All these are studied as separate types of reservoirs. Figure **4-5** shows a square reservoir with 20 wells and one region. All of the 400 different reservoirs studied under this case share the same well locations. Reservoirs that have less than 20 wells share the same well locations but would eliminate the excess wells. For example, in a reservoir with 7 wells they are located at points marked 1 through 7 in Figure **4-5**. Rest of the well locations 8 through 20 are not used in the model. Similar to the wells, the reservoirs built may have up to 20 regions. Figure **4-6a** and **4-6b** shows how the reservoir is divided into regions in this study.

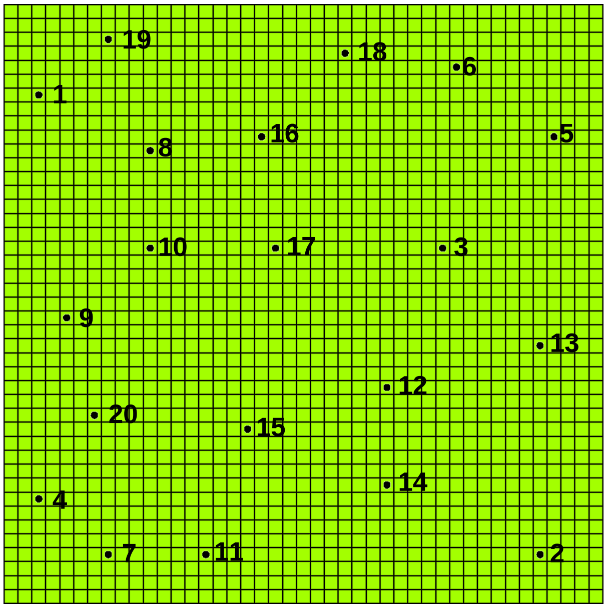


Figure **4-5**: 43 x 43 square reservoir with one region and 20 wells.

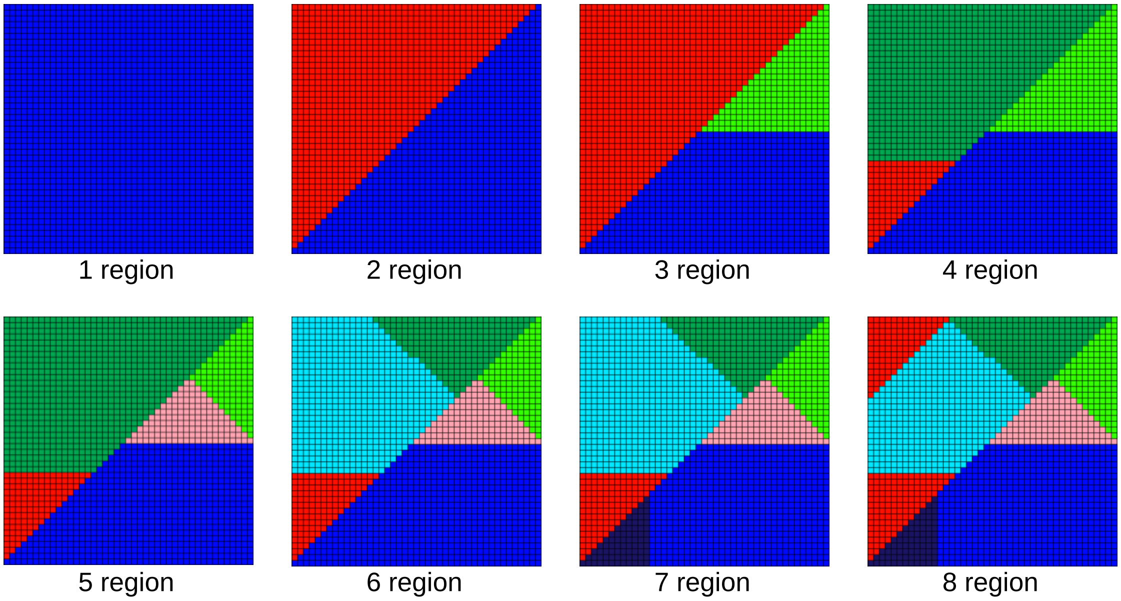
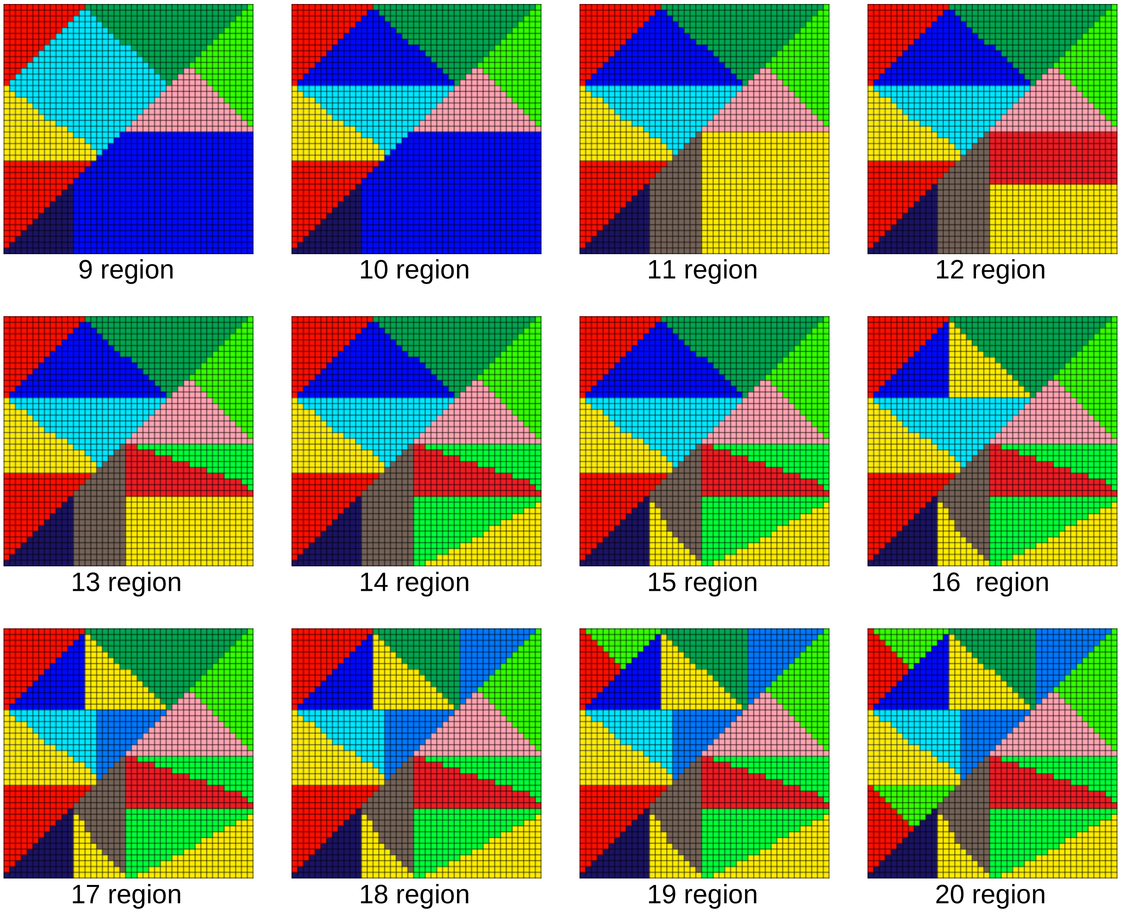


Figure **4-6a**: 43 x 43 square reservoir divided into regions.



9 region

10 region

11 region

12 region

13 region

14 region

15 region

16 region

17 region

18 region

19 region

20 region

Figure **4-6b**: 43 x 43 square reservoir divided into regions (cont’d).

As discussed in Section **4.1**, the 200 unique sets of properties are used to build reservoir models used in this study. Thus, there are 200 unique reservoirs for each of the 400 different types of models discussed. Production and pressure data from these simulation runs constitute the data required to build the artificial neural network.

4.2.2 Case 2: Multiphase Reservoir with Rectangular Boundaries

Second case studied uses a reservoir that is a little more complex by changing the reservoir boundaries from square to a rectangle. These reservoirs have a rectangular reservoir boundary that is discretized into 70 x 43 grid blocks. Each grid block measures 100 ft x 100 ft. Similar to the square reservoir the number of wells and regions in this case is also varied from 1 through 20. Figure **4-7** shows a rectangle reservoir with 20 wells and one region. All the 400 different reservoirs studied under this case share the same well locations. Similar to previous case, reservoirs that have less than 20 wells share the same well locations but eliminate the excess wells.

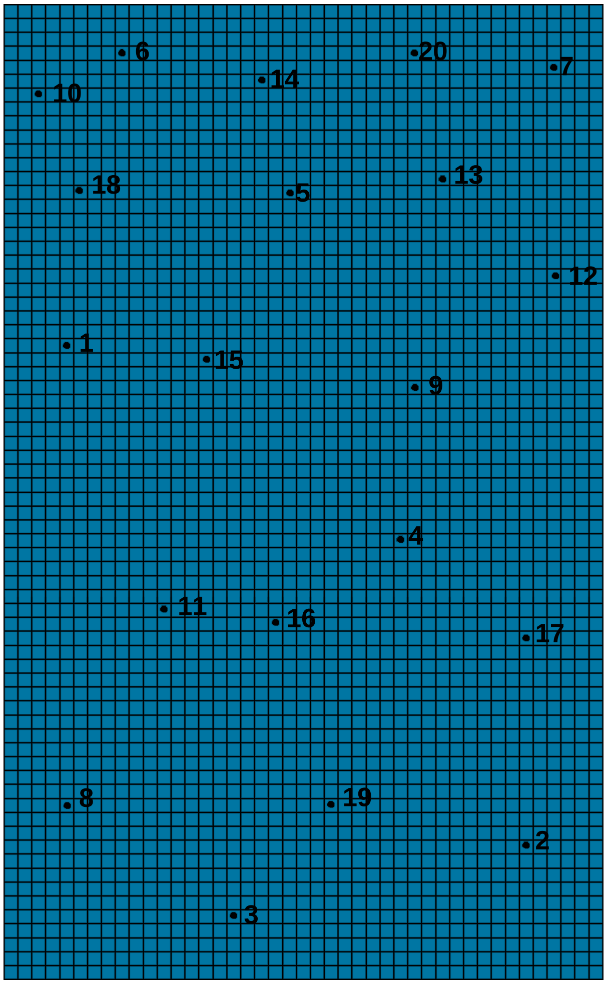
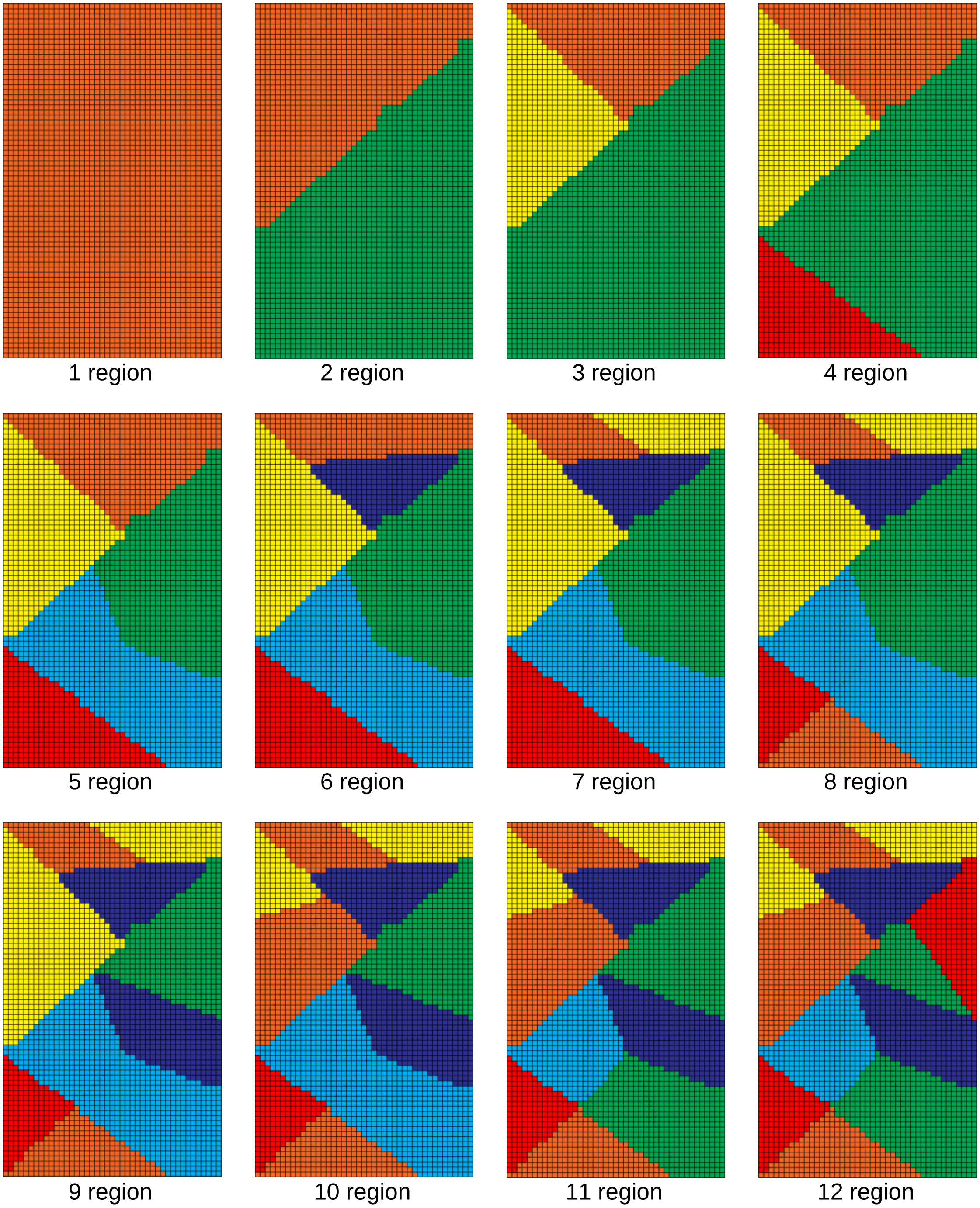


Figure **4-7**: 70 x 43 rectangular reservoir with one region and 20 wells.

Similar to the wells, the reservoirs built may have up to 20 regions. Figure **4-8a** and **4-8b** shows how the rectangular reservoir is divided into regions in this study.



1 region

2 region

3 region

4 region

5 region

6 region

7 region

8 region

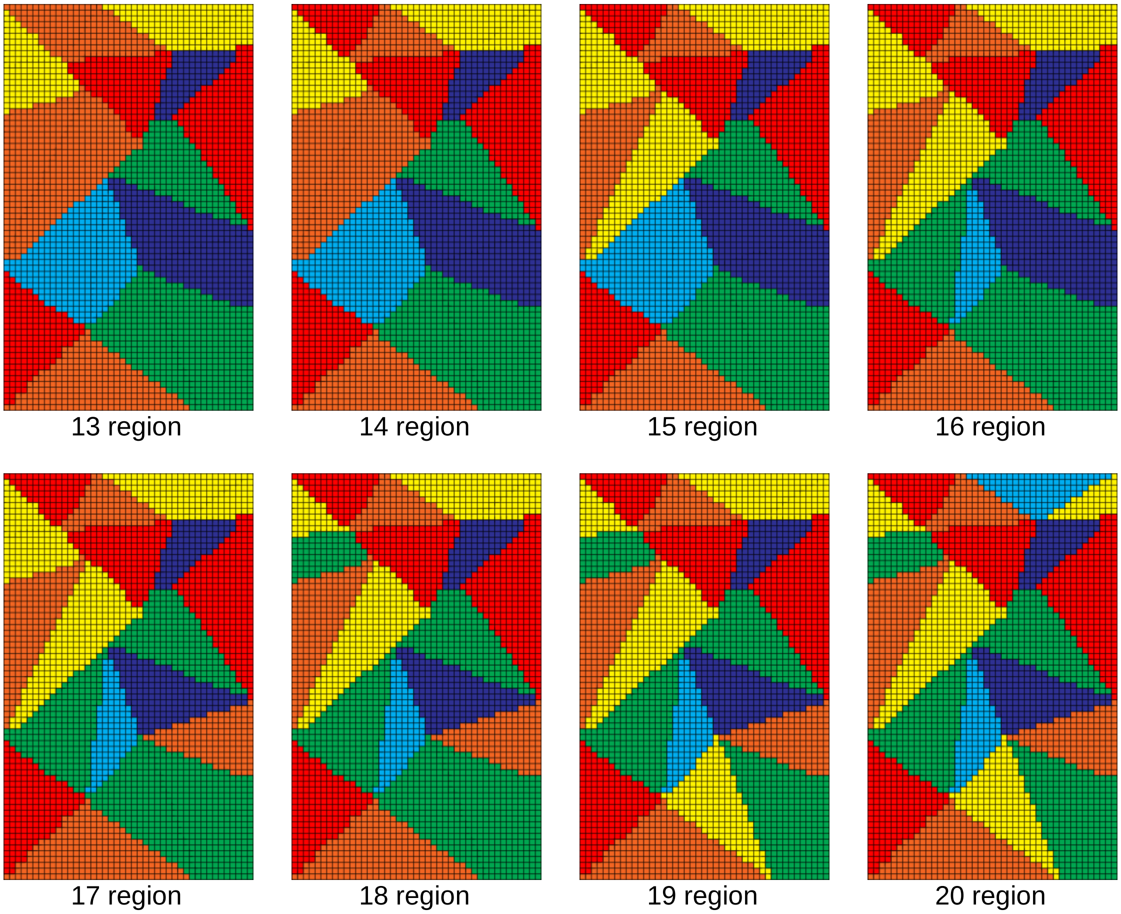
9 region

10 region

11 region

12 region

Figure **4-8a**: 70 x 43 rectangular reservoir divided into regions.



13 region

14 region

15 region

16 region

17 region

18 region

19 region

20 region

Figure **4-8b**: 70 x 43 rectangular reservoir divided into regions (cont'd).

4.2.3 Case 3: Multiphase Reservoir with Irregular Boundaries

The third case studied adds more complexity to the reservoir by using irregular boundaries for the reservoir. The irregular boundary used in this study is adapted from Ertekin et al., 2001. This reservoir is discretized into 59 x 28 grid blocks. Each grid block measures 100 ft x 100 ft. Similar to the square reservoir the number of wells and regions in this case is also varied from 1 through 20. Figure **4-9** shows a reservoir with 20 wells and one region. Figure **4-10a** and **4-10b** shows how the reservoir is divided into regions in this study.

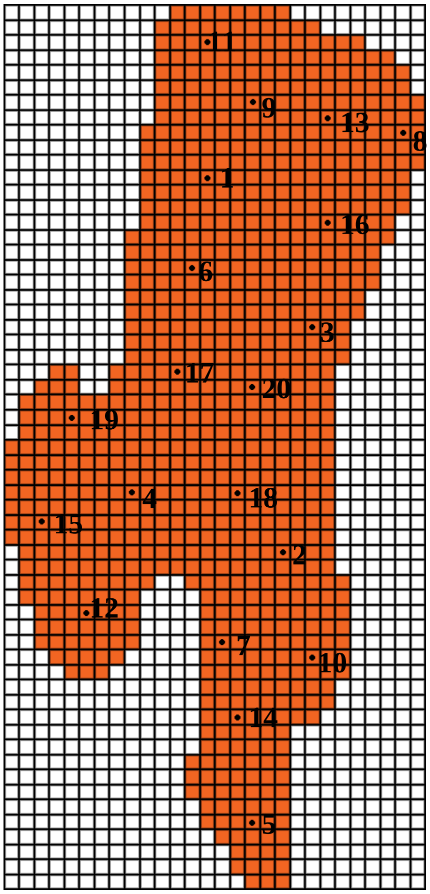
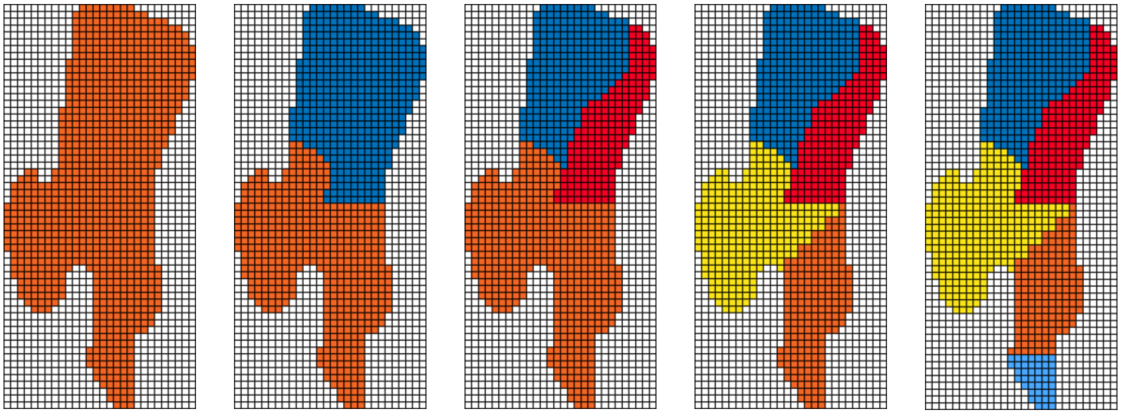


Figure **4-9**: 59 x 28 reservoir with irregular boundary one region and 20 wells.



1 region

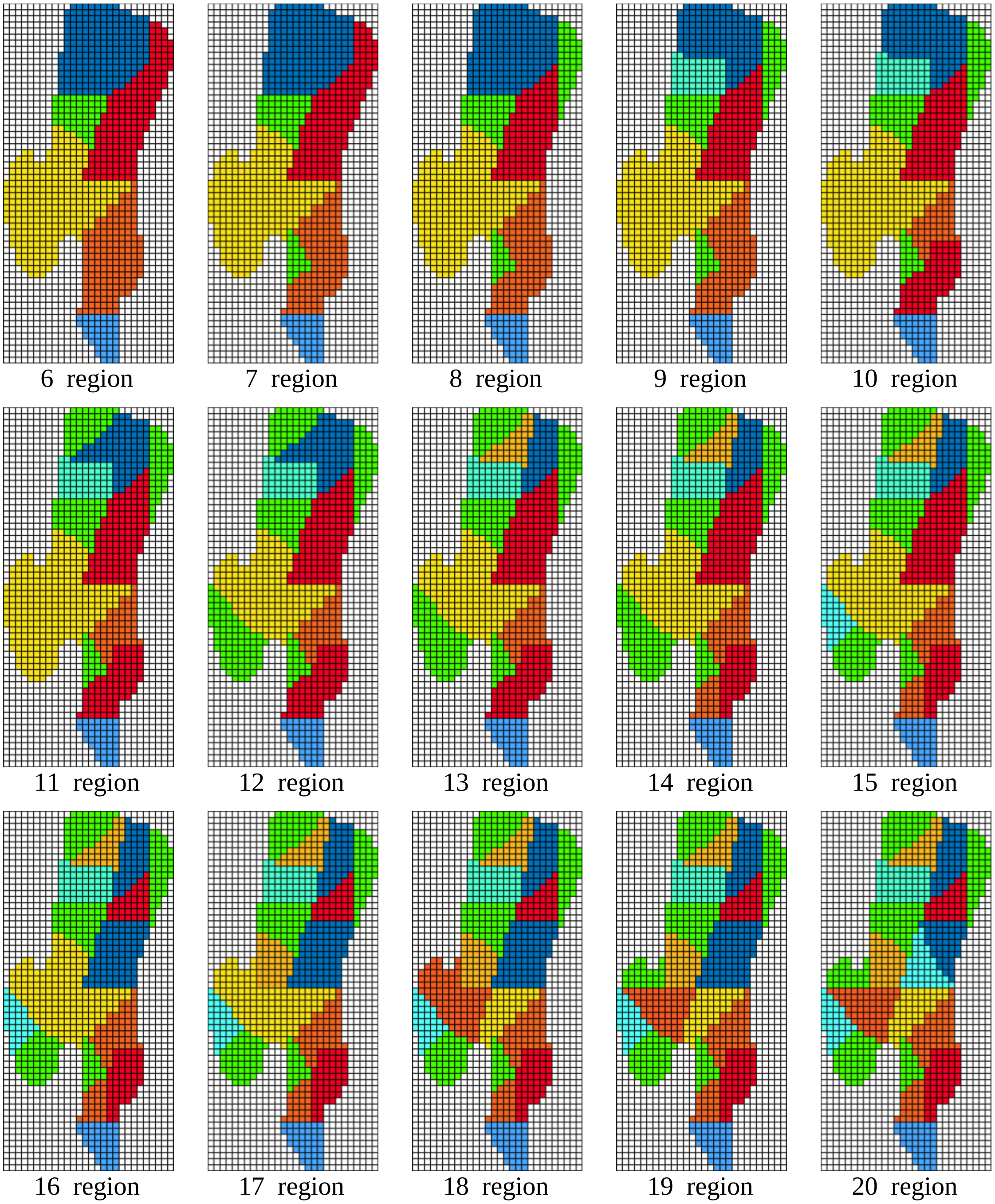
2 region

3 region

4 region

5 region

Figure **4-10a**: 59 x 28 reservoir with irregular boundary divided into regions.



6 region

7 region

8 region

9 region

10 region

11 region

12 region

13 region

14 region

15 region

16 region

17 region

18 region

19 region

20 region

Figure **4-10b**: 59 x 28 reservoir with uneven boundary divided into regions (cont’d).

4.3 Summary

The three cases that are discussed each have 400 different types of reservoirs with varying number of wells and regions/zones. And 200 different models with different porosity, permeability and net pay thickness are built for each type of reservoir.

Simulation runs are performed for all three cases discussed in Section **4.2**. The runs are used to generate 3600 days (~10 years) of production data. All the wells in the reservoir are put to production at the same time. The production and pressure results from the simulation runs are used to create the synthetic data that are necessary for developing the artificial neural network.

In the initial studies, porosity, permeability and net pay thickness are varied. Later relative permeability curves are also varied along with porosity, permeability and net pay thickness.

Chapter 5

DISCUSSION OF DEVELOPMENT OF ANN PREDICTION TOOL

This chapter describes the methodology employed in the design of the artificial neural network at each stage of its development. The network evolves all the way through the development process. Some of the challenges faced and solutions developed are also discussed. Few significant case studies are presented in increasing order of complexity. The development of network can be divided into two stages. First stages of development concentrate on capturing complexities introduced by varying reservoir geometries. Porosity, permeability and net pay thickness are predicted during this stage. During second stage of development other parameters like relative permeability curves are included to the list of parameters predicted. Development of network designer is also discussed. Based on observations made during the development process, guidelines for development of ANN prediction tool is summarized at the end.

5. 1 Factors Influencing the Design of Artificial Neural Networks

Design of ANN involves optimizing several parameters. The factors that are taken into consideration when determining the optimum architecture for ANN are the following.

- Data structure – The structure of data inputs to and outputs from the network
- Number of hidden layers – the number of middle layers in the network that would optimize the training/learning process
- Number of neurons in each layer – the number of neurons within the hidden layers of the network that would produce the desired characteristics of the output layer

- Transfer functions – a linear or non-linear function in each layer that is chosen to satisfy some specification of the problem that the neuron is attempting to solve
- Training algorithm – a training function that updates the weight and bias values in each layer of the network
- Performance functions – a function that measures the network's performance
- Functional links – mathematical functions of inputs or outputs which amplify subtle differences in the data

5.2 Data Formulation

Production and pressure data for various cases described in Section **4.2** are obtained from simulation runs. Those are formatted such that they can be effectively learnt by the network. Ramgulam (2006) has suggested ways to formulate the data for effective training of network for history match problems. Some of those suggestions are implemented in our study.

Input/target data to the network constitutes time, incremental cumulative production, and incremental pressure values (t, ΔNp, ΔWp, ΔGp, ΔP). Incremental cumulative production at time intervals of 30 days during the 3600 days (~10 yrs) period was used. In initial studies, porosity, permeability and thickness are used as output/target for the network. In later studies, parameters used to generate relative permeability curves (viz., end point saturation, relative permeability at end point saturations and exponent) are added to the output/target for the network. The input data are pre-processed by normalizing them between -1 and 1. Normalizing the data standardizes the numerical range of the input data and enhances the fairness of training by preventing an input with large values from swamping out another input that is equally important but with smaller values [Al-Fattah, 1994].

At later stages of the study it is necessary to add functional links that are mathematical functions of either input or output data. These amplify subtle differences in the data and provide more connections to the middle layers and correspondingly improve predictive capabilities. The amount of data in the input is also reduced by increasing data intervals to 90 days instead of 30 days. Production data are also limited to 1800 days (~5 years).

5.3 Case studies

Some of the significant case studies are discussed in detail. Stage I concentrates on capturing the complexities introduced by varying reservoir geometries. Porosity, permeability and net pay thickness are the parameters being predicted in this stage. Stage II concentrates on adding more parameters to the list of parameters predicted. Due to lack of well established guidelines design of neural network is by trial-and-error. Network performance is considered as acceptable if the prediction errors are under ±5%.

5.3.1 Stage-I

Study is initiated with a simplest case (square homogeneous reservoir with one region and one producing well). The study is then expanded to more complicated reservoir geometries. Figure **5-1** shows the progression of complexity of reservoir geometries during network development process.

5.3.1.1 Square boundary with one region and one producing well

Production data obtained from the simulation runs are formatted into input and output data for neural network. Input data are constructed with time, incremental cumulative production, and incremental pressure values (t, ΔNp, ΔWp, ΔGp, ΔP). Production data at 30 day intervals for the first 3600 days of production are used. Output data are constructed with porosity, permeability and thickness values. The 200 data sets are split into training and testing datasets with 50 and 150 data sets respectively.

The training data are used during the network training/learning process. During this time the neural network studies the data to understand the interrelations between the input and output data. The testing data consisting of 150 data sets are used to test the validity of the network. At this phase, only the production data are presented to the network and the network is utilized to predict the reservoir properties. Prediction errors are calculated to determine the quality of predictions. In our study, we accept the network performance as acceptable if the prediction errors are under ±5%. Once the network is trained and tested, they can be used to make predictions with novel data sets.

Initially, a network with 3 hidden layers is used. The hidden layers have 20, 30, 25 neurons in the 3 hidden layers respectively. Hyperbolic tangent sigmoid (*tansig*) transfer function is used in the hidden layers and linear (*purelin*) transfer function is used in the output layer. Levenberg-'Marquardt backpropagation' (*trainlm*), one of the fastest and hence most recommended training functions, along with 'Gradient descent with momentum' weight and bias learning function (*learngdm*) is used. In order to determine whether the data structure that is input to the network will have any effect on the network's performance and prediction results, two different data structures are examined.

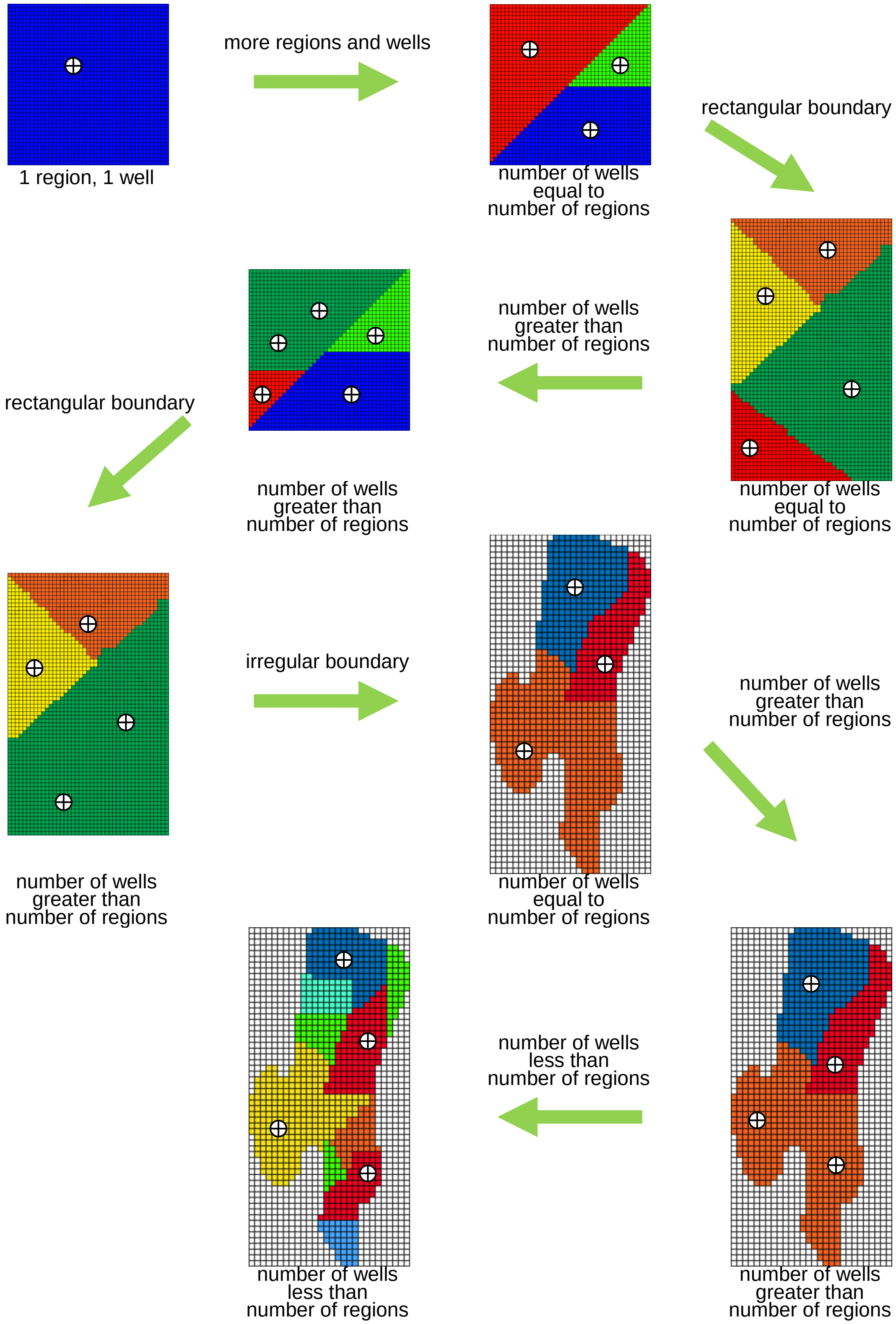
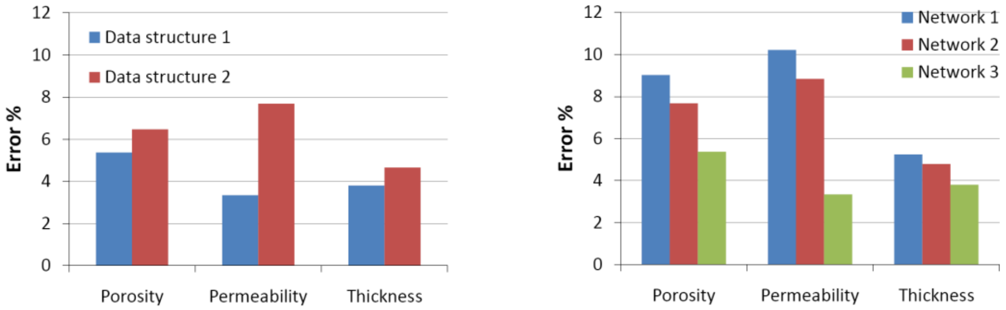


Figure **5-1**: Process of study with increasing complexity in reservoir geometries

The first data structure consists of all time steps for a specific case followed by the second case, third case etc., for all 50 cases. The second data structure consists of the first time step for each case followed by the second time step for each case, etc. Sample from both data structures are shown in Appendix **B**. It is determined that the first data structure gives lower average training errors for all properties as shown in Figure **5-2**.



a) Comprision of data structure

b) Comparison of Network Structures

Figure **5-2**: Average training errors for each property using different data structures and network architectures (square reservoir with 1 well and 1 region).

In order to study the effect of network architecture, several different architectures with differing number of neurons are tested. Three different network architectures of significance are presented. First network uses [20 20 25] neurons; second network uses [20 25 25] neurons and third network uses [20 30 25] neurons in the 3 hidden layers respectively. It is observed that third network gives lower training errors compared to others as shown in Figure **5-2**. Data structure 1 is used with all three architectures.

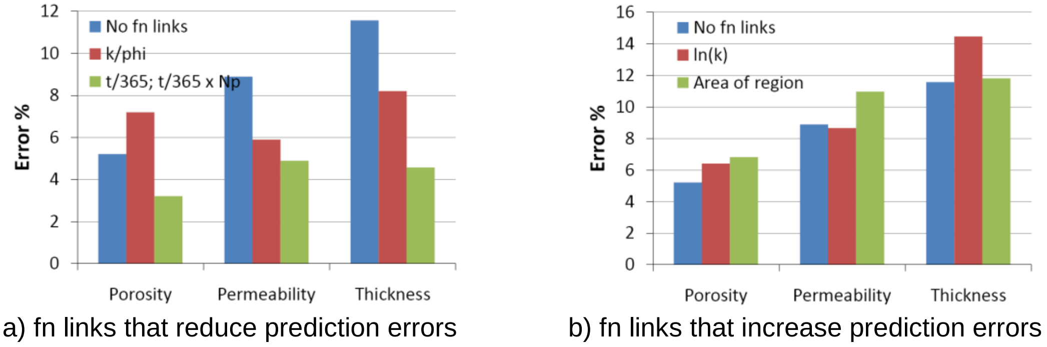
Although the training errors are low, when the network is used to predict the properties using testing data set, results are not satisfactory as shown in Figure **5-3**. The set criteria of less than ±5% error for the predicted values were not met. This led to exploration of functional links that might help improve network performance.

Several different functional links for both input and output layers are tested. Some of the significant functional links tested are listed in Table **5-1**. Testing results with those functional links are shown in Figure **5-3**.

Table **5-1**: Functional links examined (square reservoir with 1 well and 1 region).

Input Layer		Output Layer	
Functional link(s)	Prediction errors	Function link(s)	Prediction errors
t/365	↓	ln(k)	↑
t/365 x Np	↓	ln(ϕ)	↑
t/365 x Wp	↑	ln(h)	↑
t/365 x Gp	↑	k/ϕ	↓
Area of region	↑	k x h	↓
Distance of well from boundaries	↑	1/(ϕ x k)	↑
t/365; t/365 x Np	↓	ϕ x h	↑

Thus, use of functional links t/365 and t/365 x Np together in the input layer proves to be the most beneficial. Another point to be noticed is that while use of functional links can improve the network performance, it can also cause deterioration of network performance. Other parameter that is optimized is convergence criterion. After testing several different convergence criterions, it is observed that a convergence criterion of 5e-4 yielded best results. Amount of input data were also reduced to 1800 days (~5 years) production data at 90 day intervals in an attempt to speedup network training. This change reduced the training time without affecting the quality of predictions. Network architecture used is shown in Figure **5-4.**



a) fn links that reduce prediction errors

b) fn links that increase prediction errors

Figure **5-3**: Average prediction errors for each property using different functional links (square reservoir with 1 well and 1 region).

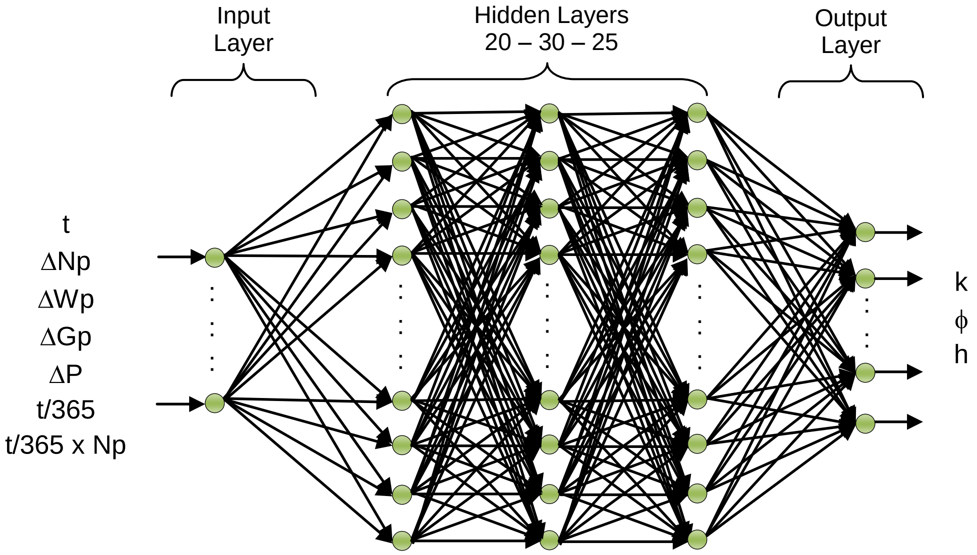


Figure **5-4**: Structure of feedforward backpropagation network for square reservoir with 1 well and 1 region.

5.3.1.2 Square boundary with number of wells equal to number of regions

After obtaining good predictions in the previous case, study is extended to reservoirs with more wells and regions. At this point, study is limited to reservoirs with number of wells equal to number of regions. All regions in the reservoir have one producing well. As the number of regions increase, the number of input and output variables in the neural network also increases. This also means that the number of neurons required in the hidden layers of the neural network to capture the interrelations between input and output also increases. So, at every stage the number of neurons required has to be determined by trial-and-error method.

With the current network architecture, it is observed that the prediction errors increase as the number of regions is increased. It is clear from Figure **5-5** that the current network architecture is not powerful enough to capture the interrelationship between input and output data for systems with more regions.

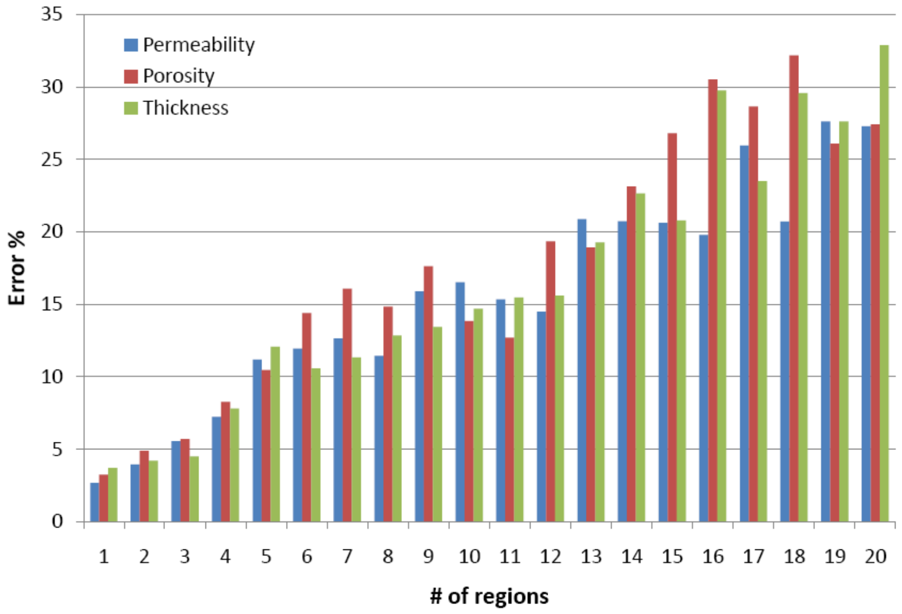
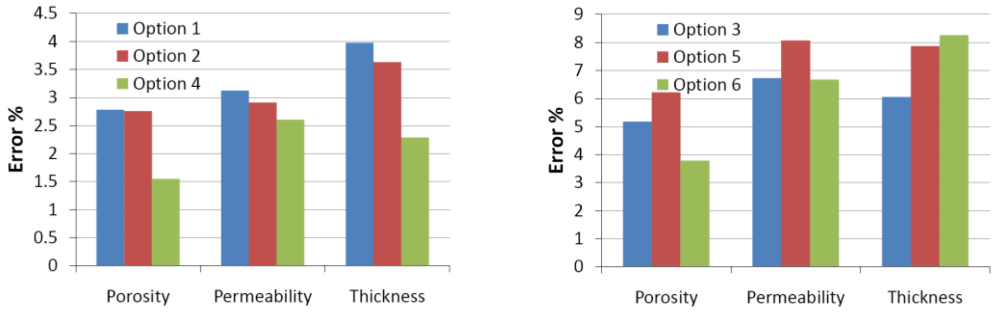


Figure **5-5**: Average prediction errors for each property using network architectures developed in Section **5.3.1.1** (number of wells equal to number of regions).

Thus at this point, possibilities of improving the network architecture are explored. As a first step, the list of functional links that helped reducing the prediction errors in previous section is revisited. Reservoir with three regions and wells are used to experiment with more functional links and combination of several functional links in the input and output layers. Several different functional links for both input and output layers are tested. Some of the significant functional links tested are listed in Table **5-2**. Testing results with some of the functional links are shown in Figure **5-6**.

Table **5-2**: Functional links examined (square reservoir with number of wells equal to number of regions).

Option	Functional link(s)	Prediction errors
1	t/365, t/365 x Np, k/ϕ	↓
2	t/365, t/365 x Np, k x h	↓
3	t/365, t/365 x Np, 1/(k x ϕ)	↑
4	t/365, t/365 x Np, k/ϕ, k x h	↓
5	t/365, t/365 x Np, t/365 x Wp	↑
6	t/365, t/365 x Np, t/365 x Gp	↑



a) fn links that reduce prediction errors b) fn links that increase prediction errors

Figure **5-6**: Average prediction errors for each property using different functional links (square reservoir with 3 wells and 3 regions).

From Figure **5-6** it is clear that use of functional links t/365 and t/365 x Np, in the input layer along with k/ϕ and k x h in output layer is the best choice of functional links for reservoir with 3 wells and 3 regions. This combination of functional links is used to train and test neural networks for reservoirs which have more regions. It is observed that this new network architecture is capable of predicting history match parameters for square reservoirs where number of wells is equal to number of regions as shown in Figure **5-7**.

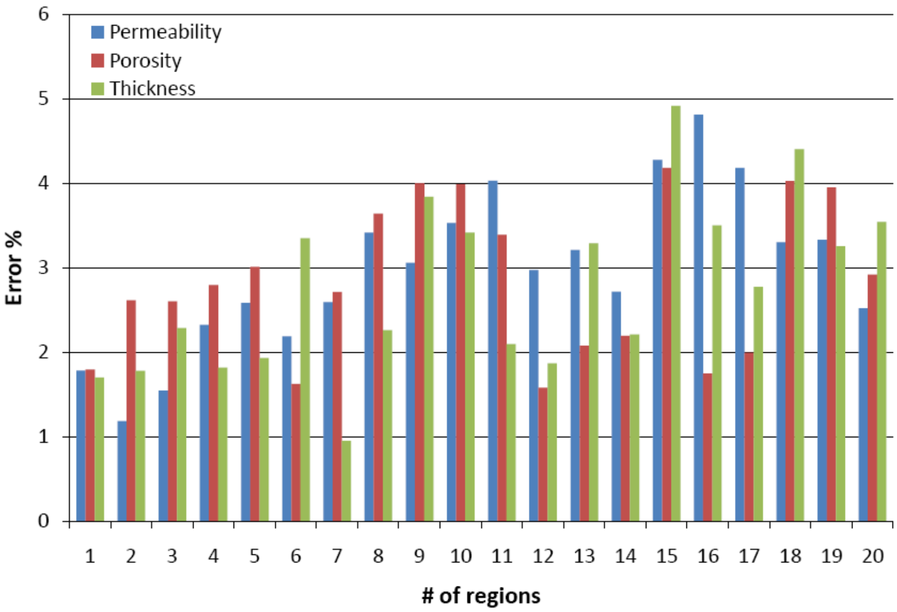


Figure **5-7**: Average prediction errors for each property using network architectures developed in Section **5.3.1.2** (square boundary with number of wells equal to number of regions).

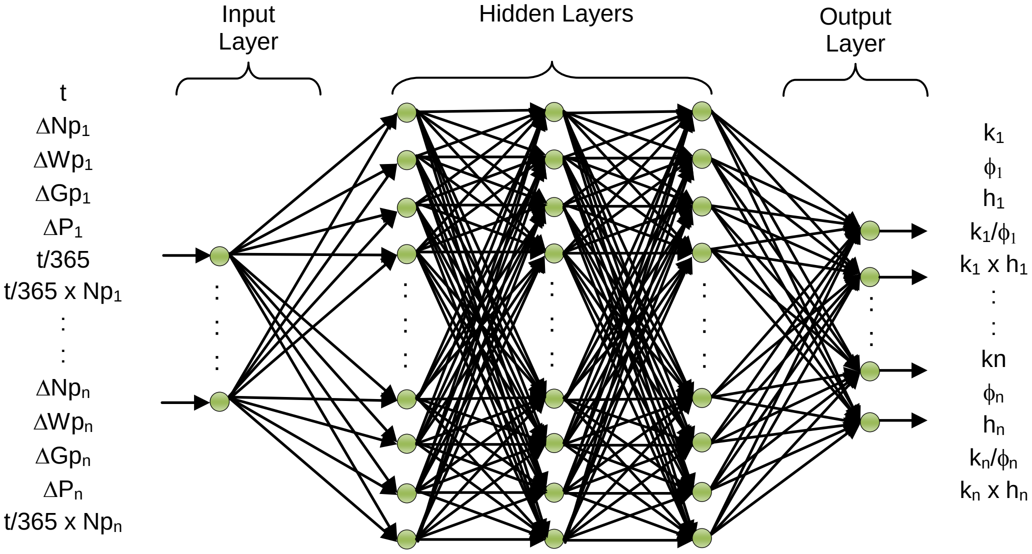


Figure **5-8**: Generalized structure of feedforward backpropagation network for square reservoir with n wells and n regions.

Figure **5-8** shows the generalized architecture of neural network used to train square reservoirs with equal number of regions and wells. This network uses, hyperbolic tangent sigmoid (*tansig*) transfer function in the hidden layers and linear (*purelin*) transfer function in the output layer. Levenberg-'Marquardt backpropagation' (*trainlm*) training function along with 'Gradient descent with momentum' weight and bias learning function (*learngdm*) is used. The number of neurons in the hidden layers varies according to the number of regions present in the reservoir. Number of neurons in the input layer is equal to 5*n+2 and number of neurons in output layer is equal to 5*n, where 'n' is the number of regions/wells.

5.3.1.3 Rectangular boundary with number of wells equal to number of regions

After obtaining satisfactory performance of network for square reservoirs in the previous case, complexity of the reservoir geometry was increased by changing the reservoir boundary from square to a rectangle. Reservoirs with rectangular boundaries that are discretized into 70x43 grid blocks (discussed in section **4.2.2**) are studied. All regions in the reservoir have one producing well. Network architecture developed in the section **5.3.1.2** is used. The number of neurons in each hidden layer is identical to the previous case. This neural network was trained and tested for reservoirs with varying number of regions. Network prediction errors are within acceptable range as shown in Figure **5-9**. Thus, it is concluded that the network architecture developed in Section **5.3.1.2** is generalized enough for use with both square and rectangular boundary reservoirs with equal number of wells and regions.

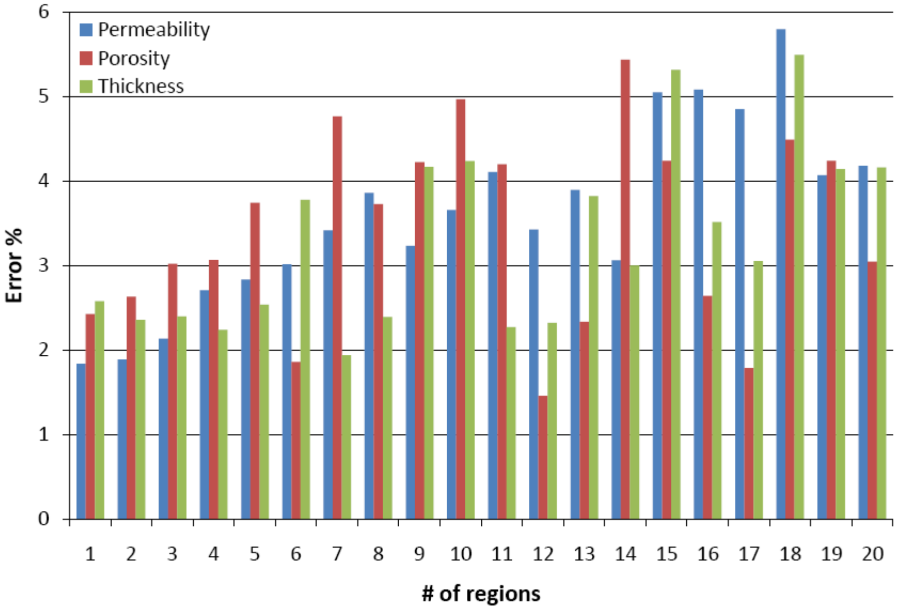


Figure **5-9**: Average prediction errors for each property using network architectures developed in Section **5.3.1.2** (rectangular boundary with number of wells equal to number of regions).

5.3.1.4 Square boundary with number of wells greater than number of regions

Network architecture developed in Section **5.3.1.2** has worked well when the number of wells is equal to number of regions. So, study is extended to reservoirs with number of wells greater than number of regions. All regions in the reservoir have at least one producing well. It has to be noted that as the number of wells increase the number of input variables also increase. It is logical to assume that with more input variables and same number of output variables, the predictions made by the network should be good. Study is pursued with this hope. The number of neurons required for each system varies according to the number of wells and regions present in the system being studied. As the number of wells and regions increase, the number of input and output variables also increases respectively. Thus, the number of neurons required in the hidden layers also increases. So, at every stage the number of neurons required has to be determined by trial-and-error method. The network architecture was able to predict properties within reasonable error margins. Prediction errors of a sample of the reservoirs tested is shown in Figure **5-10**.

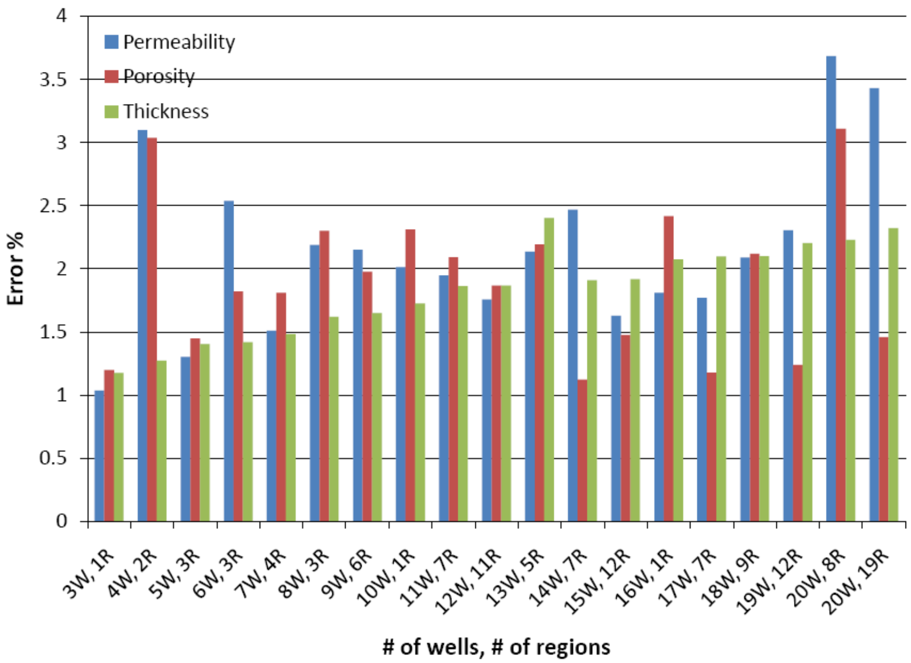


Figure **5-10:** Average prediction errors for each property using network architectures developed in Section **5.3.1.2** (square boundary with number of wells greater than number of regions).

Figure **5-11** shows the generalized form of network architecture for reservoirs with square boundary where number of wells is greater than number of regions. This network also uses, hyperbolic tangent sigmoid (*tansig*) transfer function in the hidden layers and linear (*purelin*) transfer function in the output layer along with *trainlm* training function and *learngdm* weight and bias learning function. The number of neurons in the hidden layers varies according to the number of regions and number of wells present in the reservoir. Number of neurons in the input layer is equal to 5*n+2 and number of neurons in output layer is equal to 5*m, where ‘n’ is the number of wells and ‘m’ is the number of regions in the reservoir.

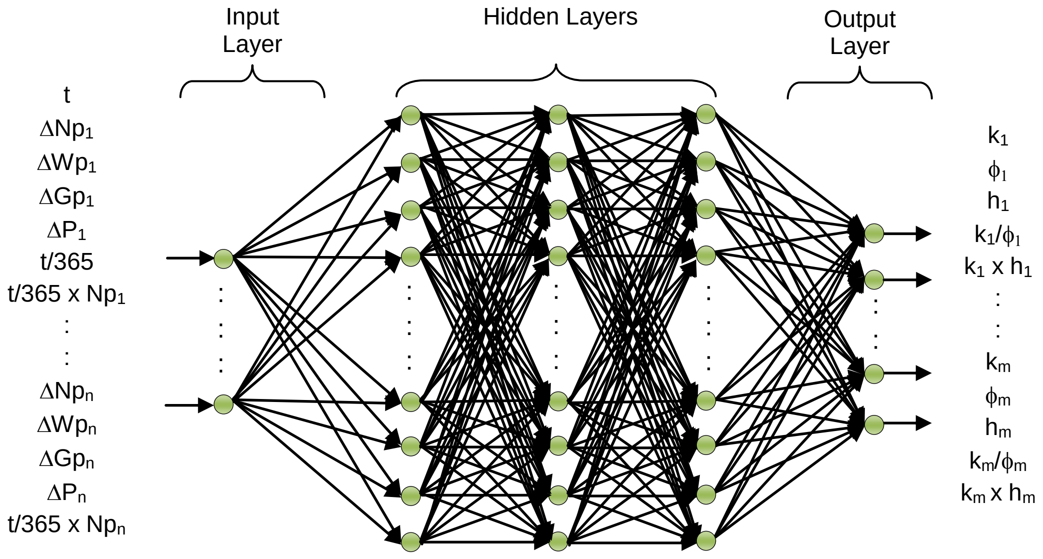


Figure **5-11**: Generalized structure of feedforward backpropagation network for square reservoir with n wells and m regions; n > m (number of wells greater than number of regions).

5.3.1.5 Rectangular boundary with number of wells greater than number of regions

Reservoir geometry is modified from square to a rectangular boundary as complexity of the reservoir being studied is increased. All regions in the reservoir have at least one producing well. This system is trained and tested using the network architecture developed in Section **5.3.1.2**. The number of neurons in each hidden layer is identical to the previous case. The network is able to predict properties within reasonable error margins. Prediction errors of a sample of the reservoirs tested is shown in Figure **5-12**.

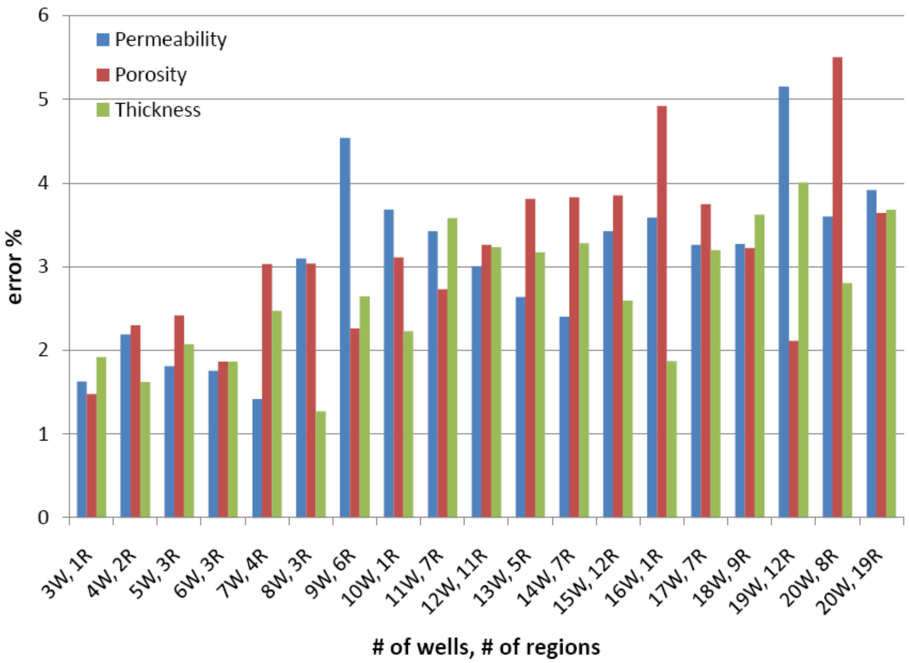


Figure **5-12**: Average prediction errors for each property using network architectures developed in Section **5.3.1.2** (rectangular boundary with number of wells greater than number of regions).

5.3.1.6 Irregular boundary with number of wells equal to number of regions

Network developed in Section **5.3.1.2** has been able to predict properties in reservoirs with square boundaries as well as rectangular boundaries. This network is next exposed to reservoir with irregular boundary and equal number of wells and regions. All regions in the reservoir have one producing well. It is observed that the network is not able to predict properties within reasonable margin of error as shown in Figure **5-13**. Network is not able to handle the complexity brought by the reservoir with irregular boundary. At this point it has become inevitable to explore options to modify the base architecture of the network again. Several functional links that could provide clues about the shape of the reservoir are examined. Some of the significant functional links tested are listed in Table **5-3**. Testing results with some of the functional links are shown in Figure **5-14**.

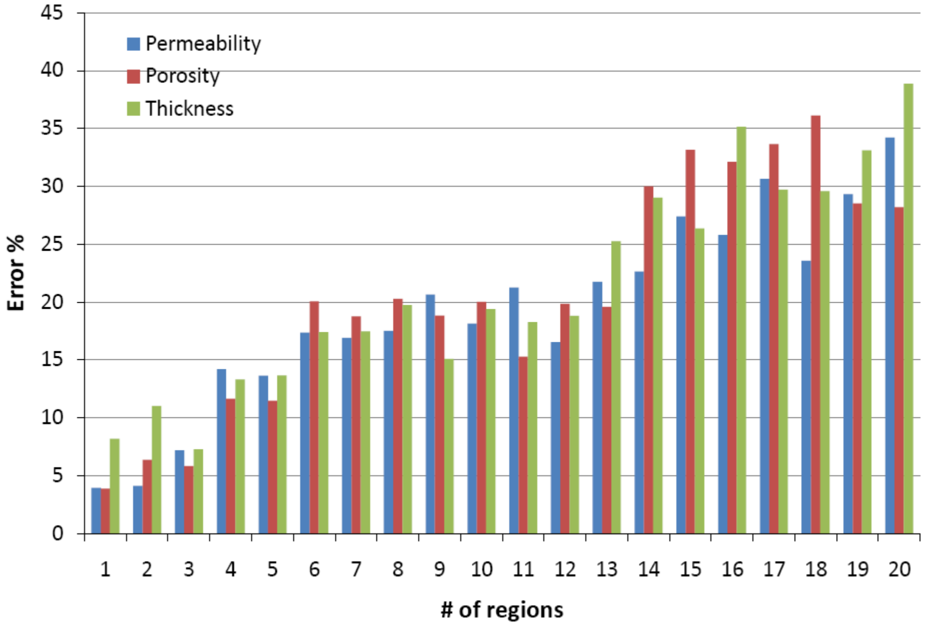
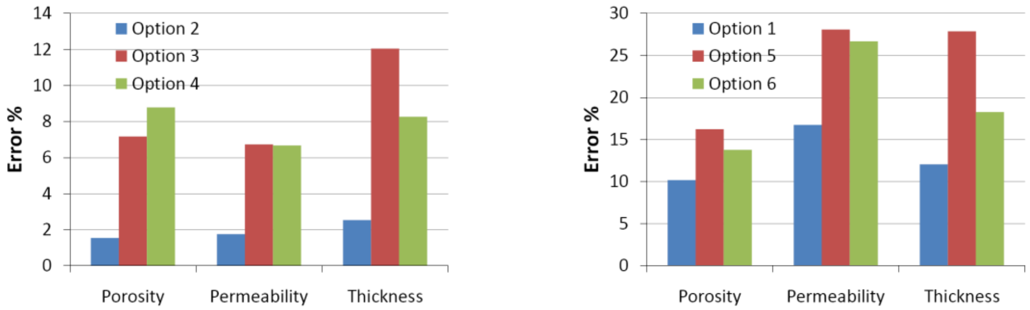


Figure **5-13**: Average prediction errors for each property using network architectures developed in Section **5.3.1.2** (irregular boundary with number of wells equal to number of regions).

Table **5-3**: Functional links examined (irregular boundary reservoir with number of wells equal to number of regions).

Option	Functional link(s)	Prediction errors
1	t/365, t/365 x Np, k/ϕ, k x h, Area of region	↑
2	t/365, t/365 x Np, k/ϕ, k x h, Area of region / k	↓
3	t/365, t/365 x Np, k/ϕ, k x h, Area of region / h	↓
4	t/365, t/365 x Np, k/ϕ, k x h, Area of region / ϕ	↓
5	t/365, t/365 x Np, k/ϕ, k x h, Distance of well to nearest boundary	↑
6	t/365, t/365 x Np, k/ϕ, k x h, Distance of well to farthest boundary	↑



a) fn links that reduce prediction errors

b) fn links that increase prediction errors

Figure **5-14**: Average prediction errors for each property using different functional links (irregular boundary reservoir with 3 wells and 3 regions).

From Figure **5-14** it is clear that use of functional links t/365, t/365 x Np, in the input layer along with k/ϕ, k x h, A/k in output layer is the best choice of functional links for reservoir with 3 wells and 3 regions. These functional links are used to training and testing other reservoirs with equal number of wells and regions. It was observed that this new network architecture works fine for irregular reservoirs with equal number of wells and regions as shown in Figure **5-15**.

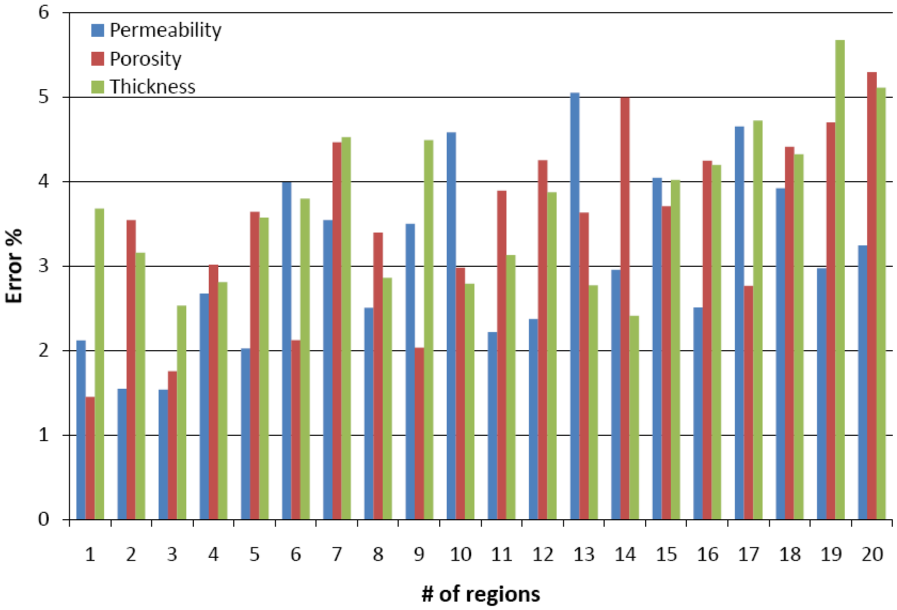


Figure **5-15**: Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary with number of wells equal to number of regions).

As the number of regions increase, the number of input and output variables also increase. Thus, the number of neurons required in the hidden layers also increase. So, at every stage the number of neurons required in the hidden layers has to be determined by trial-and-error method. Number of neurons in the input layer is 5*n+2 and number of neurons in the output layer is 6*n, where ‘n’ is the number of regions. Figure **5-16** shows the generalized form of network architecture for reservoirs with irregular boundary with equal number of wells and regions.

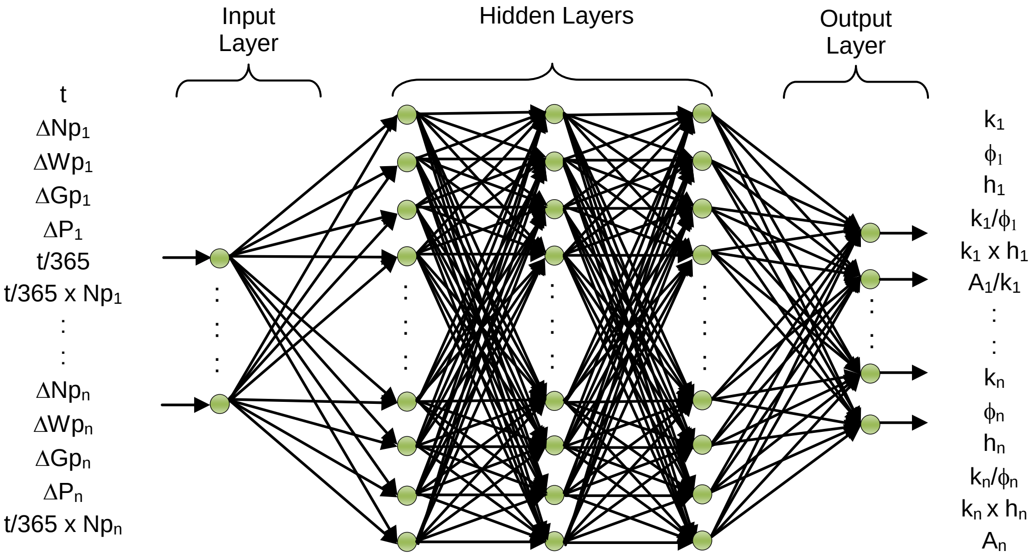


Figure **5-16**: Generalized structure of feedforward backpropagation network for square reservoir with n wells and n regions (number of wells equal to number of regions).

5.3.1.7 Irregular boundary with number of wells greater than number of regions

The new network architecture developed in Section **5.3.1.6** is tested on reservoirs with number of wells greater than number of regions. All regions in the reservoir have at least one producing well. It has to be noted that as the number of wells increase the number of input variables also increase. It is logical to assume that with more input variables and same number of outputs, the predictions made by the network should be good. Study is pursued with this hope and the logic proves right again. The neural network developed in section **5.3.1.6** is able to predict properties within reasonable error margins. Prediction errors of a sample of the reservoirs tested is shown in Figure **5-17**. The number of neurons required for each system varies according to the number of wells and regions present in the system being studied. As the number of wells and regions increase, the number of input and output variables also increases respectively. Thus, the number of neurons required in the hidden layers also increases. So, at every stage the number of neurons required has to be determined by trial-and-error method. The neural network developed in section **5.3.1.6** is able to predict properties within reasonable error margins. Prediction errors of a sample of the reservoirs tested is shown in Figure **5-17**. The number of neurons in the hidden layers varies according to the number of regions and number of wells present in the reservoir. Number of neurons in the input layer is equal to 5*n+2 and number of neurons in output layer is equal to 6*m, where 'n' is the number of wells and 'm' is the number of regions in the reservoir.

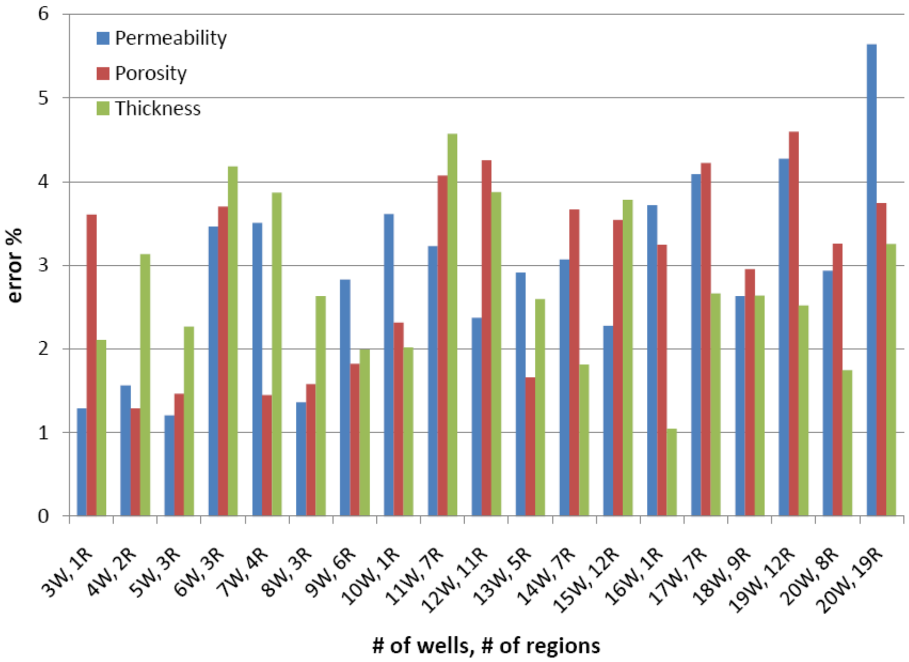


Figure **5-17:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary with number of wells greater than number of regions).

Figure **5-18** shows the generalized form of network architecture for reservoirs with irregular boundary where number of wells greater than number of regions.

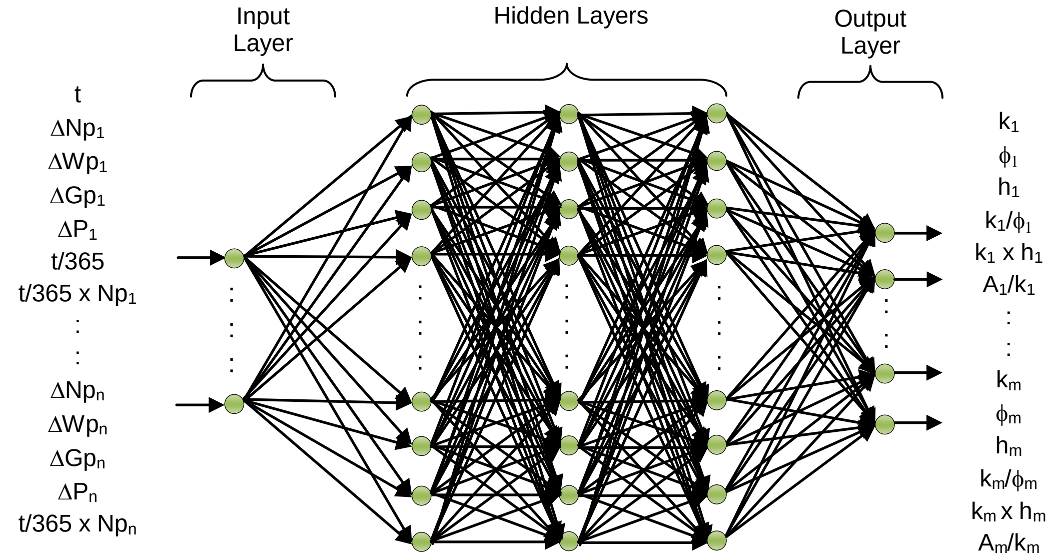


Figure **5-18**: Generalized structure of feedforward backpropagation network for reservoir with irregular boundary with n wells and m regions; n > m (number of wells greater than number of regions).

5.3.1.8 Irregular boundary with number of wells less than number of regions

This is the most complex reservoir geometry that is studied. None of the regions in the reservoir have more than one producing well. It should be noted that when the number of wells is decreased the number of input variables used by network also decreases. But, the number of output variables which depends on number of regions does not decrease. This is not a good scenario to be in because now the network has to learn to predict more properties with less production information.

Network developed in section **5.3.1.6** is used to train and test the reservoirs. As the number of regions increase, the number of input and output variables also increase. Thus, the number of neurons required in the hidden layers also increase. So, at every stage the number of neurons required has to be determined by trial-and-error method. The number of neurons in the hidden layers varies according to the number of regions and number of wells present in the reservoir. Number of neurons in the input layer is equal to 5*n+2 and number of neurons in output layer is equal to 6*m, where ‘n’ is the number of wells and ‘m’ is the number of regions in the reservoir.

In reservoirs where number of wells is less than number of regions, it is observed that as the difference between number of wells and number of regions increase, prediction errors also increase. This is a reflection of the fact that it is a harder task to predict properties with fewer production data. Figures **5-19a** and **5-19b** show the prediction errors for a sample of the reservoirs tested.

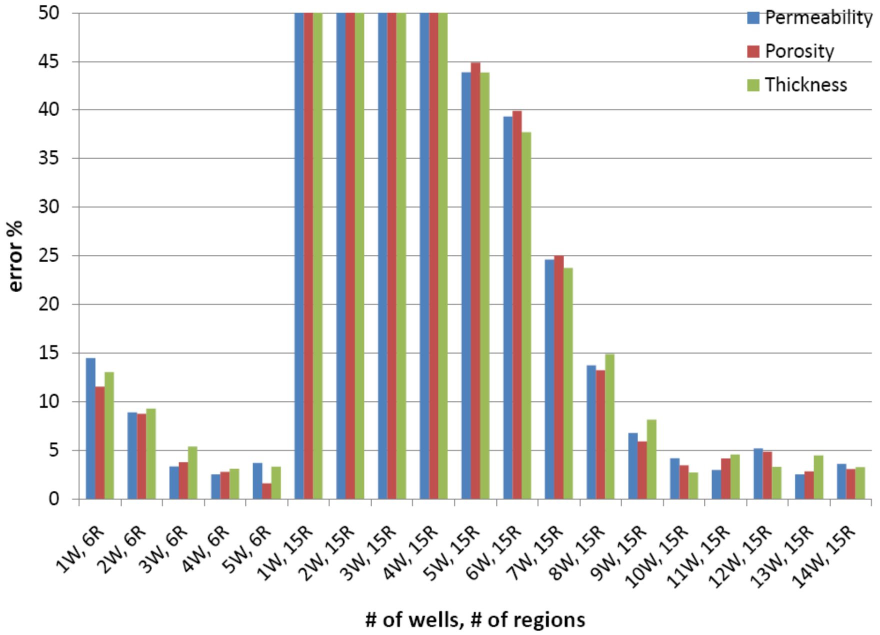


Figure **5-19a:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary with number of wells less than number of regions).

From figures **5-19a** and **5-19b** it is clear that as the number of wells go below the number of regions, the quality of predictions deteriorate. Prediction quality deteriorates rather fast as the difference between number of wells and number of regions increases. The current situation can be compared to solving a system of equation. To solve a system of equations with 'n' unknowns, 'n' simultaneous equations are required. In this case, this balance is disturbed and the number of unknowns is more than the number of relationships neural networks could make between the input and output parameters. When this happens the performance of the network gets attenuated. So, as the difference between the number of wells and regions increase the number of relationships neural network makes decrease.

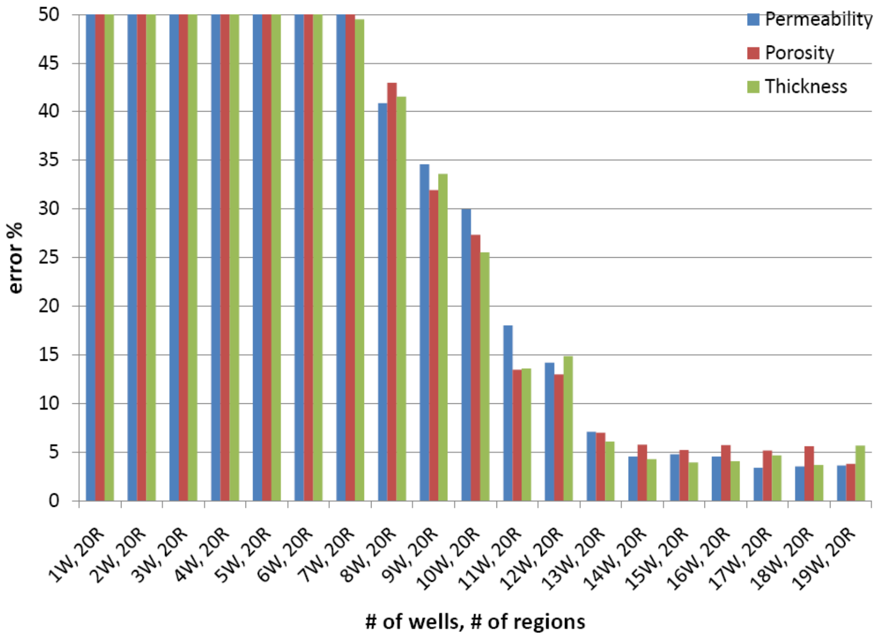


Figure **5-19b:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary with number of wells less than number of regions) (cont'd).

5.3.1.9 Summary

The neural network developed in section **5.3.1.6** has an architecture that can handle almost all the cases studied. This network was used to predict properties for reservoir with square and rectangular boundaries. The number of neurons used in each of the hidden layers is identical to those of reservoir with irregular reservoirs. The prediction errors for all the cases studied while using the network developed in section **5.3.1.6** are included in Appendix C. Figure **5-20** shows the generalized network architecture. This feedforward backpropagation network can be used to predict properties for all types of reservoirs except systems with number of wells is less than number of regions and the difference is large. The number of neurons in the input layer is 5*n+2 and number of neurons in output layer is 6*m, where ‘n’ is the number of wells and ‘m’ is the number of regions in the reservoir.

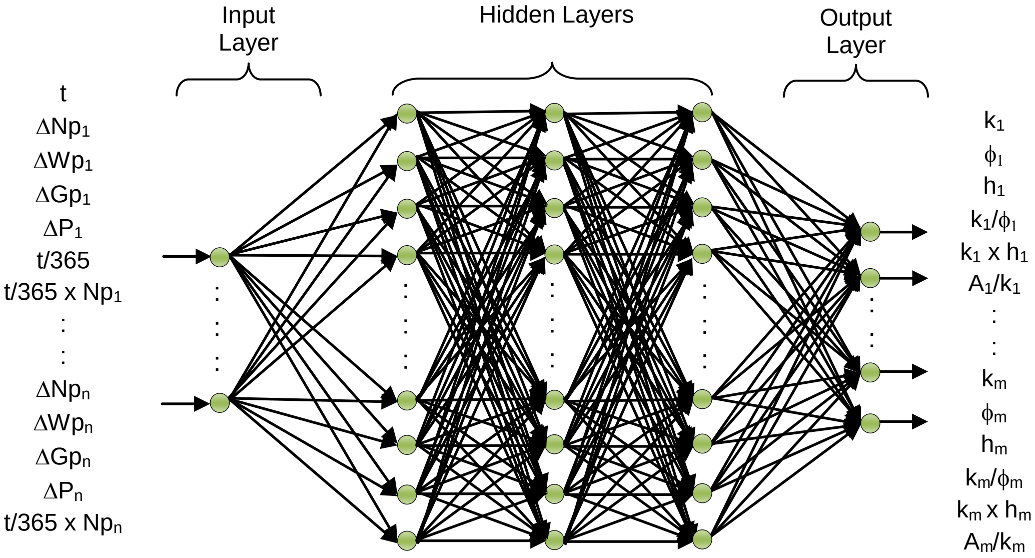


Figure **5-20**: Generalized structure of feedforward backpropagation network for reservoir with irregular boundary with n wells and m regions.

5.3.2 Stage-II

In Stage-II, relative permeability curves are varied along with porosity, permeability and net pay thickness. Reservoir with irregular boundary with n wells and m regions are studied in this stage of the study. Network developed in Stage-I is modified to include reservoirs with different relative permeability curves. To construct different sets of two-phase relative permeability curves, end point saturations (S_{orw}, S_{org}, S_{wirr}, S_{gcrit}), relative permeability at endpoint saturation (k_{rwro}, k_{roirw}, k_{rgro}, $k_{rocritg}$) and exponent values (N) are varied.

5.3.2.1 Irregular boundary with 'n' wells and 'm' regions using different relative permeability curves

Network developed in Section **5.3.1.6** has been able to predict porosity, permeability and net pay thickness in reservoirs with irregular boundary with n wells and m regions. This network is next exposed to a system that has to predict endpoint saturations, relative permeability at endpoint saturation and exponent along with porosity, permeability and net pay thickness. In preliminary tests it is observed that the network is not able to predict properties within reasonable margin of error as shown in Figure **5-21**.

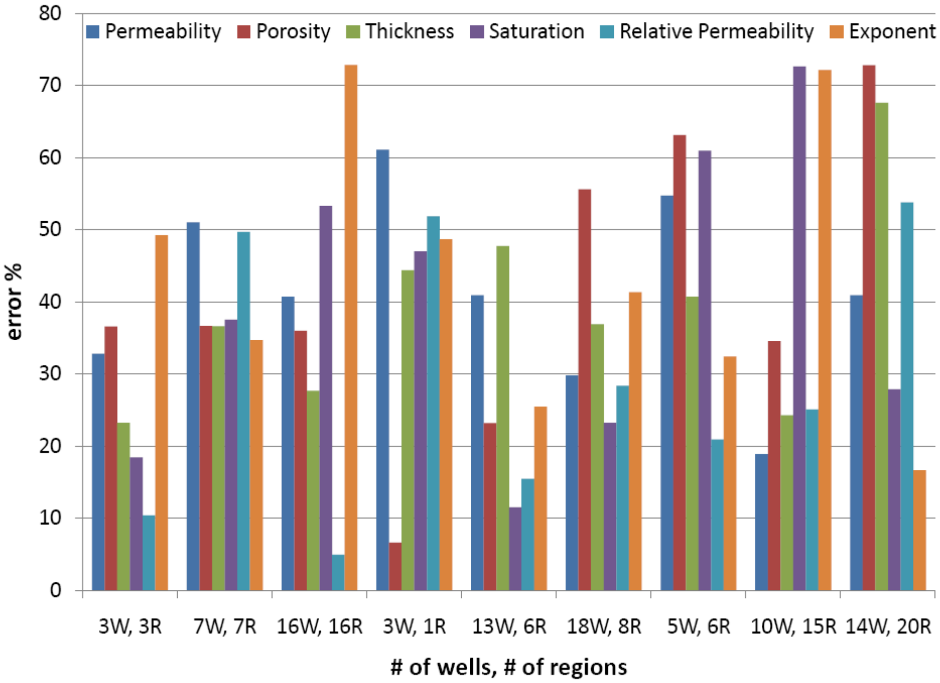


Figure **5-21**: Average prediction errors obtained during preliminary test of reservoirs with different relative permeability curves using network architectures developed in Section **5.3.1.6**

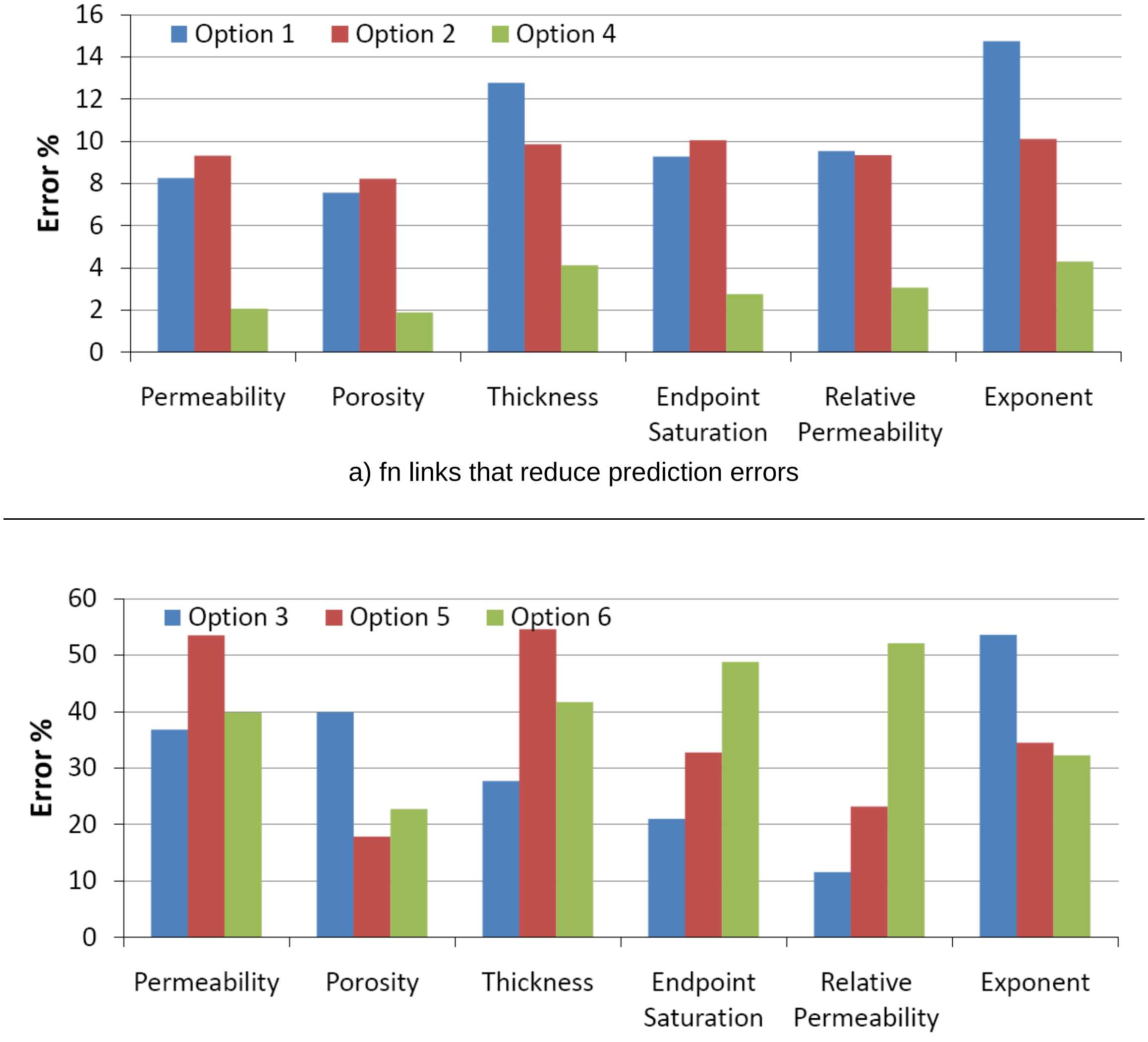
Network is not able to handle the complexity brought by reservoirs with different relative permeability curves. At this point it has become inevitable to explore options to modify the base architecture of the network again. Several functional links are examined. Some of the significant functional links tested are listed in Table **5-4**. Testing results with some of the functional links are shown in Figure **5-22**.

Table **5-4**: Functional links examined (irregular boundary reservoirs with different sets of relative permeability curves).

Option	Functional link(s)	Prediction errors
1	t/365, t/365 x Np, WOR, k/ϕ, k x h, Area of region / k	↓
2	t/365, t/365 x Np, GOR, k/ϕ, k x h, Area of region / k	↓
3	t/365, t/365 x Np, GLR, k/ϕ, k x h, Area of region / k	↑
4	t/365, t/365 x Np, WOR, GOR, k/ϕ, k x h, Area of region / k	↓
5	t/365, t/365 x Np, WOR, GLR, k/ϕ, k x h, Area of region / k	↑
6	t/365, t/365 x Np, GOR, GLR, k/ϕ, k x h, Area of region / k	↑

From Figure **5-22** it is clear that use of functional links t/365, t/365 x Np, WOR, GOR, in the input layer along with k/ϕ, k x h, A/k in output layer is the best choice of functional links for reservoir with 3 wells and 3 regions. These functional links are used to training and testing other reservoirs with irregular boundaries. It was observed that this new network architecture works fine for irregular reservoirs with equal number of wells and regions as shown in Figure **5-23**.

As the number of regions increase, the number of input and output variables also increase. Thus, the number of neurons required in the hidden layers also increase. So, at every stage the number of neurons required in the hidden layers has to be determined by trial-and-error method. Number of neurons in the input layer is 7*n+2 and number of neurons in the output layer is 6*m+9, where ‘n’ is the number of wells and m is the number of regions. Figure **5-24** shows the generalized form of network architecture for reservoir with irregular boundary with n wells and m regions. Matlab code used to train and test the network are shown in Appendix D.



a) fn links that reduce prediction errors

b) fn links that increase prediction errors

Figure **5-22**: Average prediction errors for each property using different functional links (irregular boundary reservoir with 3 wells and 3 regions).

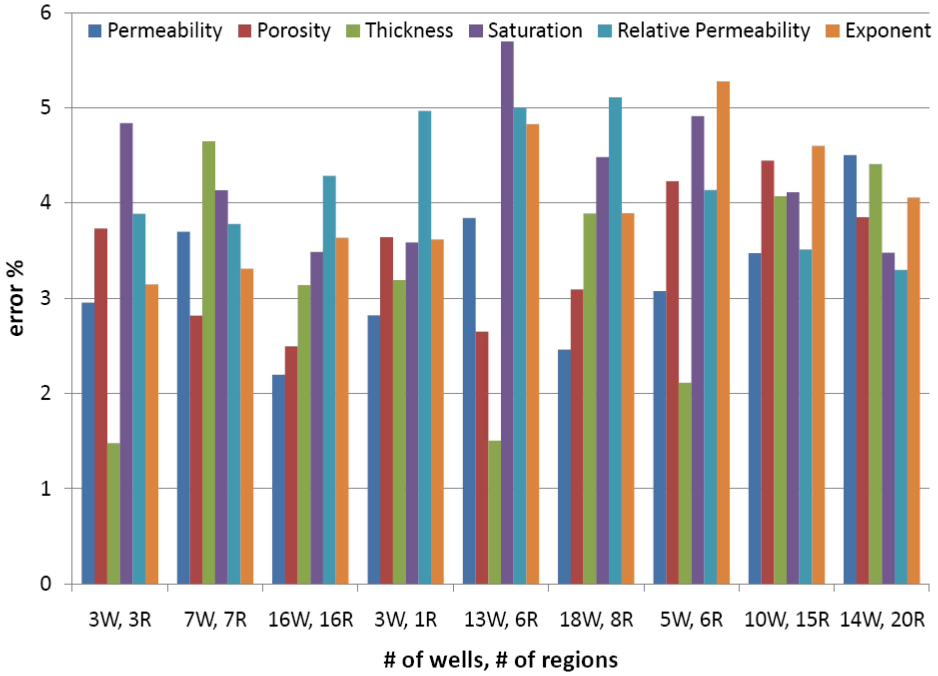


Figure **5-23**: Average prediction errors for each property using network architectures developed in Section **5.3.2.1** (irregular boundary with n wells and m regions).

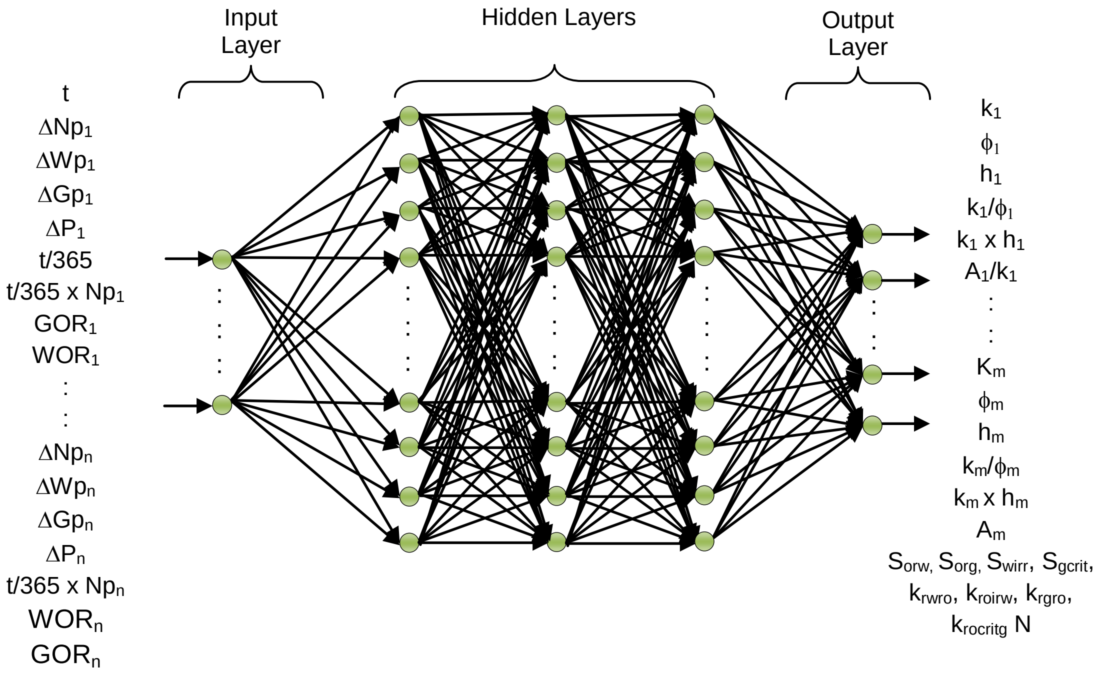


Figure **5-24**: Generalized structure of feedforward backpropagation network for reservoir with irregular boundary with n wells and m regions.

5.4 Network Designer

A network that performs well in all the scenarios studied has been developed. At this stage there is only one issue with the network. The number of neurons required in each hidden layer varies with number of wells and number of regions. So, users will be forced to refer to a manual to figure out the number of neurons required. In order to streamline the process, a network designer is developed. Network designer is a simple neural network that takes number of wells and number of regions in the reservoir as input and gives the number of neurons required in each hidden layer as output. Figure **5-25** shows the network architecture used to predict number of neurons required.

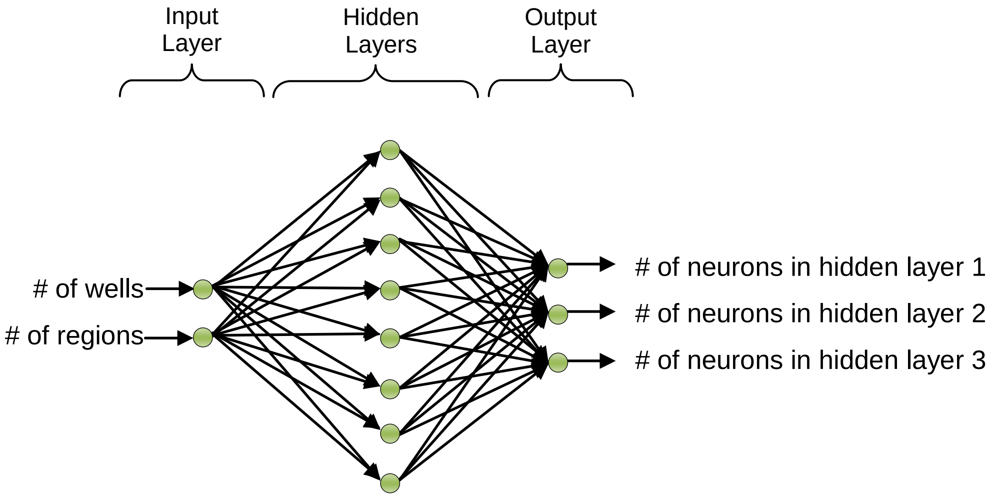

Input Layer
Hidden Layers
Output Layer
# of wells
# of regions
# of neurons in hidden layer 1
# of neurons in hidden layer 2
# of neurons in hidden layer 3

Figure **5-25**: Network Architecture of Network Designer.

Network designer uses data observed during the properties prediction network. It has 1 hidden layer with 8 neurons. Transfer functions used are hyperbolic tangent sigmoid (*tansig*) and linear (*purelin*) functions. Network designer will act as an easy to use tool to calculate number of neurons required for the reservoir being studied.

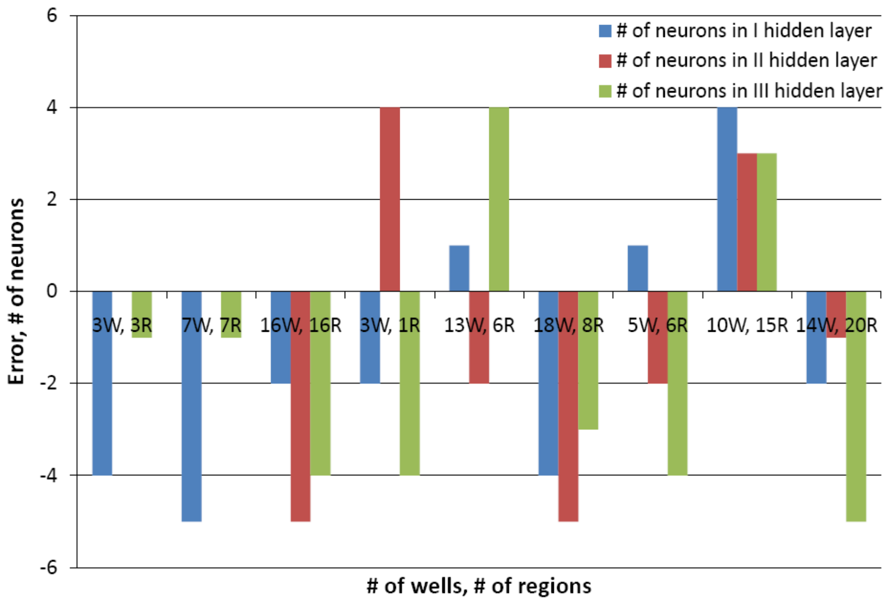


Figure **5-26**: Prediction errors for number of neurons required in each of the hidden layers in the prediction network.

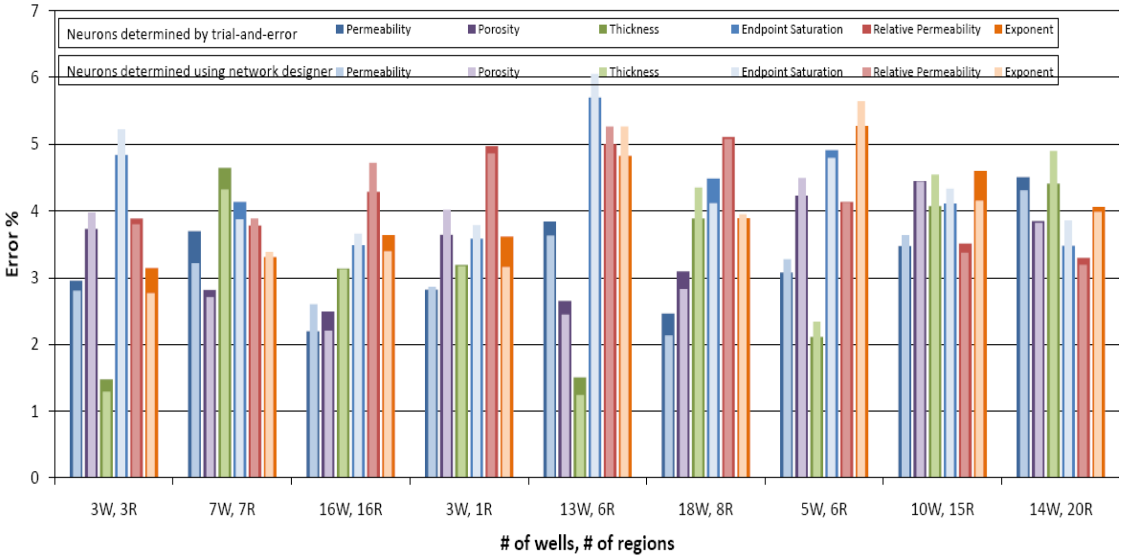


Figure **5-27**: Comparison of average prediction errors for each property using network architecture developed in Section **5.3.2.1** with number of neurons determined by trial-and-error method and number of neurons as calculated by network designer (reservoir with irregular boundary).

The number of neurons required in the prediction network for different reservoirs is determined by trial-and-error process. A small set of this data are used to train the network designer. The network designer is then used to calculate the number of neurons required for other reservoirs. Prediction errors of a sample of the reservoirs tested is shown in Figure **5-26**. Network designer is able to predict the number of neurons required in each of the three hidden layers in the prediction network within an error of ±5 neurons. Figures **5-27** shows a comparison of prediction errors for a sample of reservoirs tested using number of neurons in hidden layers determined by trial-and-error and number of neurons as calculated by network designer.

5.5 Guidelines for Development of ANN Prediction Tool

Artificial neural networks are widely used in a lot of petroleum engineering applications. But, lack of established guidelines for design of ANN prediction network has forced users to resort to trial-and-error procedure at every step along the development process. Based on the observations made during the development of ANN prediction tool for history matching, a set of guidelines have been developed. Design of ANN prediction tool can be broadly divided into 1. Network structure, 2. Input/output parameters, 3. Data formulation.

5.5.1 Network Structure

Artificial neural network has several neurons which are arranged in layers and the connections between these layers constitute the network architecture. They can be either a single layer network or a multilayer network. Single layer networks have one input layer and one output layer. Multilayer networks have one input layer, one output layer and they are connected by one or more hidden layers. Any problem can be solved provided there is enough number of neurons in the hidden layers. Most problems do not require more than four hidden layers [Fausett, 1994].

For history matching, it is has been shown that multilayer feedforward network with back propagation works well. The problem being solved requires three hidden layers for optimum performance. The actual number of neurons in each hidden layer depends on the number of input and output neurons. Hyperbolic tangents sigmoid (*tansig*) transfer function works well between all layers except output layer. Typically, linear transfer function like *purelin* is used in the output layer. As choice for training algorithm, Levenberg-Marquardt algorithm is highly recommended as a first-choice supervised algorithm. Although it does require more memory than other algorithms, it is one of the fastest backpropagation algorithms. It has been observed that using a performance goal between e^{-4} and e^{-5} works well. So, all through the study a performance goal of $5e^{-5}$ has been used.

5.5.2 Input/Output Parameters

A good starting point to begin would be to list the parameters present in the governing equations to solve the particular problem. In our case, for history matching process, mass balance equations are used. From this, preliminary list of parameters are complied. Output parameters are the easiest list to assemble because we already have our objectives defined. Most of the times both the input and output layers end up using extra parameters which are functions of other parameters. They are called functional links. They help amplify minor changes in parameters. This in-turn help the network understand the relationship between input and output values. Some of the functional links may have some physical meaning while others may look very random yet improve network performance. For example, functional link *t/365* which is nothing but time in years has helped improve network performance. But, there is no real explanation why time in days along with time in yrs used in input improves network performance. Functional links k/ϕ and kxh were introduced when more regions and wells were added to the reservoirs being studied. Now, the network not only predicts the property values k, ϕ and h, it also ensures that k/ϕ and kxh relationships are satisfied. This essentially acts as a mechanism to cross check predicted property values using other relationships. This type of reinforcing helps improve the network performance. Table **5-5** lists the input and output parameters used by network developed to predict history match parameters of reservoirs with different relative permeability curves.

Table **5-5**: Input and output parameters used in network used to predict history match parameters. (irregular boundary reservoirs with different sets of relative permeability curves).

Input Parameters	Output Parameters
t	k
ΔN_p	ϕ
ΔG_p	h
ΔW_p	k / ϕ
ΔP	$k \text{ x } h$
$t/365$	A / k
$t/365 \text{ x } \Delta N_p$	S_{orw}
WOR	S_{org}
GOR	S_{wirr}
	S_{gcrit}
	k_{rwro}
	k_{roirw}
	k_{rgro}
	$k_{rocritg}$
	N

5.5.3 Data Formulation

Data formulation is an important step that determines the performance and accuracy of predictions made by the network. This step involves making several critical decisions about the data being presented to the network during training phase of the process. Some of them significant ones are, 1. Data generation strategy, 2. Number of data sets to be presented to the network during training phase, 3. Order in which data are presented to the network.

5.5.3.1 Data Generation Strategy

Neural network works better at interpolation than extrapolation. So, the data presented to the network during training stage should cover the whole spectrum of parameter values in the output layer. In this study, data used in training were generated using a random number generator then manually inspected to ensure there is uniform distribution of properties.

5.5.3.2 Number of Data Sets

Number of data sets used in the training phase of the network development is very critical. Use of too few data sets may lead to a network that has not captured the actual relationship between input and output parameters. Use of too many data sets may lead to a network that memorizes the values rather than understanding the relationship between input and output parameters. Both scenarios are undesirable since neither will give good predictions for actual problem. Apart from this, time required to train the network also increases as the number of data sets increase. So, there is always incentive to reduce the number of data sets. It is essential to find an optimum number of data sets that give good results. In this study, 50 data sets were used in training.

5.5.3.3 Sort Order of Data in Data Sets

The order in which data are presented to the network can make some difference in the performance. But there is no documented evidence that suggests that change in sort order will make a big difference in network performance. In this study, it was observed that presenting all time steps of a single data set together gave slightly better results compared to sorting all the data based on time steps.

Chapter 6

IMPLEMENTATION OF ANN PREDITION TOOL TO ACTUAL FIELD DATA

The artificial neural network developed is implemented to actual field data obtained from Perry reservoir located in Brayton field. To gauge the performance of the developed artificial neural network, history match is performed using the on predicted properties.

6.1 Perry Reservoir in Brayton Field

Brayton field is located about 30 miles west of the City of Corpus Christi in the western Nueces County, Texas. Brayton field is a local productive area on the east flank of the Agua Dulce structure in District 4, Nueces County, Texas. This study will use field production data from Perry reservoir. It is located at a depth between 7100 to 7300 feet below the ground. It was first proved productive in late 1945.

Perry sand is a lenticular north-south strike oriented Lower Frio sand that is located in the Brayton field. The Perry sand dips uniformly east at 100 feet per mile. Figure **6-1** shows the structure contour map of Perry sand. Perry sand has a maximum thickness of 30 feet. Figure **6-2** shows the isopach map of Perry sand. The type of hydrocarbon trapping mechanism is a stratigraphic pinchout or strandline re-entry from eastern direction. This sand as well as other overlying sands was deposited in a lagoonal environment as wash over sands or bar sands that were confined by adjacent lagoonal muds. Perry sand has good porosity and permeability

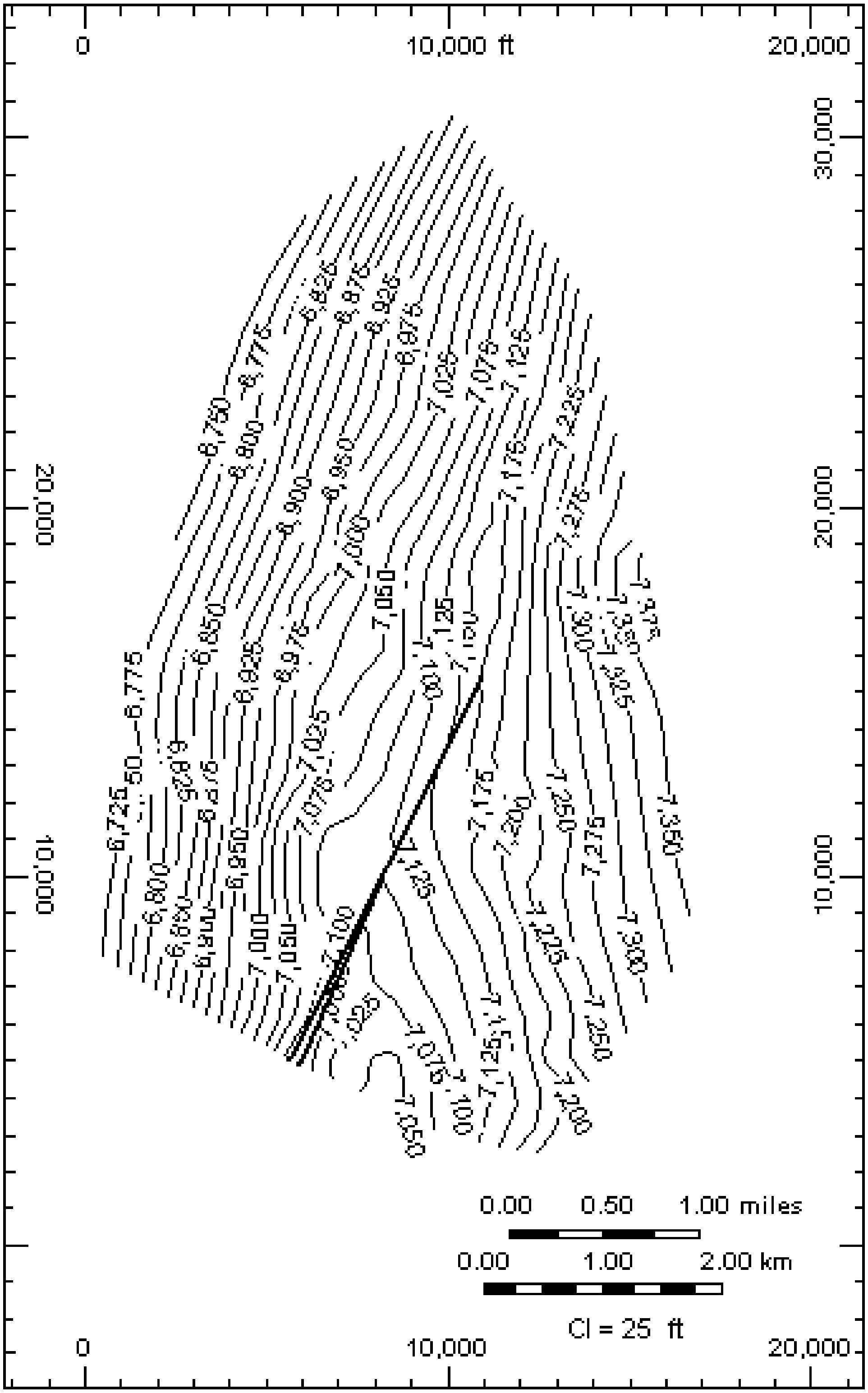


Figure **6-1**: Structural contour map on top of Perry sand.

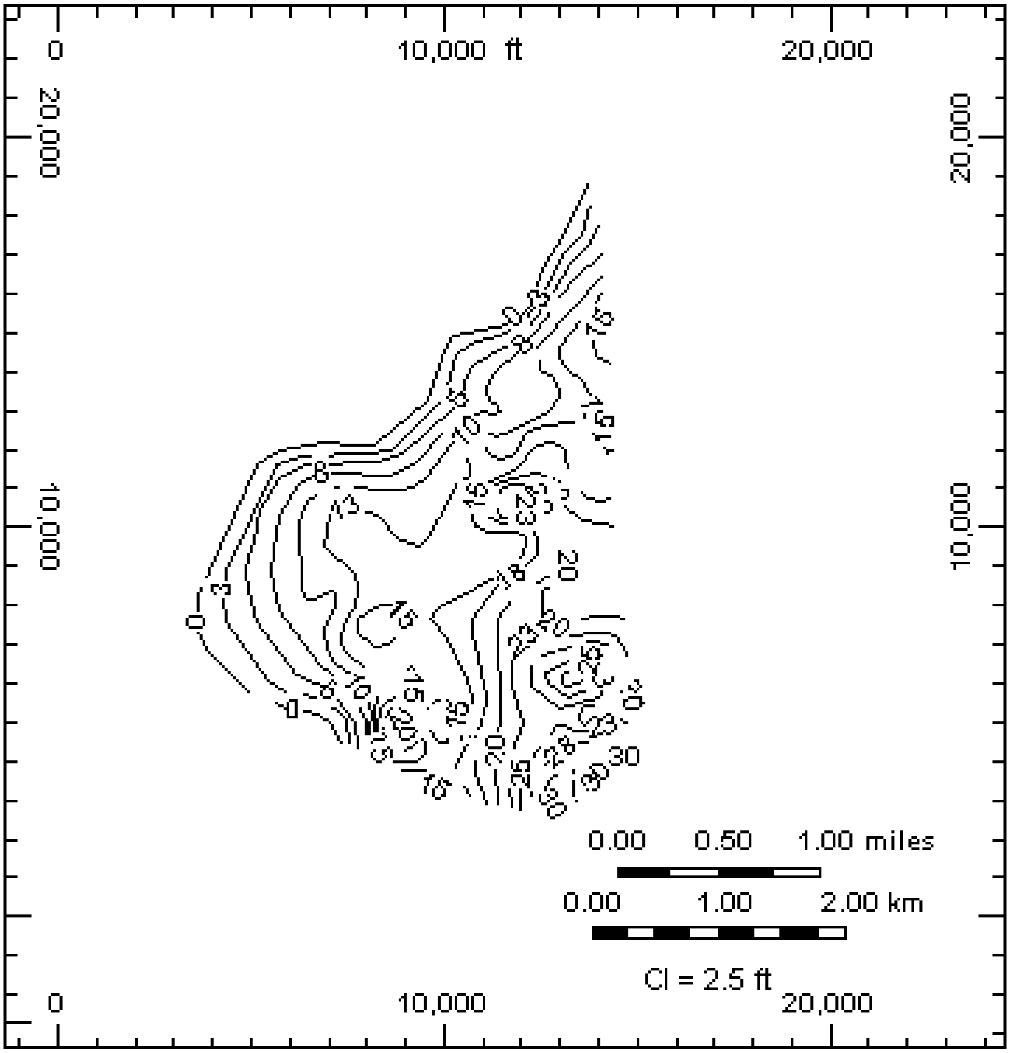


Figure **6-2**: Isopach map of Perry sand.

6.2 Implementation of Artificial Neural Network

Before implementing the artificial neural network developed for predicting properties of the Perry reservoir, it has to be classified based on the number of wells and number of regions. Perry sand has 18 producing wells. From the Isopach map, the reservoir was delineated into four regions. Figure **6-3** shows well locations in Perry reservoir on top structure contour map. Figure **6-4** shows Perry reservoir delineated into 4 regions on isopach map.

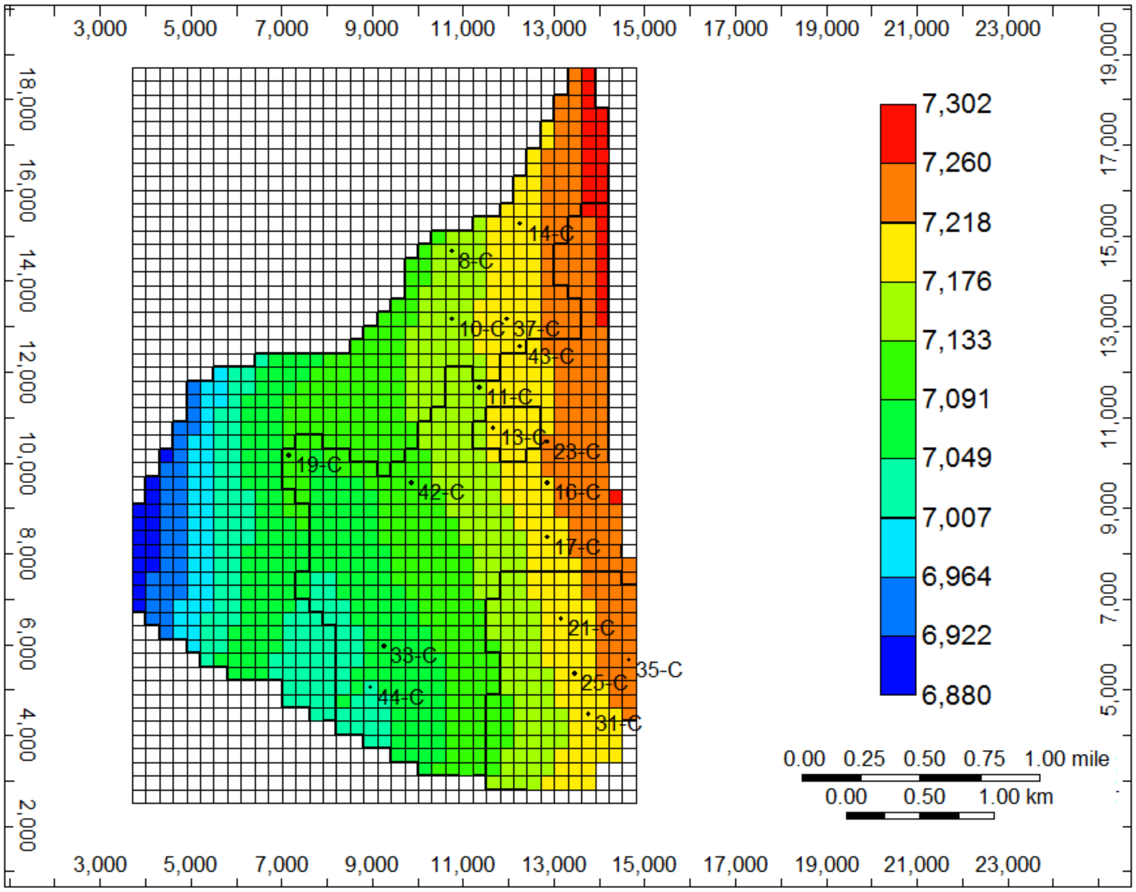


Figure **6-3**: Well locations in Perry reservoir on top structure contour map.

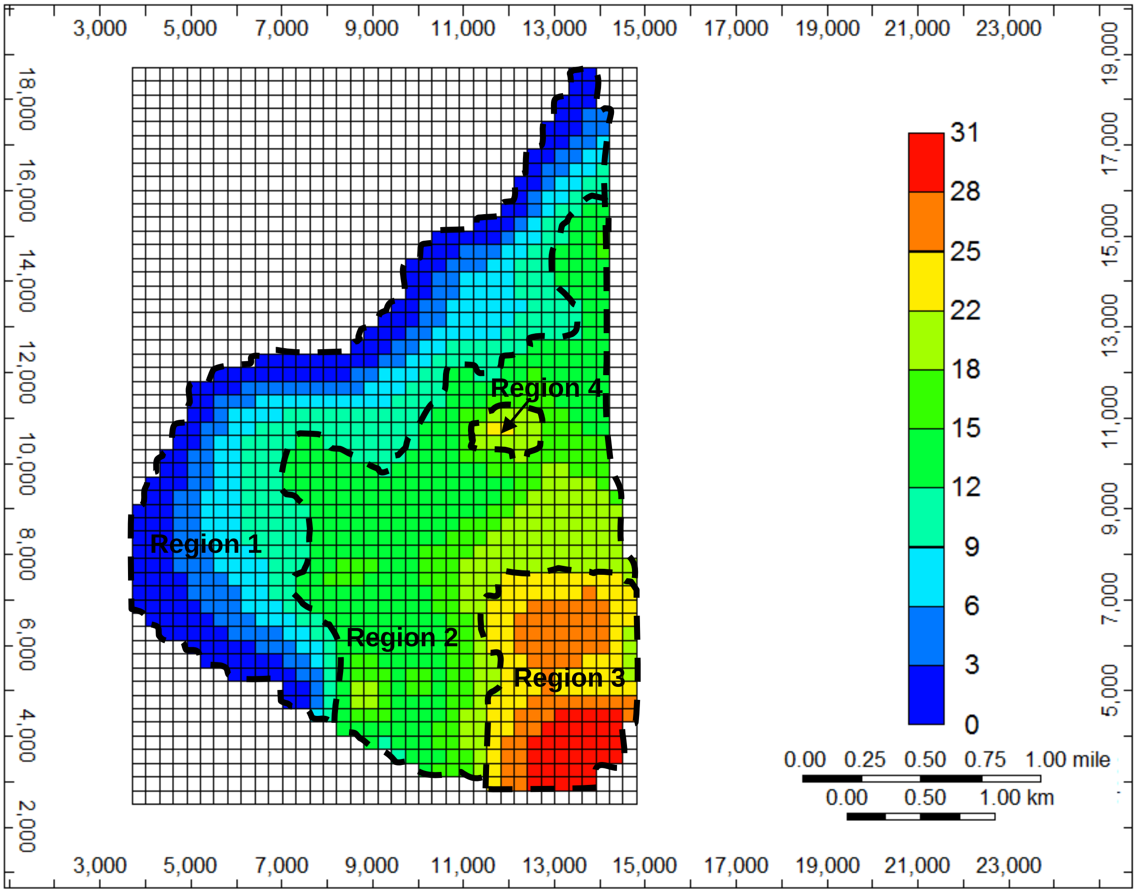


Figure **6-4**: Perry reservoir delineated into 4 regions on isopach map.

6.2.1 Training the Artificial Neural Network

Next step is to generate training data set. Training data set are generated using 50 different reservoirs with different porosity, permeability, net pay thickness and relative permeability curves. The rock and fluid properties used to build the simulation model for Perry sand is outlined in Appendix E. Production data obtained from these reservoirs simulations becomes the training data. They are formatted into the pattern and target data. They are used to train the multilayer feedforward network with back propagation. Network is trained until a performance goal of $5e^{-5}$ is achieved. Thus we have a trained network that can be used to predict the properties of Perry reservoir.

6.2.2 Predicting the Artificial Neural Network

Production data from the field are formatted into the target data. This constitutes the input data to our prediction network. Prediction network is used to predict the porosity, permeability, net pay thickness of the four regions along with parameters required for constructing two-phase relative permeability curves. Table **6-1** shows the predicted property values.

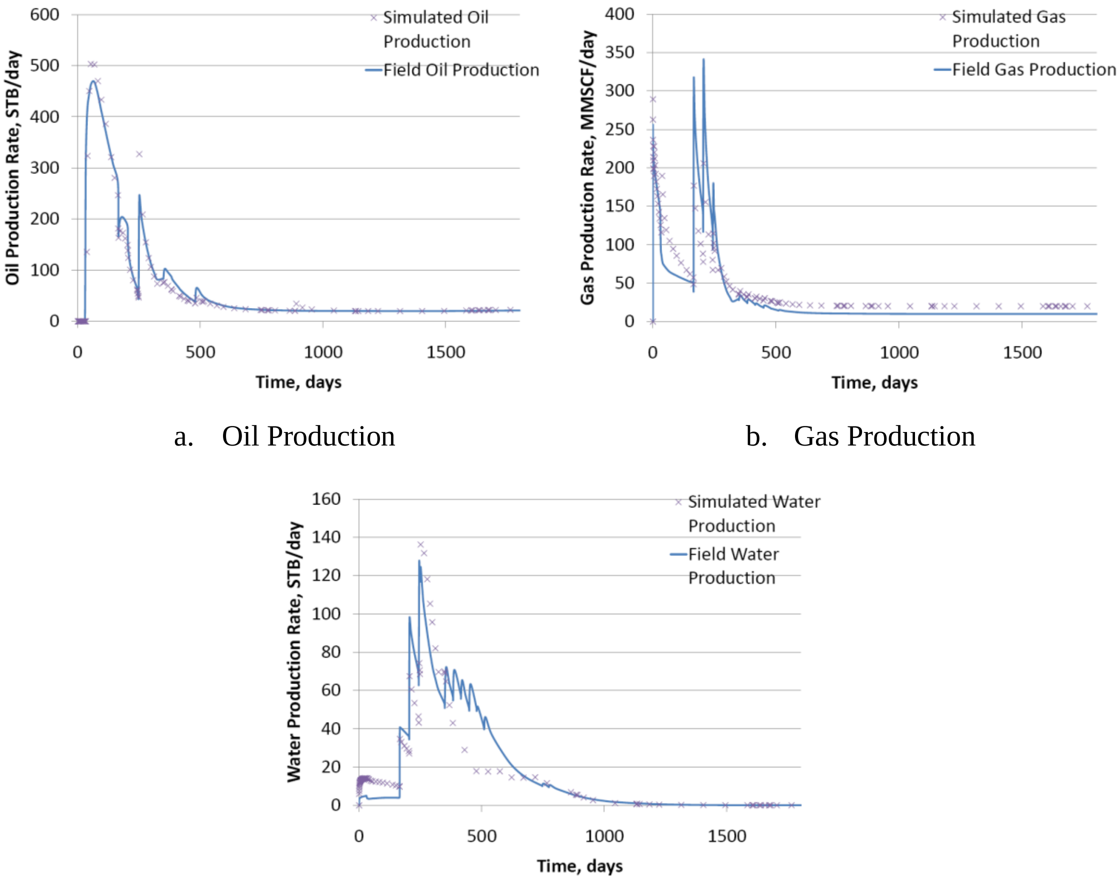
Table **6-1**: Property values for Perry reservoir predicted by artificial neural network.

Region	Porosity	Permeability	Net Pay Thickness
1	0.269	1652	8.3
2	0.281	1684	18.2
3	0.275	1671	29.4
4	0.284	1677	22.8
Sorw = 0.41	Sorg = 0.37	Swirr = 0.29	Sgcrit = 0.06
Krwro = 0.73	Krgro = 0.66	Kroirw = 0.92	Krocritg = 0.91
N = 3.1			

In a separate study, Al-Saadoon et.al (1990) had evaluated various enhanced oil recovery schemes for Simmonds and Perry reservoirs. In that study, it has been shown that an average porosity of 0.28 and an average permeability of 1672md in the Perry reservoir gave good history match. The porosity and permeability values predicted by the artificial neural network are close to the values from the EOR study. This may indicate that a good history match is likely to be obtained using predicted property values.

6.2.3 History Matching

Using the predicted properties a reservoir simulation model is built. The model is run for 1800 days of production. Then, the production results from the model are compared to the field production data. Figure 6-4 shows comparison of production data.



a. Oil Production

b. Gas Production

c. Water Production

Figure **6-5**: Simulated production profiles for the Perry reservoir built using ANN predicted properties compared to field production data.

From Figure 6-4 it is clear that model production and field production are in close agreement with each other. This can be called a successful history match. Thus, the model created using predicted properties is a good history match to the actual reservoir.

6.3 Summary

The neural network developed in section **5.3.2.1** has been successfully implemented to predict properties of Perry reservoir using actual field data. Training data were generated for the reservoir with 18 wells and 4 regions. Artificial neural network was trained using this training data. The trained network was then used in prediction mode with actual field data to predict the properties of Perry reservoir. Predicted properties were used to build a simulation model whose production is compared with actual field production. The properties predicted by the artificial neural network have provided a good history match to the field production data. Thus it has been shown through this field application that the artificial neural network developed in this study can serve as an effective tool in predicting the properties of actual reservoirs.

Chapter 7

CONCLUSIONS

7.1 Conclusions

In this study, an artificial neural network tool was developed that can help simulation engineers obtain a good history match for black oil reservoirs with fewer simulation runs. This tool is capable of predicting porosity (ϕ), permeability (k), net pay thickness (h) for each zone/region and parameters required to construct the two phase relative permeability curves (S_{orw}, S_{org}, S_{wirr}, S_{gcrit}, k_{rwro}, k_{roirw}, k_{rgro}, $k_{rocritg}$, N). Cumulative production and pressure data constitute input data to the tool. The property and parameter values predicted by the tool will provide a good history match or the least serve as a good starting point for history matching. The main advantage of this tool is that it doesn't require an initial estimate of properties and parameters to obtain a good history match. This tool actually obtains a good history match with fewer simulation runs compared to conventional history matching techniques. Using this tool, only a finite number of simulation runs (in this study, typically 50 sets of training data were used), to generate synthetic training data, are required to obtain a good history match. While by conventional history matching techniques, a few hundred runs could be required, especially when a good initial estimate of these properties and parameters are not available.

Implementation of the tool for history matching involves two steps, 1. Training of network using synthetic data, 2. Prediction of property and parameter values of actual reservoir. During the training stage, network is shown several sets of synthetic data. Once the network is trained, it can be used to predict property and parameter values of actual reservoir by providing production and pressure data as input to the network. The predicted values should be useful in building a model that provides good history match to the actual reservoir.

The number of neurons required in each of the hidden layers of the artificial neural network tool developed depends on the number of wells and regions in the reservoir. A separate tool called network designer was developed to determine the number of neurons required. Network designer takes number of wells and regions as input to predict the number of neurons in each hidden layer.

The developed tool was successfully implemented to a real field. Field data from Perry reservoir was tested using the prediction tool. History match was performed using properties predicted by the tool. It has been shown that a good history match has been obtained between the model and field production data.

The major conclusions from this study are as follow:

1. Artificial neural network tool using multilayer feedforward back propagation network algorithm is effective in history matching applications.
2. The constructed prediction tool is capable of predicting the history match parameters porosity, permeability, net pay thickness, endpoint saturations, relative permeability at endpoint saturations and exponent for black oil reservoirs within acceptable margins of error.
3. A good history match to Perry reservoir was obtained using the artificial neural network tool developed. Thus it has been shown through this field application that the artificial neural network developed in this study can serve as an effective tool in predicting the properties of actual reservoirs.
4. Network designer is effective in designing the prediction network.
5. Prediction network designed by network designer is capable of predicting history match properties within acceptable margins of error.

6. When the number of wells is greater than or equal to number of regions, history match parameters can be predicted within acceptable margins of error.
7. When the number of wells is less than number of regions, prediction errors are high and continue to increase as the difference between number of wells and number of regions increase.
8. Production data input to the network at every 90 day intervals for the 1800 days effectively trained the network while reducing training times compared to production inputs at 30 day intervals for 3600 days.
9. Levenberg-Marquardt backpropagation (*trainlm*) is an effective and fast training function.
10. Functional links t/365, t/365 x Np, WOR and GOR in input layer are most effective.
11. Functional links k/f, k x h and A/k in output layer are most effective.

7.2 Recommendations for Future Work

The study can be further improved with the following potential research:

- Initial conditions of the reservoir like reservoir pressure, saturation, water-oil contact, gas-oil contact; rock-fluid properties like capillary pressure curves; and PVT data are some of the parameters that are usually adjusted during history matching process. Including some of these to the list of parameters to be predicted would be good logical next step.
- Current study works well with reservoirs upto 20 wells and 20 regions. It is essential to generalize the network structure such that the limit on the number of

wells and number of regions that can be handled by the network is expanded or removed.

- Performing a sensitivity analysis on the amount of production history required by the neural network could help reduce the amount of production history provided to the neural network. The amount of production history required may depend on the size of the reservoir itself. Network should be analyzed from that perspective as well.
- Some of the properties predicted like porosity may be known to reservoir engineer. There is an option of keeping that value constant across the training data set and in-turn reduce the number of data sets used in training. A protocol may be developed for performing such changes.
- What happens when some production data is not available? Can the network still be trained to predict the properties?
- Current study involves only reservoir with producing wells only. Exploring reservoirs with injectors and producers would be good next step.
- Current study has history matched black oil reservoir. The possibility of expanding it to compositional and may be even thermal reservoirs can be explored.

Bibliography

Al-Fattah, S.M. Startzman, R.A.: *Predicting Natural Gas Production Using Artificial Neural Network*, paper SPE 68593, 2001.

Al-Saadoon, F.T., Perry, J.M., The M.G. and Johnnye D. Perry Foundation: *Cooperative Study of Enhanced Oil Recovery, Simmonds and Perry Reservoir, Nueces County, Texas*, report, March 1990

Ali, J.K.: *Neural Networks: A New Tool for the Petroleum Industry?*, paper SPE 27561, 2006.

Artun, F.E.: *Optimized Design of Cyclic Pressure Pulsing in Naturally Fractured Reservoirs Using Neural-Network Based Proxy Models – a dissertation in Petroleum and Natural Gas Engineering*. The Pennsylvania State University, 1998.

Ayala, L.F., Ertekin, T. and Adewumi, M.: *Optimized Exploitation of Gas-Condensate reservoir Using Neuro-Simulation*, paper SPE 88471 presented at the Asia Pacific Oil and Gas Conference and Exhibition, Perth, Australia,18-20 October, 2004.

Aziz, K.: *Ten Golden Rules for Simulation Engineers,* JPT (November 1989) 1157.

Bear, J.: *Dynamics of Fluids in Porous Media*, Dover Publishing, Inc., New York, ISBN 0486-65675-6, 1972.

Centilman, A.: *Applications of Neural-networks in Multi-well Field Development – a thesis in Petroleum and Natural Gas Engineering*. The Pennsylvania State University, 1999.

Chang. S, Grigg, R.B., Sung, A.: *Use of MASTER Web to Improve History Matching*, paper SPE 62617 presented at the 2000 SPE/AAPG Western Regional Meeting, Long Beech, CA, 19-23 Junes 2000.

Coalson, E.B., Hartmann, D.J., and Thomas, J.B.: *Applied Petrophysics in Exploration and Exploitation*: Notes from short course sponsored by University of Colorado, Denver, 1990.

Doraisamy, H., Ertekin, T., Grader, A.S.: *Key Parameters Controlling the Performance of Neuro-Simulation Applications in Field Development*, paper SPE 51079 presented at SPE Eastern Regional Meeting, Pittsburgh, Pennsylvania, 9-11 November 1998.

Dye, L.W., Horne, R.N. and Aziz, K.: *A New Method for Automated History Matching of Reservoir Simulators*, paper SPE 15137 presented at the 56th California Regional Meeting of the Society of Petroleum Engineers Held at Oakland, CA, 2-4 April.

Ertekin, T., Abou-Kassem, J. and King, G.: *Basic Applied Reservoir Simulation,* SPE Textbook Series Vol. 7. Richardson, TX, ISBN 1-55563-089-8, 2001.

Fanchi, J.R.: *Principles of Applied Reservoir Simulation*, Gulf Publishing Company. Houston, TX, ISBN 0-88415-117-4, 1997.

Fanchi, J.R.: *Principles of Applied Reservoir Simulation*, Third Edition, Gulf Professional Publishing. Burlington, MA, ISBN 0-7506-7933-6, 2006.

Fausett, Laurene: *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*, Prentice-Hall, Inc., New Jersey, ISBN 0-13-334186-0, 1994.

Gorucu, F.B., Ertekin, T., Bromhal, G. S., Smith, D. H., Sams, W.N., Jikich, S.: *A Neurosimulation Tool for Predicting Performance in Enhance Coalbed Methane and CO2 Sequestration Projects*, SPE paper 97164 presented at the 2005 SPE Annual Technical Conference and Exhibition held in Dallas, Texas, 9-12 October, 2005.

Hagan, M.T., Demuth, B.T. and Beale, M.H.: *Neural Network Design*, PWS Publishing Company, ISBN 0-9717321-0-8

Hartmann, D.J., Beaumont, E.A. and Coalson, E.: *Predicting Sandstone Reservoir System Quality and Example of Petrophysical Evaluation*, Search and Discovery Article #40005, 2000

Hyne, N.J.: *Dictionary of Petroleum Exploration, Drilling & Production*, PennWell Books, ISBN 978-0-87814-352-8, 1991.

Maren, A., Harston, C and Pap, R.: *Handbook of Neural Computing Applications*, Academic Press, Inc., San Diego, ISBN 0-12-546090-2, 1990.

Mattax, C.C. and Dalton, R.L.: *Reservoir Simulation, Monograph Series*, SPE, Richardson, TX (1990) 13.

Mohaghegh, S.: *Virtual-Intelligence Applications in Petroleum Engineering, Parts 1-3* papers SPE 58046, 61925, 62415, Distinguished Author Series, 2000.

Parada, C.H.M.: *An Artificial Neural network Based Tool-Box for Screening and Designing Improved Oil Recovery Methods – a dissertation in Petroleum and Natural Gas Engineering*. The Pennsylvania State University, 2008.

Parish, R.G., Watkins, A.J., Muggeridge, A.H., Goode, A.T. and Robinson, P.R.: *Effective History Matching: The Application of Advanced Software Techniques to the History-Matching Process*, paper SPE 25250 presented at the 12th SPE symposium on Reservoir Simulation in New Orleans, LA, 28 February – 3 March 1993.

Ramgulam, A.: *Utilization of Artificial Neural Networks in the Optimization of History Matching – a thesis in Petroleum and Natural Gas Engineering*. The Pennsylvania State University, 2006.

Raza, S.H.: *Data Acquisition and Analysis for Efficient Reservoir Management*, JPT (1992) , p. 466-468.

Saleri, N.G. and Toronyi, R.M.: *Engineering Control in Reservoir Simulation*, paper SPE 18305 presented at the SPE Annual Technical Conference and Exhibition, Houston, TX, 2-5 October, 1998.

Appendix A

Reservoir Rock and Fluid Properties Used to Build Reservoir Models

This section describes the parameters used in the reservoir model that is used generate synthetic data for training and testing of neural network.

A.1 Initialization Data

Table **A-1**: Rock and fluid data used in the initialization of reservoir model.

Type of fluid model	Black oil
Stock tank water density	61.9073 lb/ft3
Water salinity	20,000 ppm
Water formation volume factor	1.01501
Water viscosity	0.440258 cp
Water compressibility	3.3225e^{-006} 1/psi
Rock compressibility	3.2 e^{-006} 1/psi
Reservoir pressure	1000 psi
Reservoir temperature	158 ^{o}F
Gas gravity	0.65
Oil density	53.0013 lb/ft3

A.2 Rock Property Data

Table **A-2**: Endpoint input data used to generate relative permeability curves.

Connate water saturation	0.2
Critical water saturation	0.2
Irreducible oil saturation for water-oil table	0.4
Residual oil saturation for water-oil table	0.4
Irreducible oil saturation for gas-liquid table	0.2
Residual oil saturation for gas-liquid table	0.2
Connate gas saturation	0.05
Critical gas saturation	0.05
Oil relative permeability at connate water saturation	0.8
Water relative permeability at irreducible oil saturation	0.3
Gas relative permeability at connate liquid saturation	0.3
Gas relative permeability at connate gas saturation	0.8
Exponent for calculating krw	2
Exponent for calculating krow	2
Exponent for calculating krog	2
Exponent for calculating krg	2

Table **A-3**: Two-phase relative permeability data.

Water–Oil Table		
S_w	K_{rw}	K_{row}
0.200	0.00000	0.80000
0.225	0.00117	0.70313
0.250	0.00469	0.61250
0.275	0.01055	0.52813
0.300	0.01875	0.45000
0.325	0.02930	0.37813
0.350	0.04219	0.31250
0.375	0.05742	0.25313
0.400	0.07500	0.20000
0.425	0.09492	0.15313
0.450	0.11719	0.11250
0.475	0.14180	0.07813
0.500	0.16875	0.05000
0.525	0.19805	0.02813
0.550	0.22969	0.01250
0.575	0.26367	0.00313
0.600	0.30000	0.00000

Gas-Oil Table		
S_g	K_{rg}	K_{rog}
0.050	0.00000	0.80000
0.084	0.00117	0.70313
0.119	0.00469	0.61250
0.153	0.01055	0.52813
0.188	0.01875	0.45000
0.222	0.02930	0.37813
0.256	0.04219	0.31250
0.291	0.05742	0.25313
0.325	0.07500	0.20000
0.359	0.09492	0.15313
0.394	0.11719	0.11250
0.428	0.14180	0.07813
0.463	0.16875	0.05000
0.497	0.19805	0.02813
0.531	0.22969	0.01250
0.566	0.26367	0.00313
0.600	0.30000	0.00000

A.3 Black Oil PVT Data

Table **A-4**: Black oil PVT data used in the reservoir simulator to generate synthetic data.

P, psi	R_s, (SCF/STB)	B_o, (RB/STB)	B_g, (RB/MSCF)	μ_o, (cp)	μ_g, (cp)	co, 1/psi
14.70	3.81	1.0455	0.2107	2.5412	0.0125	3.00E-05
76.57	11.72	1.0484	0.0402	2.4081	0.0125	3.00E-05
138.44	20.74	1.0517	0.0221	2.2746	0.0126	3.00E-05
200.31	30.51	1.0553	0.0151	2.1482	0.0127	3.00E-05
262.18	40.84	1.0592	0.0115	2.0311	0.0127	3.00E-05
324.05	51.65	1.0632	0.0092	1.9236	0.0128	3.00E-05
385.92	62.86	1.0675	0.0077	1.8253	0.0129	3.00E-05
447.79	74.41	1.0719	0.0066	1.7355	0.0130	3.00E-05
509.66	86.29	1.0765	0.0058	1.6535	0.0131	3.00E-05
571.53	98.44	1.0812	0.0051	1.5786	0.0132	3.00E-05
633.40	110.86	1.0861	0.0046	1.5099	0.0133	3.00E-05
695.27	123.52	1.0910	0.0041	1.4468	0.0134	3.00E-05
757.14	136.41	1.0962	0.0038	1.3887	0.0135	3.00E-05
819.01	149.50	1.1014	0.0035	1.3352	0.0136	3.00E-05
880.88	162.80	1.1068	0.0032	1.2856	0.0137	3.00E-05
942.75	176.28	1.1122	0.0030	1.2397	0.0139	3.00E-05
1769.46	370.83	1.1948	0.0015	0.8457	0.0161	2.45E-05
2596.18	585.33	1.2923	0.0010	0.6503	0.0190	1.49E-05
3422.89	814.37	1.4023	0.0008	0.5337	0.0222	1.04E-05
4249.61	1055.03	1.5233	0.0007	0.4559	0.0252	7.86E-06
5076.32	1305.48	1.6542	0.0006	0.4002	0.0280	6.25E-06

Appendix B

Sample Data

Table **B-1**: Sample training input data - *Data structure 1*.

#	t, days	ΔN_p, bbl	ΔW_p, bbl	ΔG_p, bbl	ΔP, psi	t/365, yrs	t/365 x ΔN_p, bbl x yr
1	90	-47165.1	499.908	1.22E+08	-197.306	0.246575	-11629.8
2	180	-44977	-2486.6	-4E+07	19.856	0.493151	-22180.4
3	270	-37678	-1649.4	-4.7E+07	34.351	0.739726	-27871.4
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
18	1620	-1764	-184.2	-1499000	1.5223	4.438356	-7829.26
19	1710	-1609	-169.8	-1322000	1.3309	4.684932	-7538.05
20	1800	-1475	-157	-1172000	1.1748	4.931507	-7273.97
21	90	47639.07	2522.356	1.29E+08	-42.915	0.246575	11746.62
22	180	-5592	469.35	50548000	13.255	0.493151	-2757.7
23	270	-2768	277.1	22947000	12.808	0.739726	-2047.56
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
38	1620	29	19.2	266000	-0.0566	4.438356	128.7123
39	1710	15	15.7	219000	-0.0511	4.684932	70.27397
40	1800	4	12.7	182000	-0.0438	4.931507	19.72603
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
981	90	-56098.8	-1168.03	13194090	-120.603	0.246575	-13832.6
982	180	-62105	-2873.6	-1.7E+08	18.735	0.493151	-30627.1
983	270	-17643	-1365.1	-9.3E+07	15.551	0.739726	-13051
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
998	1620	-917	-138.7	-1924000	1.0169	4.438356	-4069.97
999	1710	-835	-126.5	-1663000	0.893	4.684932	-3911.92
1000	1800	-764	-116	-1464000	0.79	4.931507	-3767.67

Table **B-2**: Sample training output data - *Data structure 1*.

#	k, md	ϕ	h, ft
1	1326	0.31	93
2	1326	0.31	93
3	1326	0.31	93
:	:	:	:
:	:	:	:
:	:	:	:
18	1326	0.31	93
19	1326	0.31	93
20	1326	0.31	93
21	1058	0.2	88
22	1058	0.2	88
23	1058	0.2	88
:	:	:	:
:	:	:	:
:	:	:	:
38	1058	0.2	88
39	1058	0.2	88
40	1058	0.2	88
:	:	:	:
:	:	:	:
:	:	:	:
981	652	0.22	148
982	652	0.22	148
983	652	0.22	148
:	:	:	:
:	:	:	:
:	:	:	:
998	652	0.22	148
999	652	0.22	148
1000	652	0.22	148

Table **B-3**: Sample training input data - *Data structure 2*.

#	t, days	ΔN_p, bbl	ΔW_p, bbl	ΔG_p, bbl	ΔP, psi	t/365, yrs	t/365 x ΔN_p, bbl x yr
1	90	-47165.1	499.908	1.22E+08	-197.306	0.246575	-11629.8
2	90	47639.07	2522.356	1.29E+08	-42.915	0.246575	11746.62
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
50	90	-56098.8	-1168.03	13194090	-120.603	0.246575	-13832.6
51	180	-44977	-2486.6	-4E+07	19.856	0.493151	-22180.4
52	180	-5592	469.35	50548000	13.255	0.493151	-2757.7
42	180	-62105	-2873.6	-1.7E+08	18.735	0.493151	-30627.1
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
998	1800	-1475	-157	-1172000	1.1748	4.931507	-7273.97
999	1800	4	12.7	182000	-0.0438	4.931507	19.72603
1000	1800	-764	-116	-1464000	0.79	4.931507	-3767.67

Table **B-4**: Sample training output data - *Data structure 2*.

#	k, md	ϕ	h, ft
1	1326	0.31	93
2	1058	0.2	88
:	:	:	:
:	:	:	:
:	:	:	:
50	652	0.22	148
51	1326	0.31	93
52	1058	0.2	88
42	652	0.22	148
:	:	:	:
:	:	:	:
:	:	:	:
998	1326	0.31	93
999	1058	0.2	88
1000	652	0.22	148

Appendix C

Prediction Errors

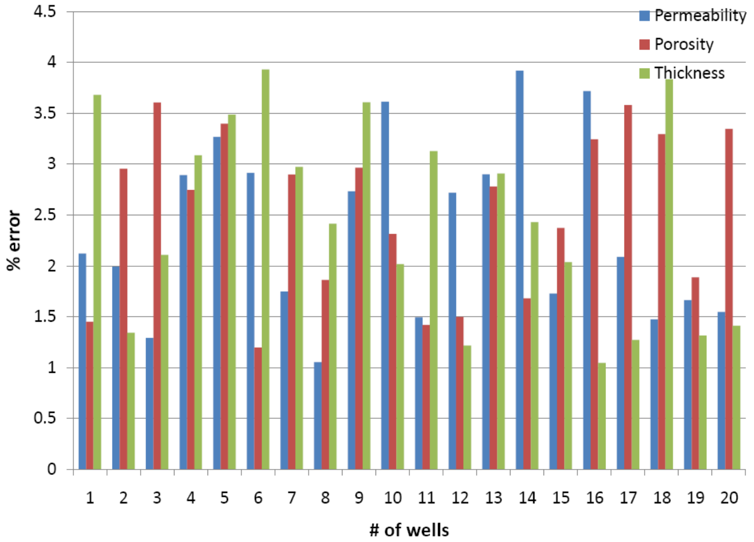


Figure **C-1:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 1 region system).

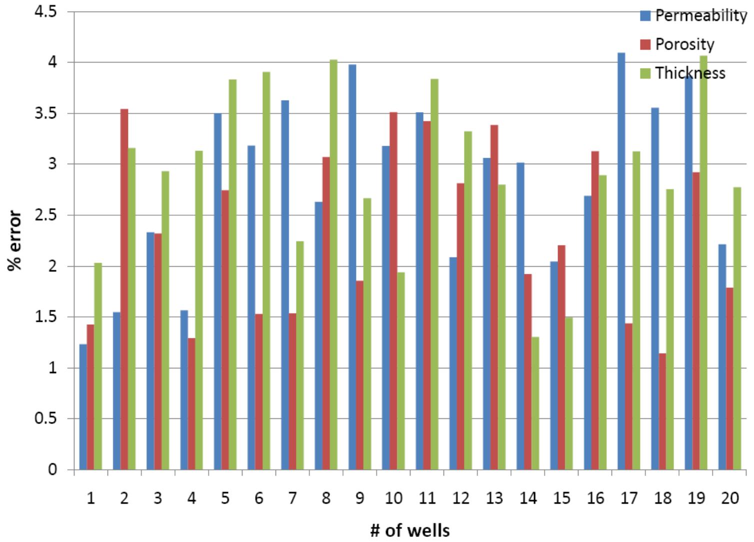


Figure **C-2:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 2 region system) (cont'd).

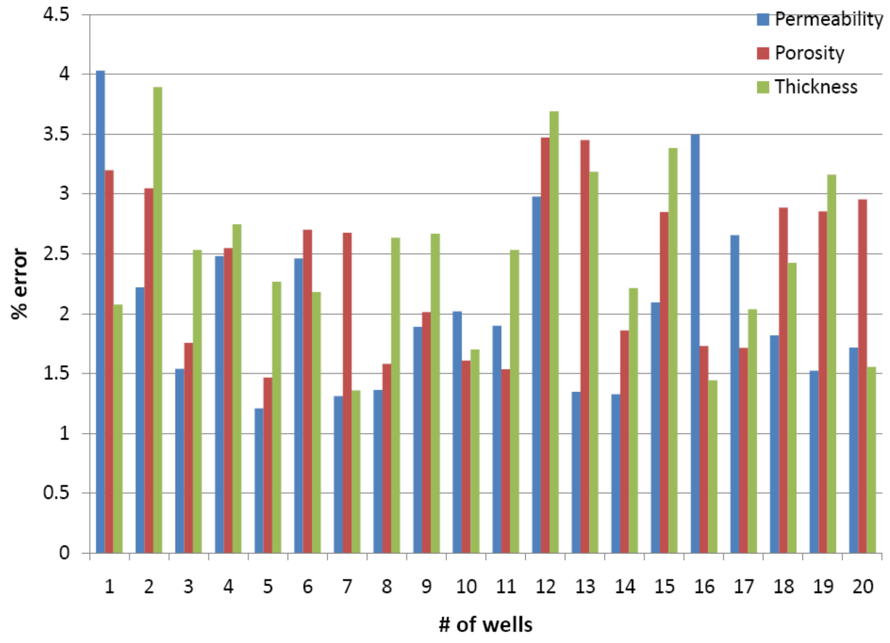


Figure **C-3:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 3 region system) (cont'd).

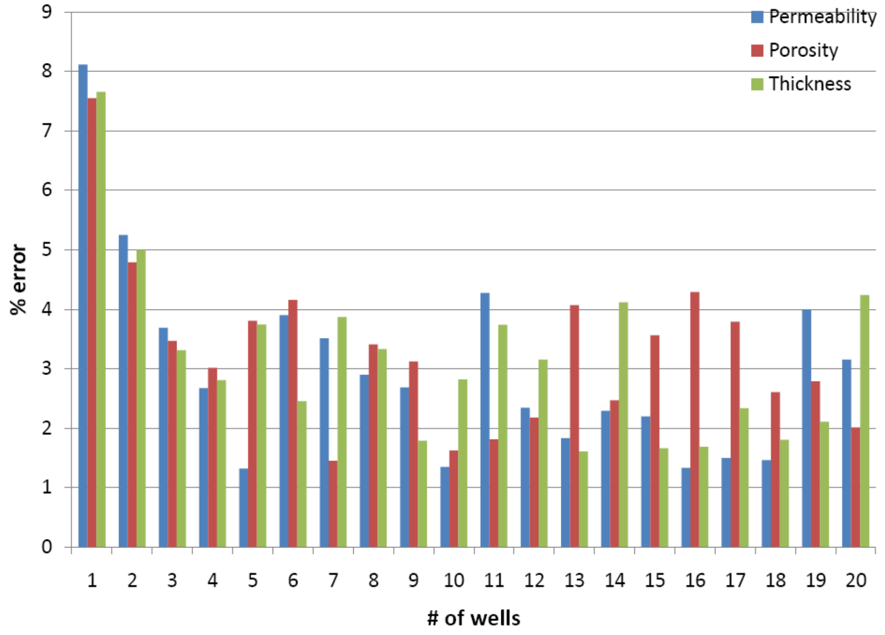


Figure **C-4:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 4 region system) (cont'd).

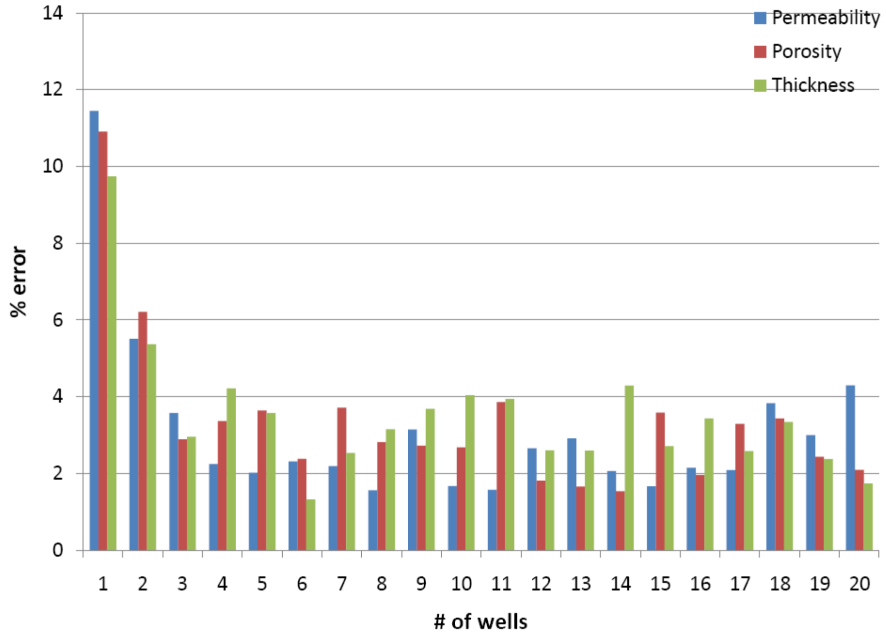


Figure **C-5:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 5 region system) (cont'd).

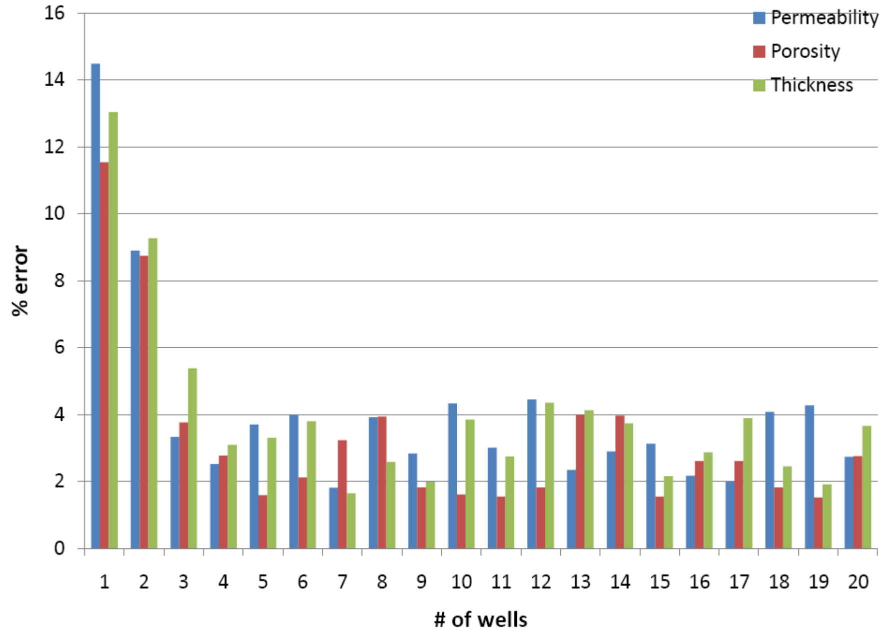


Figure **C-6:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 6 region system) (cont'd).

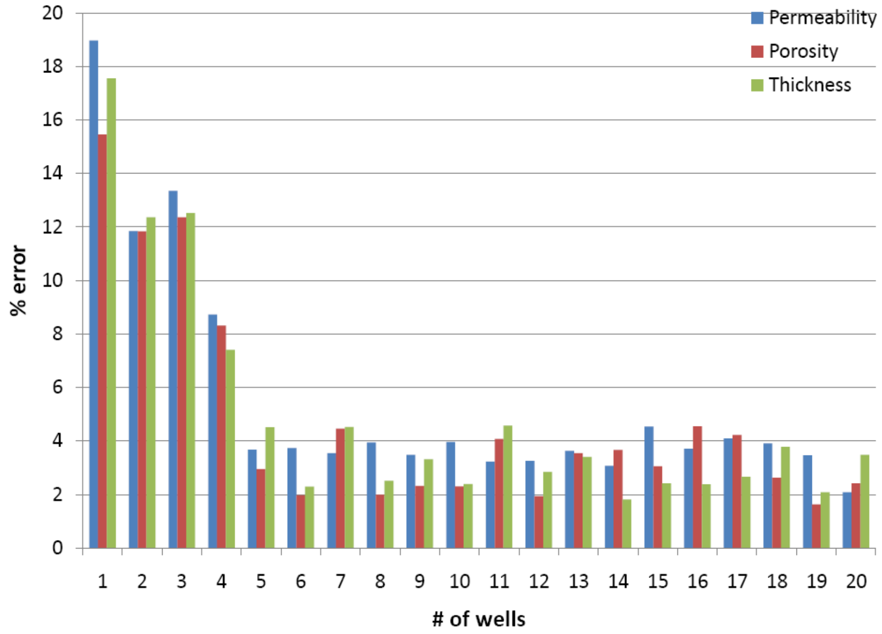


Figure **C-7:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 7 region system) (cont'd).

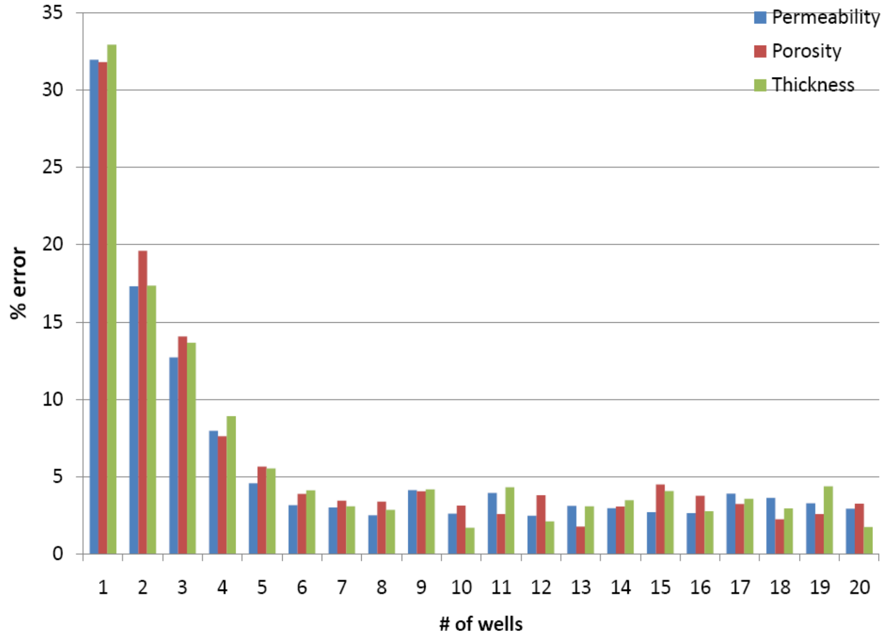


Figure **C-8:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 8 region system) (cont'd).

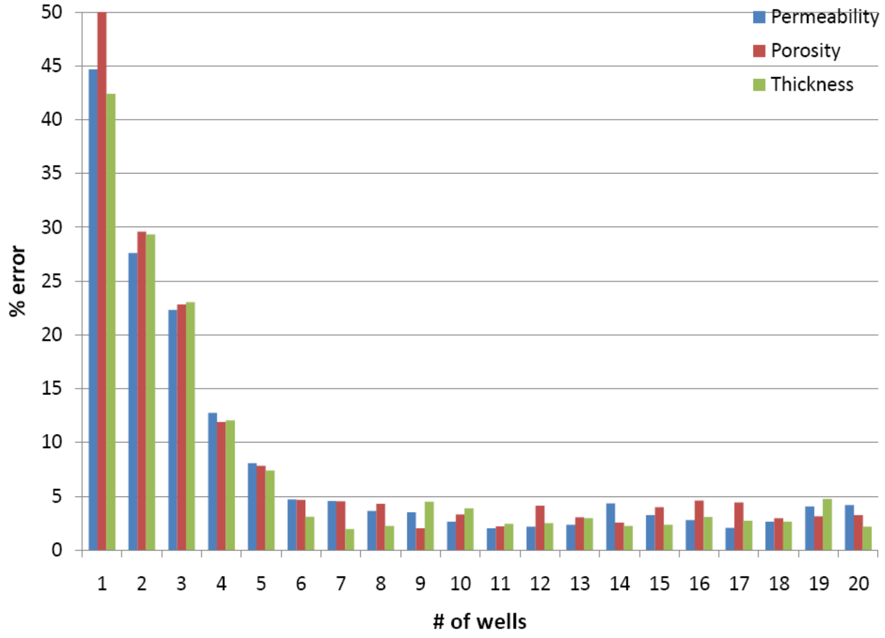


Figure **C-9:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 9 region system) (cont'd).

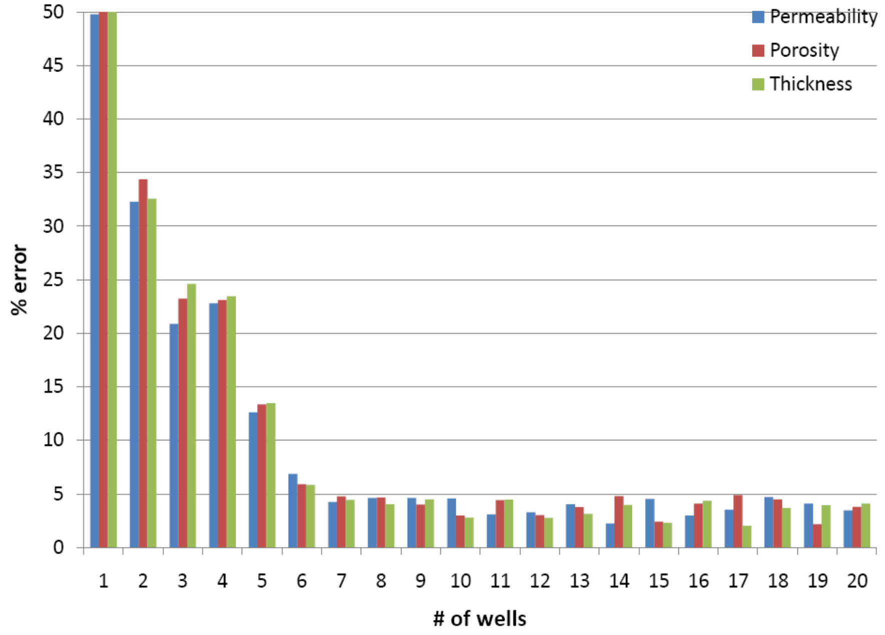


Figure **C-10:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 10 region system) (cont'd).

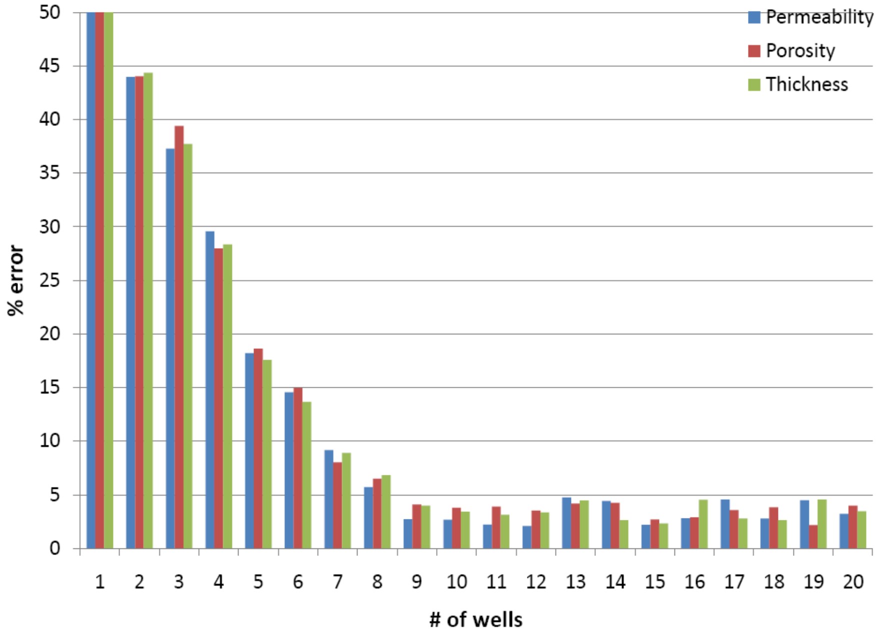


Figure **C-11:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 11 region system) (cont'd).

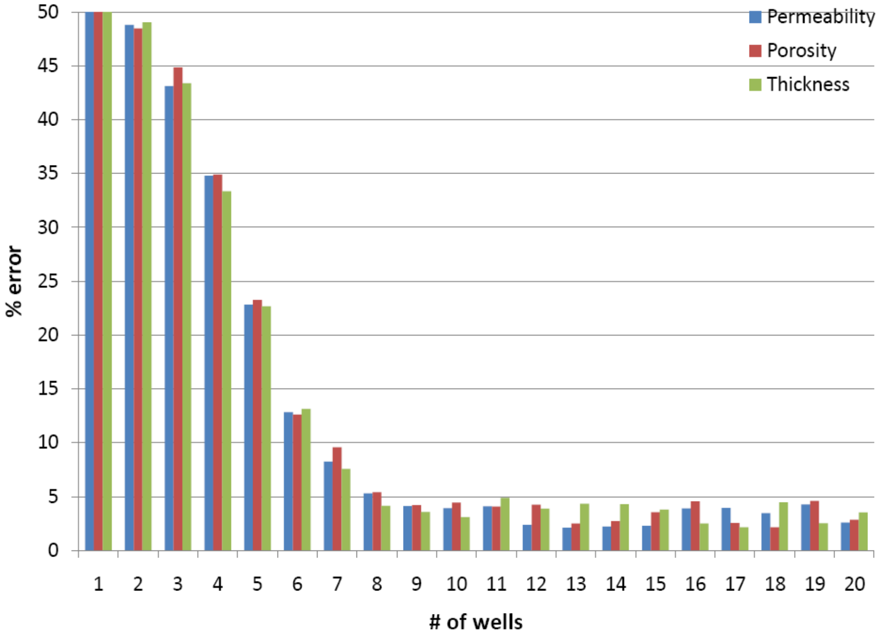


Figure **C-12:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 12 region system) (cont'd).

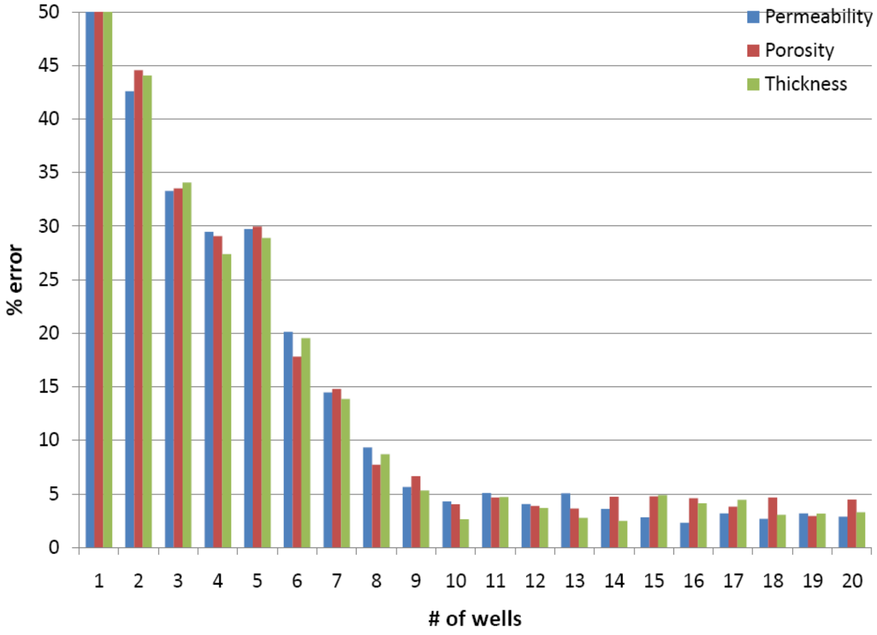


Figure **C-13:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 13 region system) (cont'd).

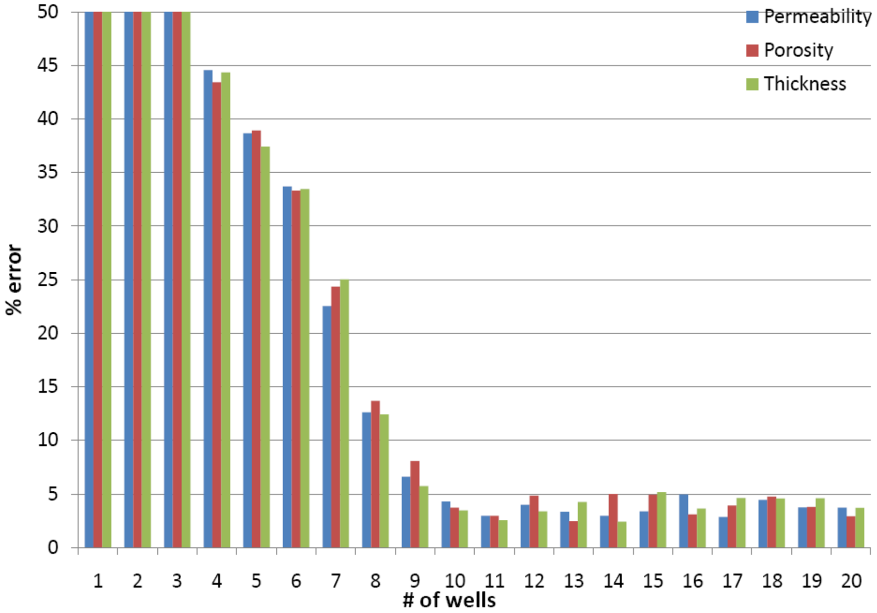


Figure **C-14:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 14 region system) (cont'd).

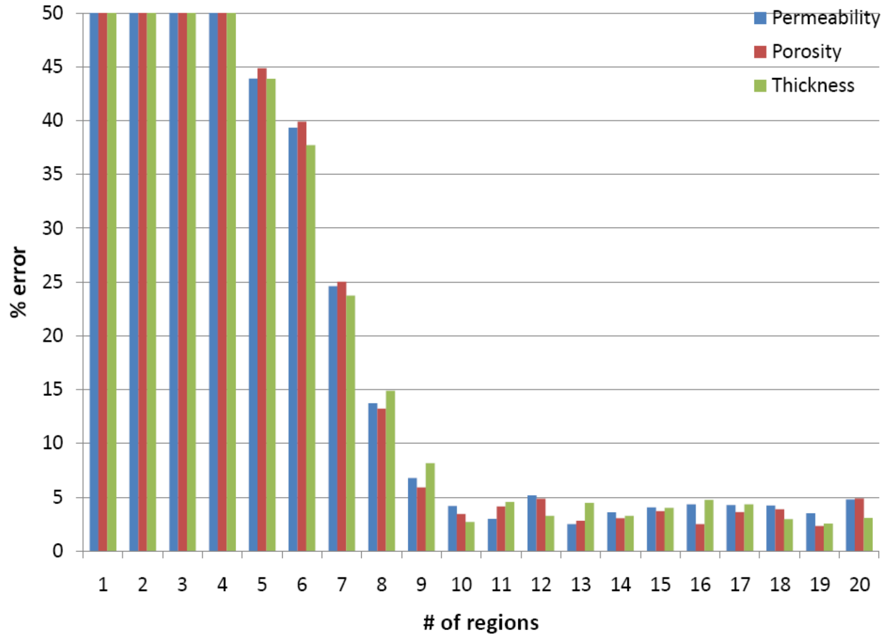


Figure **C-15:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 15 region system) (cont'd).

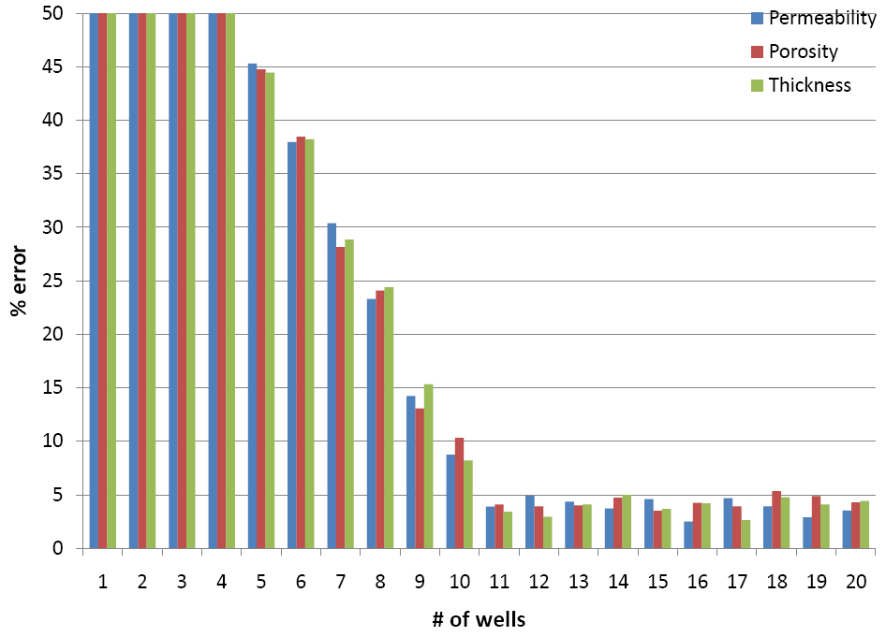


Figure **C-16:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 16 region system) (cont'd).

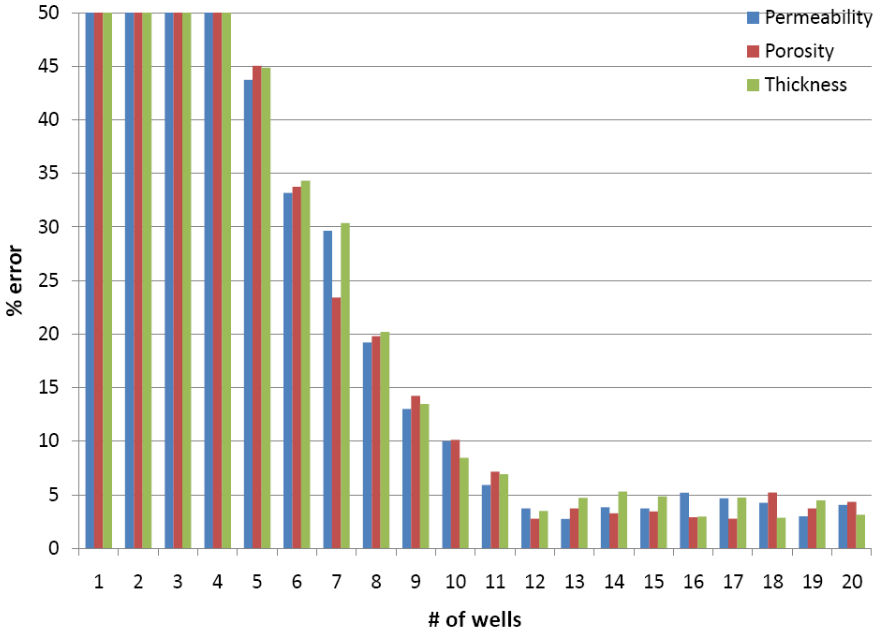


Figure **C-17:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 17 region system) (cont'd).

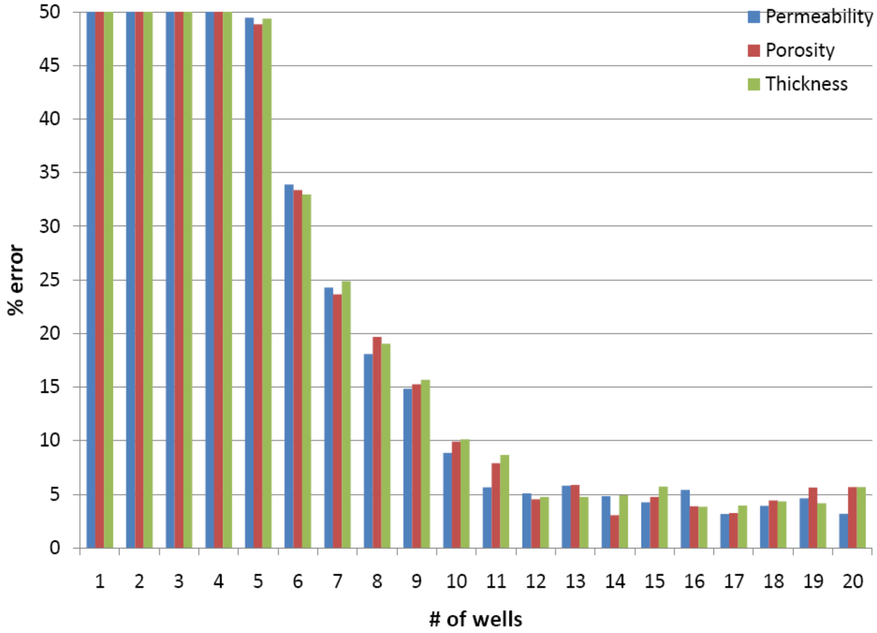


Figure **C-18:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 18 region system) (cont'd).

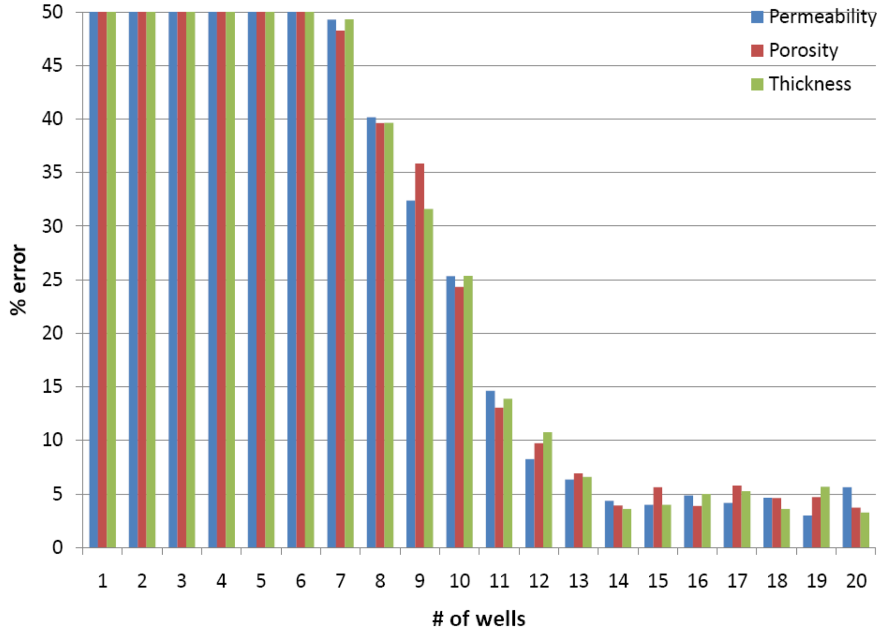


Figure **C-19:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 19 region system) (cont'd).

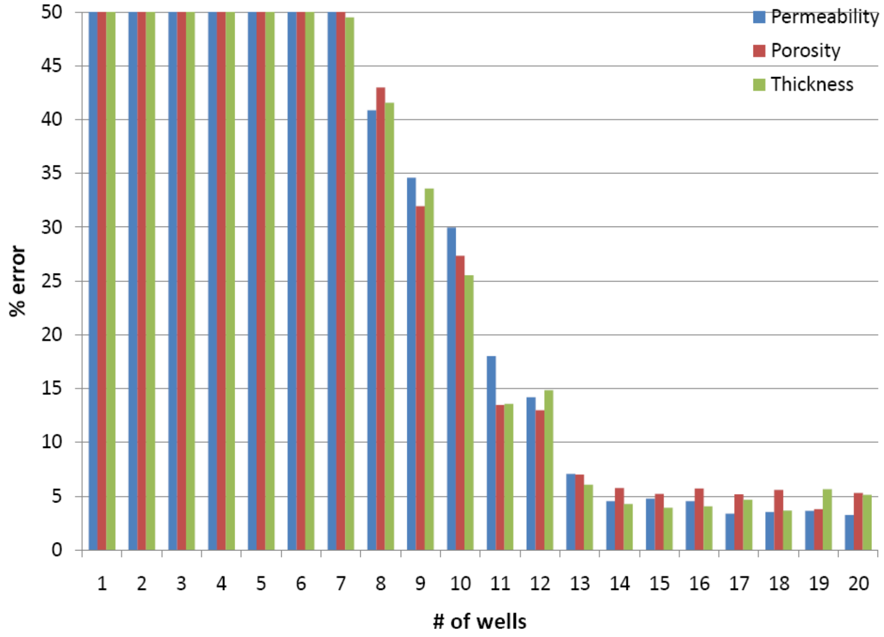


Figure **C-20:** Average prediction errors for each property using network architectures developed in Section **5.3.1.6** (irregular boundary, 20 region system) (cont'd).

Appendix D

Matlab Code

D.1 Training the network designer

```matlab
% Prasanna Chidambaram
% The Pennsylvania State University
% Petroleum and Natural Gas Engineering

% This program is part of the network designer. This will be used to
train the network designer.

clear all;
format compact;

% Loads the pattern/input and target/output data
% Inputs: number of wells, number of regions
% Outputs: number of neurons in I, II and III hidden layers required in
prediction network
load Data\net_train.xls;

net_train_in=net_train(:,1:2);
net_train_out=net_train(:,3:5);

% Transposes the pattern and target data
pt=net_train_in';
tt=net_train_out';

% Process matrices by mapping row minimum and maximum values to [-1 1]
[npt,ps]=mapminmax(pt);
[ntt,ts]=mapminmax(tt);

% Training the network using one hidden layer with 8 neurons
net=newff(minmax(npt),[8,3],{'tansig','purelin'},'trainlm','learngdm','
mse');

% Network parameters
net.trainparam.goal=1e-5;
net.trainparam.epochs=5000;
net.trainparam.show=1;
[net,tr]=train(net,npt,ntt);

% Network weight and bias values saved for use in prediction stage
save files\net_design.mat;
```

D.2 Predicting number of neurons required in prediction network

```
% Prasanna Chidambaram
% The Pennsylvania State University
% Petroleum and Natural Gas Engineering

% This program is part of the network designer. This will be used to
calculate the number of neurons required in the prediction network.

clear all;
format compact;

% Loads the pattern/input data
% Inputs: number of wells, number of regions
% Outputs: number of neurons in I, II and III hidden layers required in
prediction network
load files\net_design.mat;
load data\net_test.xls;

% Transposes the pattern data
test_T=net_test';

% Process matrices by mapping row minimum and maximum values to [-1 1]
n_test_T=mapminmax('apply',test_T,ps)

%Simulate neural network
neuron=sim(net,n_test_T)

% Reverse the processing of matrix to get actual values
neuron=mapminmax('reverse',neuron,ts)

% Storing neuron information calculated by the network designer
f1=fopen('output\neurons.txt','wt');
fprintf(f1,'%10.0f %10.0f %10.0f',neuron);
fprintf(f1,'%10.0f\n',net_test(2)*6+9);
fclose(f1);
```

D.3 Training the prediction network

```
% Prasanna Chidambaram
% The Pennsylvania State University
% Petroleum and Natural Gas Engineering

% This program is part of the prediction network. This will be used to
train the prediction network.

clear all;
format compact;
```

```
% Loads the neuron information calculated by the network designer data
load output\neurons.txt;

% Loads the pattern/input and target/output data
% Inputs: t, Np, Wp, Gp, P, t/365, t/365*Np, WOR, GOR
% Outputs: k, phi, h, k/phi, k*h, A/h, Sorw, Sorg, Swirr, Sgcrit,
krwro, kroirw, krgro, krocritg, N
load data\training.xls;
load data\training_prop.xls;

% Transposes the pattern and target data
pt=training';
tt=training_prop';

% Process matrices by mapping row minimum and maximum values to [-1 1]
[npt,ps]=mapminmax(pt);
[ntt,ts]=mapminmax(tt);

% Training the network using three hidden layers with number of neurons
as
% calculated by the prediction network
net=newff(minmax(npt),[neurons(1),neurons(2),neurons(3),neuron(4)],{'ta
nsig','tansig','tansig','purelin'},'trainlm','learngdm','mse');

% Network parameters
net.trainparam.goal=5e-4;
net.trainparam.epochs=5000;
net.trainparam.show=1;
[net,tr]=train(net,npt,ntt);

% Network weight and bias values saved for use in prediction stage
save files\training.mat;
```

D.4 Testing the prediction network

```
% Prasanna Chidambaram
% The Pennsylvania State University
% Petroleum and Natural Gas Engineering

% This program is part of the prediction network. This will be used to
predict the history match parameters.

clear all;
format compact;

% Loads the pattern/input and target/output data
% Inputs: t, Np, Wp, Gp, P, t/365, t/365*Np, WOR, GOR
% Outputs: k, phi, h, k/phi, k*h, A/h, Sorw, Sorg, Swirr, Sgcrit,
krwro, kroirw, krgro, krocritg, N

load files\training.mat;
load data\testing.xls;
load data\testing_prop.xls;
load data\net_test.xls;

regions=net_test(2);

% Transposes the pattern and target data
testprop_T=testing_prop';
test_T=testing';

% Process matrices by mapping row minimum and maximum values to [-1 1]
n_test_T=mapminmax('apply',test_T,ps);

%Simulate neural network
error3=sim(net,n_test_T);

% Reverse the processing of matrix to get actual values
error3=mapminmax('reverse',error3,ts);

% Storing predicted properties
f1=fopen('files\output.txt','wt');
j=1
for i=1:50;
    for jj=1:regions*6+9;
        k(i,jj)=mean(error3(jj,j:j+rows-1));
    end
    j=j+rows;
end

for i=1:50;
    for jj=1:regions*6+8;
        fprintf(f1,'%10.4f',k(i,jj));
    end
    fprintf(f1,'%10.4f\n',k(i,regions*6+9));
end;
fclose(f1);
```

Appendix E

Reservoir Rock and Fluid Properties Used to Build Perry Reservoir Simulation Models

This section describes the parameters used in the reservoir model that is used generate synthetic data for training and testing of neural network.

E.1 Initialization Data

Table **E-1**: Rock and fluid data used in the initialization of reservoir model.

Type of fluid model	Black oil
Stock tank water density	62.86 lb/ft3
Water salinity	50,000 ppm
Water formation volume factor	1.0273
Water viscosity	0.34 cp
Water compressibility	3.44e^{-006} 1/psi
Rock compressibility	1.695702 e^{-005} 1/psi
Reservoir pressure	3380 psi
Reservoir temperature	198 ^{o}F
Gas gravity	0.796
Oil density	52.03 lb/ft3

E.2 Black Oil PVT Data

Table **E-2**: Black oil PVT data used in the reservoir simulator to generate synthetic data.

P, psi	R_s, (SCF/STB)	B_o, (RB/STB)	B_g, (RB/MSCF)	μ_o, (cp)	μ_g, (cp)
14.70	4.06	1.0669	0.224301	1.5352	0.013086
242.05	39.82	1.0812	0.013355	1.2788	0.013270
469.40	83.54	1.0991	0.006757	1.0805	0.013534
696.76	131.74	1.1194	0.004471	0.9356	0.013848
924.11	183.19	1.1416	0.003315	0.8269	0.014204
1151.46	237.25	1.1654	0.002621	0.7429	0.014597
1378.82	293.52	1.1908	0.002161	0.6761	0.015026
1606.17	351.68	1.2175	0.001836	0.6218	0.015491
1833.52	411.53	1.2456	0.001597	0.5766	0.015992
2060.88	472.91	1.2749	0.001414	0.5385	0.016529
2288.23	535.67	1.3053	0.001271	0.5059	0.017103
2515.59	599.71	1.3369	0.001157	0.4777	0.017715
2742.94	664.93	1.3695	0.001065	0.4529	0.018368
2970.29	731.26	1.4031	0.000989	0.4310	0.019062
3197.65	798.64	1.4377	0.000926	0.4115	0.019799
3425.00	866.99	1.4733	0.000873	0.3941	0.020582
3800.00	945.00	1.4900	0.000700	0.0390	0.021582

VITA

Prasanna Chidambaram

Prasanna Chidambaram was born in Dindigul, India, on January 18th, 1979. In 2000, Prasanna received a bachelor’s degree in Chemical Engineering from University of Madras. After graduation Prasanna came to the United States for graduate studies. Prasanna received a master's degree in Natural Gas Engineering from Texas A&M University-Kingsville, Texas in 2003. Then he joined The Pennsylvania State University for his doctoral studies. During his graduate studies Prasanna has received several awards and recognitions. Prasanna received graduate scholarship from Dotterweich Chair Fund awarded by the College of Engineering, for the academic year 2001-2002, Outstanding Graduate Teaching Assistant in General Education’ Award & Scholarship from Department of Energy & Geo-Environmental Engineering in 2005 and 2007, ‘Teaching Assistant of the year’ Award & Scholarship from Department of Energy & Geo-Environmental Engineering in 2007, ‘George H. K. Schenck Teaching Assistant Award’ from College of Earth and Mineral Sciences in 2008 and Graduate Scholarship from George E. Trimble Chair in Earth and Mineral Sciences in 2009. Prasanna is a member of Society of Petroleum Engineers.

He successfully defended his Ph.D. dissertation on February 2, 2009, and accepted a position with BP America Inc. as reservoir engineer in Houston, Texas.