THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF ENERGY AND MINERAL ENGINEERING

DIAGNOSIS OF RELATIVE PERMEABILITY CHARACTERISTICS FROM RATE TRANSIENT DATA: AN ARTIFICIAL INTELLIGENCE APPROACH

NATHAN CASE
SPRING 2017

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Petroleum and Natural Gas Engineering
with honors in Petroleum and Natural Gas Engineering

Reviewed and approved* by the following:

Turgay Ertekin
Head, John and Willie Leone Family Department of Energy and Mineral Engineering
Professor of Petroleum and Natural Gas Engineering
George E. Trimble Chair in Earth and Mineral Sciences
Thesis Supervisor
Honors Adviser

Zuleima T. Karpyn
Professor of Petroleum and Natural Gas Engineering
Quentin E. and Louise L. Wood Faculty Fellow in Petroleum and Natural Gas Engineering
Faculty Reader

* Signatures are on file in the Schreyer Honors College.

ABSTRACT

The world's dependence on oil and natural gas is evident by the constant coverage they receive in the media regarding their pricing. These two hydrocarbons power the economies of the world despite the recent alternative energy movement. At the heart of hydrocarbon production lies the study of fluid flow. Permeability serves as a critical property to provide information on the rock's ability to transmit fluids, but in the case of multiphase hydrocarbon reservoirs, relative permeability plays a crucial role in predicting and interpreting flow regimes.

Several studies have been conducted to predict relative permeability in reservoirs, and numerous equations, models, and correlations have been developed to provide numerical solutions. One common way to obtain relative permeability data involves obtaining core samples of the reservoir and running saturation tests to provide the permeability data. Coring proves to be an expensive investment for companies, while also posing a high potential for data errors in terms of locality of the core, as core samples only provide a snapshot of the relative permeability data from one specific location in the reservoir, therefore leaving the representation of the relative permeability of the entire reservoir in question. In addition, the removal of the core from the reservoir provides data that fails to represent in situ conditions.

This study seeks to develop representative permeability curves for the entire drainage area of gas-water reservoirs through the use of artificial neural networks (ANNs). The networks were developed using Corey's parameters for the representation of relative permeability characteristics. By generating multiple datasets from reservoir simulations, the network was trained to understand the relationship between the inputs and the outputs (Corey's Model parameters) to develop representative permeability curves based upon rate transient data. The

network performed well in these predictions and the final network produced relative permeability graphs that closely match the relative permeability characteristics utilized in performing the simulation runs.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to begin by thanking my parents Brian and Ann for their love, care, support, and commitment to ensuring my brother and I would receive an education. Also, a thank you to my older brother Jared, for he was a great role model growing up and set a great example of success. Without my family, I would not be in the position I am today.

I would also like to thank Dr. Ertekin for introducing me to artificial neural networks and his guidance in helping to make this research project a possibility. It has been a pleasure and honor to work with Dr. Ertekin and learn from an individual with such great experience and knowledge. I would also like to thank Dr. Karpyn for taking the time to serve as my faculty reader for this project.

In addition, a special thank you to Zhenzihao Zheng for his assistance in teaching me to code Artificial Neural Networks in MATLAB, providing me with a template to generate relative permeability graphs with the extracted data from the neural networks, and for helping me to learn the CMG reservoir modeling software.

Lastly, a thank you to my friends here in State College and to the gang back home. I have thoroughly enjoyed your company and friendships over the years.

Nathan Case

University Park, Pennsylvania

April 2017

Chapter 1

Introduction

Fluid flow dynamics lies at the heart of petroleum and natural gas engineering studies and practices. The flow and ultimately the production of either the oil or natural gas serves as the money-making commodity that drives the industry. Without a proper understanding of the flow within the reservoir, companies fail to properly estimate the recoverable hydrocarbons leading to inaccurate economic forecasts. Such failure proves detrimental to the longevity of the company and its investors.

The flow of fluids is governed by the permeability of the porous media, the rock properties, and the rock's flowability. Darcy's Law developed numerous years ago illustrates the significance of permeability in fluid flow dynamics in porous media. The flow of the fluid is directly proportional to the permeability of the porous medium. When more than one fluid exists within the porous media, the relative permeability becomes of interest for engineers and geologists. Relative permeability is described by the flowability of the respective fluid divided by the absolute permeability of the reservoir.

Relative permeability serves as one of the most significant properties in understanding the flow of the reservoir. Engineers seek to obtain relative permeability data. Rock core samples prove to be one of the most effective methods at obtaining such results. Core analysis comes at a high cost, and often times companies are unable to afford the luxury of obtaining core samples. One problem with coring is that the removal of the core from the reservoir disturbs the sample

and fails to accurately represent the in situ conditions. A second issue with core data is that it represents just one section of the reservoir. As a result, the accuracy of the permeability only applies to that one section of the reservoir. Reservoir and rock property distributions are dynamic and can vary in great magnitude only moving a few feet. Therefore, the need exists to develop representative permeability characteristics representative of the entire reservoir. Such curves are used in reservoir simulation exercises and play an important role in the development of oil and gas fields. Much like coring analysis, reservoir simulation can be both a significant capital expenditure and may require a significant amount of time.

Technological advancements play a critical role in the science and engineering disciplines. One such advancement, is the development of "artificial expert" systems. An Expert system like an artificial neural networks (ANN), which in its simplest form can be explained as an artificial brain that solves complex problems by finding patterns and relationships between various items and parameters where patterns and relationships seem ambiguous or not readily available.

The ultimate goal of this study sought out to develop an "expert" system to predict a representative permeability curve for a reservoir with a given set of certain parameters. The data comes from acceptable, common ranges of values for the pertinent parameters controlling the physical rock properties and the flow of dry, natural gas. CMG IMEX (Version 2015.20)[1] was used to run multiple simulation cases for various, randomly generated reservoirs. MATLAB's[2] Artificial Neural Network toolbox extension was used for the construction of several neural

[1] CMG IMEX: Computer Modeling Group's IMEX reservoir simulator used to develop reservoir simulations to provide data for this study

[2] MATLAB: MATrix LABoratory is a computer software developed by The MathWorks. This software was used to generate all the computer codes required in this study.

network systems with different amounts of neuron layers. These networks were trained with a select amount of the datasets, while the rest of the data was used in the validation of the network and in the blind testing of the expert systems. The system's performance in predicting the relative permeability curves using Corey's Model parameters determined whether or not the system properly recognized the controlling properties and could be used for applications in the field.

This paper is organized into six distinct chapters highlighting the different stages of the research process and the resulting analyses and conclusions from the data. Chapter two provides an overview of pertinent literature pertaining to relative permeability, Corey's Model for Relative Permeability, previous studies using relative permeability results, and an overview of artificial neural networks. Chapter three goes in depth to the problem statement and the workflow of the research process. Chapter four provides an analysis of the results from the neural network predictions. Chapter five highlights the development of the graphical user interface to utilize the neural network for user applications. The final chapter, chapter six discusses the conclusions and the possible applications and further work that could be conducted.

Chapter 2

Literature Review

This chapter focuses on a literature review of the key petroleum and natural gas engineering principles surrounding relative permeability. This includes Darcy's Law and the explanation of both absolute permeability and relative permeability, Corey's Model for relative permeability, and the controlling parameters and properties of the model. Additionally, Artificial Neural Networks (ANNs) and case studies which have used ANNs for relative permeability applications will also be discussed.

2.1 Introduction

From a reservoir engineer's perspective, the formal definition for permeability states that permeability is a rock property measuring the ability of the medium to transmit fluids when a reservoir rock is 100% saturated with a given fluid (Dandekar 2006). Permeability applies to the flow of water in porous medias, and in the case of hydrocarbons, the flow of hydrocarbon through a porous rock medium. In essence, permeability lies at the heart of understanding petroleum engineering (Dandekar 2006).

2.2 Permeability and Relative Permeability

2.2.1 Darcy's Law

In 1856, the French hydraulic engineer Henry Darcy sought out to understand the flow of groundwater (Fetter 1994). He then published an equation, which explains that "the fluid flow through a porous media is proportional to the product of the cross sectional area through which flow can occur, or the hydraulic conductivity" (Fetter 1994). Darcy replaced the term

hydraulic conductivity with what is today identified as permeability. Permeability is a rock property illustrating its ability to transmit flow (Dandekar 2006). The formal equation can be written in different formats; however, one common way Darcy's Law is illustrated in Equation 1.1 (Fetter 1994).

$$Q = -KA(\frac{dh}{dl}) \quad \textbf{Eq. 1.1}$$

This equation solves for the volume flow or flux through a given porous media.

where: K = hydraulic conductivity (m/day)

A = area (m^2)

dh/dl = head loss per given length (m/m)

The hydraulic conductivity is dependent upon the specific or absolute permeability, the specific weight of the flowing fluid and the viscosity of the flowing fluid (Bengtson 2011). The specific permeability is a property specific to the porous medium (Bengston 2011). This illustrates the importance of understanding the characteristics of reservoir rock, specifically permeability in the determination of fluid flow.

Hydrocarbon extraction is another example of fluid flow, therefore Darcy's Law lies at the heart of numerous petroleum and natural engineering principles, applications, and equations. However, the derivation of Darcy's Law is applied to a one-phase system of just water. Hydrocarbon reservoirs exhibit the behavior of either 2-phase (gas-water, oil-gas or oil-water) or 3-phase (gas-oil-water) systems. Therefore, the hydraulic conductivity or permeability no longer remains dependent upon just a single fluid in the porous media.

2.2.2 Relative Permeability

In the case of a system where more than one fluid or one phase resides within the porous medium, absolute permeability fails to effectively describe the flow of fluids. Instead, each phase

will have its own fluid flow equation, which relies upon that particular phase's relative permeability. According to Dandekar, relative permeability is defined as "the effective permeability of a specific fluid phase divided by the absolute permeability of the medium through which the fluid flows" (2006). Therefore, each fluid phase has a specific relative permeability value. Relative permeability is a function of the saturation of the phase. A specific value of relative permeability is only applicable at a certain saturation (Relative Permeability 2016). As the saturation of the phase increases, so too will its relative permeability. This increase in relative permeability and saturation of one phase will correlate to a decrease in the saturations and relative permeability of the other phases present in the porous medium. This relationship is illustrated in Figure 1.

2.2.3 Relative Permeability Charts

Charts are typically used to display relative permeability characteristics. Figure 1 provides an example of a relative permeability curve for a gas-water system.

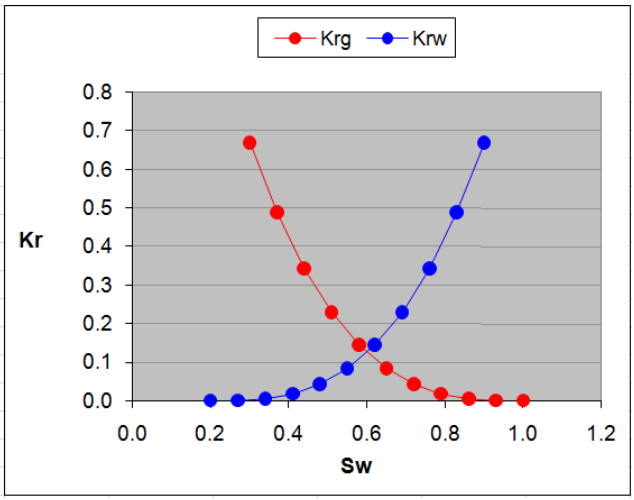


Figure 1. Example of a relative permeability curve for a gas-water system (Effendi 2011).

The endpoints of these curves denote important, controlling saturation values. The k_{rw} curve will exhibit zero values up until the critical water saturation (Effendi 2011). This is the water saturation value where water becomes mobile. The lowest saturation value provided in the data is the connate water saturation and this serves as the left endpoint of the water relative permeability curve. The right end point of the water relative permeability curve indicates the relative permeability of water at the residual gas saturation (Petrofaq 2013). For the gas relative permeability curve, the left endpoint indicates relative permeability of gas at the critical water saturation, and the right endpoint is determined by one minus the critical gas saturation (Petrofaq 2013). The critical gas saturation is where the gas becomes mobile within the reservoir (Dandekar 2006).

A relative permeability chart provides engineers with valuable information regarding the reservoir's flowability. One of these pieces of information pertains to the wettability of the reservoir system. Wettability is defined as "the preference of solid to contact one liquid or gas, known as the wetting phase, rather than the other" (Schlumberger 2017). In a water-wet system the water will tend to spread out over the surface of the reservoir rock leaving the gas or oil unattached to the rock grains. In a hydrocarbon-wet system, the gas or oil will tend to spread out over the rock grains and stick to the rock leaving the water free to flow between the rock grains. Water-wet systems prove to provide better production capabilities than do oil or gas-wet systems. In the case of a hydrocarbon-wet reservoir, engineers try to take measures to reverse the wettability of the system. Wettability can be identified by the intersection point of the relative permeability curve for water and the relative permeability curve for gas. A hydrocarbon-wet system exhibits an intersection point prior to 50% water saturation on the x-axis, while a water-wet system exhibits an intersection point past 50% water saturation. Systems with an intersection

point exactly at the 50% water saturation point are neither water-wet nor hydrocarbon-wet. Instead, these systems are known as intermediate-wet, with the reservoir rock not exhibiting any preference for either of the fluids (Schlumberger 2017). Other properties that can be obtained from a relative permeability curve include the aforementioned endpoint saturations. This includes connate water saturation, critical gas or critical oil saturation, critical water saturation, saturation of gas or oil at critical water saturation, etc. (Petrofaq 2013).

2.3 Corey's Model for Relative Permeability

In 1954, A.T Corey sought out to discover the relationship between gas and oil relative permeabilities using a bundle of capillary pressure tubes. This type of capillary pressure method experiment was first done by Gates and Lietz, Corey based the groundwork of his experiments on the work of several other individuals prior to his research (Corey 1954). Corey's "equipment employed [was] similar to that described by Gates and Lietz" and his measurement procedure for gas relative permeability was similar to that of Osoba et al" (Corey 1945). A benefit of Corey's method was the reproducibility and ease in determining the gas relative permeability, which then enabled him to determine the relative permeability of the oil. He plotted values of oil relative permeability as a function of oil saturation and then compared these to measured values from the experiments (Corey 1945). In his publication, Corey stated that "with the use of such a plot complete oil relative permeability curves [could] be obtained from gas relative permeability data in a few minutes" (1954). The oil relative permeability curve was obtained via Equation 1.2 (Corey 1954).

$$k_{ro} = \left(\frac{S_o - S_{or}}{1 - S_{or}}\right)^4 \quad \textbf{Eq. 1.2}$$

where: S_o = saturation of oil
S_{or} = residual oil saturation

In 1964, Corey collaborated with R.H Brooks at Colorado State University to effectively expand Corey's earlier findings to other systems aside from the original oil-gas system he first developed in 1954 (1964). The bases of their experiment incorporated a lambda value into the exponent provided in Equation 1.2. Different values of lambda were applied for different orientations of rock and pore size distribution (Brooks and Corey 1964). In the case where lambda equaled 2, the original expression (Eq.1.2) for the relative permeability of oil applied (Brooks and Corey 1964). Their findings resulted in equations that described the relative permeability of water, oil and gas for all systems. The equations and the corresponding nomenclature for the gas-water system is described below.

$$k_{rw} = k_{rwiro}\left(\frac{S_w - S_{wcrit}}{1 - S_{gcrit}}\right)^{n_w} \quad \textbf{Eq. 1. 3}$$

where: k_{rwiro} = relative permeability of water at 100% water saturation
S_w = water saturation
S_{wcrit} = critical water saturation
S_{gcrit} = critical gas saturation
n_w = Corey's exponent for water

$$k_{rg} = k_{rgcl}\left(\frac{S_g - S_{gcrit}}{1 - S_{gcrit} - S_{wcon}}\right)^{n_g} \quad \textbf{Eq. 1.4}$$

where: k_{rgcl} = relative permeability of gas at connate liquid
S_g = gas saturation
S_{gcrit} = critical gas saturation
S_{wcon} = connate water saturation
n_g = Corey's exponent for gas

Equation 1.3 and Equation 1.4 serve as two of the equations developed from Brooks' and Corey's experiments (1964). These two equations served as the relative permeability model used to generate relative permeability data. The data was then used to construct representative relative permeability charts for each reservoir model.

2.4 Overview of Artificial Neural Networks

2.4.1 Historical Background

The study of the human brain has been around for several millennia. However, the development of artificial neurological pathways came to fruition during the same time as the onset of the modern electronics era (Reingold). In 1943, neurophysiologist Warren McCulloch and mathematician Walter Pitts wrote a paper describing their thoughts on how the neurons of the brain might work (Reingold). They constructed a model of their thoughts using electrical circuits. A short time later, work done by Frank Rosenblatt and two Stanford professors, Bernard Woodrow and Marcian Hoff resulted in the development of the first operational neural networks (Reingold; Cladbaugh, Myszewski, & Pang). Rosenblatt established a single-layered network named *perceptron,* while Woodrow and Hoff developed a multi-layered network called MADELINE that today, still remains in commercial use (Reingold; Cladbaugh, Myszewski, & Pang). Both of these works led to the realization that neural networks could be used for pattern recognition. Between the 1960's and 1980's a lack of further work and research involving artificial neural networks persisted in large part to the failure of computer processing to evolve at a fast enough pace to handle the complexity of the networks. In addition, fears of thinking machines surfaced and their potential impacts began to raise philosophical questions (Cladbaugh, Myszewski, & Pang).

The year of 1982 saw the reemergence of neural networks and their capabilities after John Hopfield presented a paper to the National Academy of Science explaining the potential functionality and application of such networks (Reingold). New technological developments, in particular the back propagation function allowed for the development of multi-layered networks

to become more applicable. By 1985, the American Institute for Physics held its first annual Neural Networks for Computing Meeting marking the permanent presence of neural networks in the scientific community. Today, neural networks find use in several different arenas and applications. This includes the electronics industry, defense intelligence, financial and banking applications, medical research, the oil and gas industry etc.

2.4.2 Model of the Artificial Neural Network

Much like the name Artificial Neural Network (ANN) suggests, the basis of the network mimics the functionality of the neurons of the human brain, but on a smaller scale. The neurons of the human brain are the foundation of the nervous system. The neurons receive information via electrical signals through the dendrites. This information is processed by the soma, and is then passed to the axon. Much like the human brain, the core of the ANN structure lies in the interconnected neuron layers. Each neuron obtains an input, generates a sum, takes the sum and performs some sort of function, and then outputs the new results to the next neuron or layer (Almousa 2013). Each connection is associated with a numeric value called a weight. Figure 2 and Figure 3 below, illustrate the structural make-up of both the human neuron and the neuron of an artificial neural network.

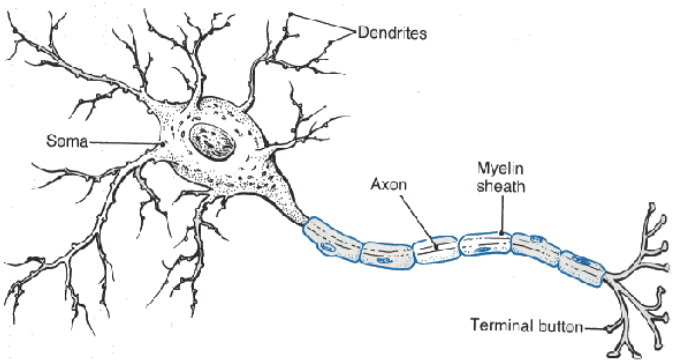


Figure 2. Model of human neuron (Carlson 1992).

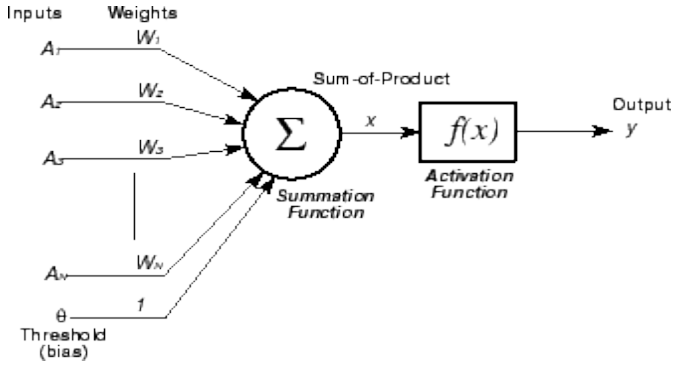


Figure 3. Model of artificial neuron (Kawaguchi 2000).

The summation function that occurs within the artificial neuron (Fig 3) is described below in Equation 1.5 (Suzuki 2011).

$$y_i(k) = F[\sum_{i=0}^{m} w_i(k) \cdot x_i(k) + b] \quad \textbf{Eq 1. 5}$$

where:

x_i = input value
w_i = weight value
k = discrete time
F = transfer function
b = bias
y_i = output

2.4.3 Artificial Neural Network Structures

For almost all complex problems or in pattern recognition, a single neuron will not suffice. As is the case with the human brain where several neurons are linked together, so too several linked artificial neurons are required to make up the structure of an artificial neural network. The neurons can be arranged into layers to enhance the processing capability of the ANN. An ANN can either have a single layer of neurons or two or more layers of neurons. These networks with more than one neuron layer are considered multilayer networks and the layers are termed hidden layers. Figure 4 illustrates the difference in the framework of a single-layer

network and a multilayer network. Typically, a two or three-layer network has the capability to solve more complex problems or recognize more complex patterns than a single-layer network (Beale, Hagan, & Demuth 2014). While some researchers develop rules or guidelines to establish when constructing an artificial network, the rules often do not hold true from problem to problem (Beale, Hagan, & Demuth 2014). Therefore, in establishing the best network for one's own problem, the best approach includes the trial of multiple different network structures.

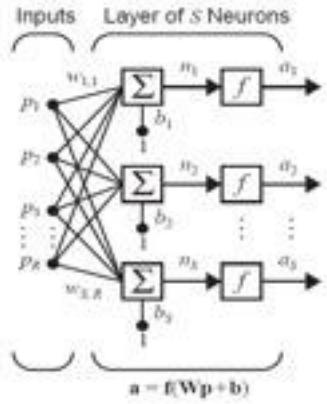


Figure 5. Model of single layer artificial neural network structure (Beale, Hagan, & Demuth 2014).

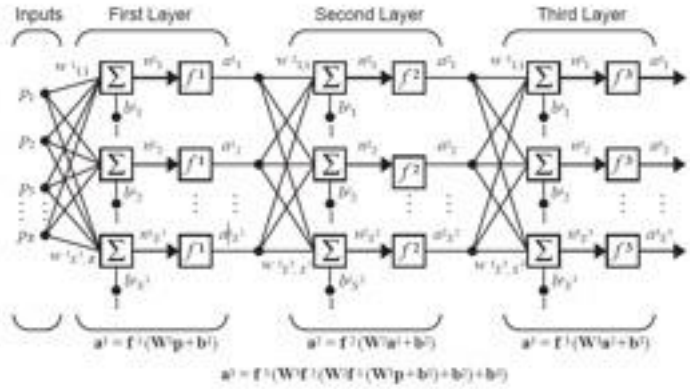


Figure 4. Model of multi-layer artificial neural network structure (Beale, Hagan, & Demuth 2014).

Regardless of how many hidden the layers the network contains the final piece of the network is the output layer. Like the inputs, the outputs are known values. The successful ANN will learn the functional relationship between the input layer and the output layer. Once the network learns the relationship, the network possesses the capability of predicting future outputs from a given set of inputs.

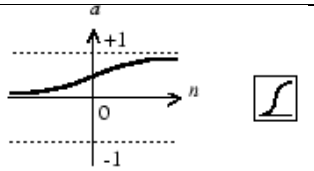
2.4.4 Learning, Transfer, and Training Functions

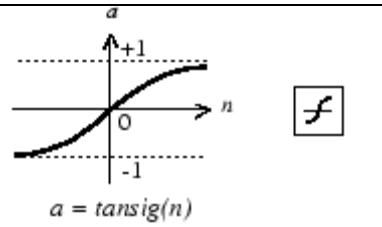
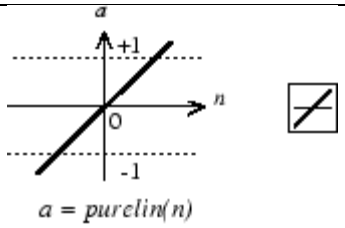
Within the ANN there are three main types of functions that are specified. These include learning functions, transfer functions, and training functions. There are several different types of

each these three main functions, which carry out their processes in various ways. Learning takes place during the training of the network when it is attempting to identify the patterns between the inputs. The learning process is dependent upon the type of algorithm used by the learning function. Each algorithm differs in the way it updates the aforementioned synaptic weights and biases during each epoch (iteration). The weights continue to update until the error specification between the input and output is reached (Guler, Ertekin, & Grader 2003). MATLAB's Neural Network Toolbox offers eleven learning function (The MathWorks Inc. 2017) Some of the common learning functions include gradient descent weight and bias learning function (*learngd*) and gradient descent with momentum weight and bias learning function (*learngdm*).

Once "learning" takes place, the next function utilized within the network is the transfer function. This training function prepares the information for the next layer. They operate by calculating a layer of neuron's output from the net input then send this information along to the next layer of neurons. Transfer functions typically operate with values between negative one and positive one, which requires the data to be normalized prior to being input into the network. Common transfer functions include the *log-sigmoid*, the *tan-sigmoid*, and the *linear* transfer functions. Table 1 below summarize the key components to these transfer functions.

Table 1. Three commonly used transfer functions within ANNs and their key features.

Function (MATLAB code)	Range of Values	Graphical Example
Log-Sigmoid (*logsig*)	[0,1]	 *a = logsig(n)* Log-Sigmoid Transfer Function

Tan-Sigmoid (*tansig*)	[-1,1]	 Tan-Sigmoid Transfer Function
Linear (*purelin*)	[-1,1]	 Linear Transfer Function

The last of the three main functions within an ANN structure are the training functions. These functions are the overarching, controlling factor in the workflow of the network. There are two main different types of training. Incremental training where the weights and biases of the network are updated each time an input is presented to the network and batch training where the weights and biases are only updated after all the inputs are presented. This study will utilize batch style training primarily because they are more efficient in both MATLAB and within MATLAB's Neural Network Toolbox.

Each training function has a number of different parameters which must be specified during the function's calling. These parameters include the number of epochs, the performance goal, the maximum number of validation failures, the number of epochs between displays, the maximum time for training, and whether or not the user wants to generate command line output. These different, specified parameters help lead a user to which mode or type of training to utilize. This is because each training function will update the weights and biases in a different

way depending on the functional algorithm's rules for optimization (Almousa 2013). Two examples of how functions differ from one another is how they present the input to the neurons and at what point within the training the weights and biases update.

Depending upon the goal of the network, different training functions see various uses. A few such examples of training functions include the conjugate gradient backpropagation with Polak-Ribiere updates, the random order incremental training with learning functions, and the Levenburg-Marquardt backpropagation. The conjugate gradient function and the Levenburg-Marquardt both offer the backpropagation technique allows for the network to essentially go backwards during the algorithms. The neuron layers have connections both in the forwards and backwards direction. The Levenburg-Marquardt (*trainlm*) technique operates in accordance with the Levenburg-Marquardt optimization technique. The conjugate gradient backpropagation training function (*traincgp*) requires that the weight, the net input, and the transfer functions have a derivative function. The backpropagation is carried out in calculating the derivatives of performance (The MathWorks 2017). The random order incremental training with learning functions (*trainr*) operates by presenting the input to the neurons and then incrementally updates the weights and biases after each presentation of input (The MathWorks 2017).

2.5 Applications of Artificial Neural Networks in the Oil and Gas Industry

Artificial Neural Networks (ANNs) have become an attractive tool for solving problems in the oil and gas industry. ANNs prove to be helpful in solving complex, non-linear problems in the absence of enough data (Alghazal 2015). There are multiple pieces of literature which span across various aspects of petroleum engineering including reservoir simulation, history matching, Enhanced Oil Recovery, polymer injection, and well testing. In addition, previous studies utilizing ANNs have proven the capability of these networks to predict relative permeability

values. The following provides discussion on ANN studies surrounding relative permeability, which is the main focus of this study. Al-Fattah and Al-Naim (2009) showed that water-oil relative permeability data could be predicted using water-flooded core samples from the carbonate reservoirs of the Saudi Arabian oil fields. The study provided a generalized regression neural network algorithm to create an ANN capable of predicting the relative permeability values of the complex carbonate reservoirs. The network outperformed the results provided by empirical equations and correlations (Al-Fattah and Al-Naim 2009). Al-Fattah and Al-Naim (2009) had a multitude of input variables including various saturation values obtained from core samples, well location, wettability, and functional links between different saturation.

Guler, Ertekin, and Grader (2003) developed an ANN that also predicted oil-water permeability data. However, the inputs for the model were from experimental data sets found in literature. Unlike Al-Fattah and Al-Naim (2009) where the data only came from the carbonate Suadi Arabian oilfields, Guler, Ertekin and Grader (2003) took data from various locations around the world allowing for a more universally applicable ANN. The inputs included permeability data from the experiments as well as various rock and fluid properties associated with the reservoirs. The study found that increased accuracy occurred by utilizing more property-based data, utilizing normalized saturations, and developing a network model for each phase (Guler, Ertekin, Grader 2003).

Silpngarlmers et al (2002) provided an ANN model for both oil-water and gas-oil relative permeability data. Much like the previously mentioned study, this study also used rock and fluid properties as input for both the oil-water model and the gas-oil model. In addition, the end point saturations were used as input parameters. The two different models differed in the mathematical functional links used between relevant properties in the particular two phase reservoirs

(Silpngarlmers et al 2002). The results of the ANNs highlighted that the controlling factors of flow differ from liquid-liquid systems to liquid-gas systems and also showed that utilizing normalized saturations improved the network performance. The work of all three studies provides a basis that ANN technology can accurately provide if not improve the accuracy of relative permeability data compared to correlations, while limiting the costs and time associated with today's most widely used method of obtaining relative permeability data, core samples.

Chapter 3

Problem Statement

Relative permeability lies at the heart of nearly all reservoir flow calculations in multi-phase systems. Therefore, its determination is crucial to obtaining accurate production forecasting models, simulations, and history matching. The most common way to obtain relative permeability data is through core samples, which is an expensive, time-consuming, and complex process. Even "successful" coring data provides a limited outlook of the reservoir by capturing permeability at just a small fraction of the total reservoir. Regardless of the experimental accuracy in the lab, core samples fail to accurately capture the data of in situ conditions within the reservoirs. In addition, smaller operating companies may not have the necessary capital to obtain multiple core samples, leaving them without an accurate representation of one of the most important rock-fluid properties, relative permeability.

The existing classical theory has failed to provide a relationship between relative permeability and production data. Furthermore, the mathematical models that have been created without production data often are not universally applicable or provide erroneous data due to the existence of the non-linear controlling variables of relative permeability. Chapter 2 illustrated the capability of artificial neural networks (ANNs) to quickly provide solutions to complex, non-linear problems by learning the functional relationship between the input and output variables. ANNs also contain the dynamic ability to be updated as more data becomes available. Therefore, this study sets out to establish an ANN model capable of producing representative permeability curves based upon Corey's Relative Permeability Model for the entire drainage area of gas-water

reservoir systems. The following steps below provide a summarized workflow of the systematic process used to develop the reservoir models and the subsequent ANN models.

- Compile pertinent rock-fluid and reservoir properties into acceptable ranges and then randomly generate this data into matrices using MATLAB

-Input random data into a commercial model software to develop natural gas reservoir models and validate the data set prior to running the simulations (Figure 6 and Figure 7 provide an example of a reservoir model used in simulations).

- Obtain production data for the first month in small time increments and the following 23 months at larger time increments

- Compile the input and output matrices for the artificial neural networks

- Develop and feed the ANN with input and output to start training the network

-Compare results from network with those of the simulation runs to determine if changes must be made within the network structure or functional links between inputs should be included

-Move on to a new stage of network development if Corey's model parameters were not predicted within a $\pm$10% error.

-Finalize the network structure of the "best" ANN then produce relative permeability charts to analyze the performance of the network

Chapter 4

Natural Gas Reservoir Development and Modeling

Chapter noted that CMG IMEX was used in this study to build the reservoir model and generate the simulation results which would later be used as input parameters for the neural network. The development of the model and the methods used to generate the inputs for the simulation are discussed throughout this chapter. This includes the determination of the acceptable ranges for the reservoir properties and rock fluid properties, the determination of the controlling processes in the reservoir, and the project design specifications. MATLAB computer coding was used extensively in preparing the input matrices, developing the input parameters based on reservoir engineering equations, and in validating data sets.

4.1 Reservoir Description

Each reservoir generated in reservoir modeling software was a three dimensional radial reservoir with dual permeability. For each model, the reservoir contained one uniform section of thickness. Figure 6 and Figure 7 provide illustrations of an example reservoir constructed in reservoir modeling software in both two and three dimensions. Each reservoir contained a range of other input properties to encompass the various ranges encountered in gas-water reservoirs around the world. In finding an acceptable range of data for different fields and formations around the world, the dynamic nature of the models allowed the neural network to develop a universally applicable solution to the problem rather than just limiting the reservoir models to a specific region (i.e. the Marcellus Shale).

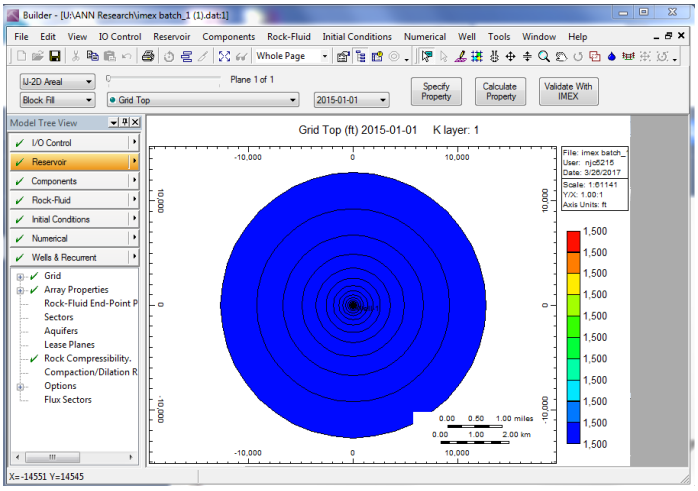


Figure 6. 2-D CMG Builder reservoir model for case 1, which makes the radial discritizations visible

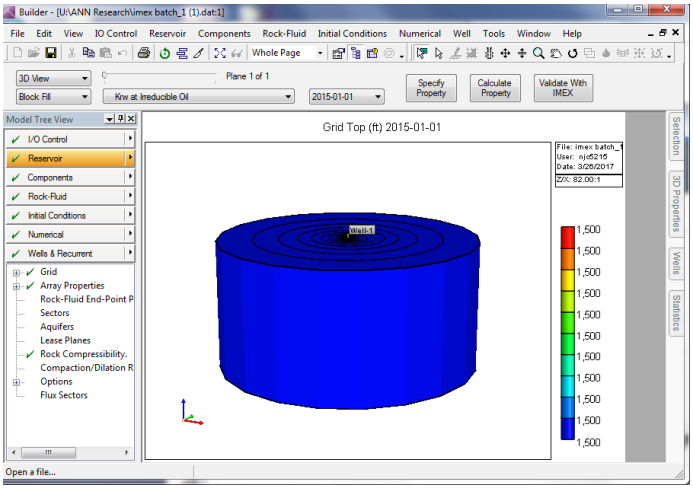


Figure 7. 3-D CMG Builder reservoir model constructed for Case 1

In order to arrive at an "acceptable" range, a multitude of different resources were consulted, including well logs, books, reservoir data sets, and websites regarding information from various hydrocarbon fields across the world. The ranges of the properties used for the model construction of the reservoirs can be found in Table 2.

Table 2. Range of values used for parameters in reservoir model construction.

Range of Reservoir Properties			
Property	**Min**	**Max**	**Unit**
Reservoir Radius (r)	10	5000	ft.
Reservoir Thickness (h)	20	500	ft.
Porosity (Φ)	0.02	0.4	fraction
Initial Pressure (P)	1000	6400	psia
Reservoir Temperature (T)	100	300	degrees Fahrenheit
vertical permeability (k_v)	0.1	10	mD
horizontal permeability (k_h)	0.1	1500	mD
Gas Specific Gravity	0.55	0.75	fraction
rock compressibility (c_f)	0.0000001	0.000025	1/psia
well bore radius (r_w)	0.25	0.25	ft.

The same approach was taken in the generation of data for Corey's Model parameters. Extensive background research was conducted in referencing several previously drilled natural gas wells for saturation and permeability data. Literature was also referenced to find the acceptable ranges for the controlling variables within Corey's relative permeability model. Table 3 provides the variables and the ranges used in this study

Table 3. The range of values used for the variables within Corey's Relative Permeability Model

Corey's Model Parameters		
Property	**Min**	**Max**
Initial Water Saturation (S_{wi})	0.4	0.8
Connate Water Saturation (S_{wcon})	0.05	0.2
Critical Water Saturation (S_{wcrit})	0.05	0.4
Initial Gas Saturation (S_{gi})	0.2	0.6
Connate Gas Saturation (S_{gcon})	0.01	0.2
Critical Gas Saturation (S_{gcrit})	0.01	0.2
Relative Permeability water at 100% water (k_{wiro})	1	1
Relative Permeability gas at connate liquid (k_{rgcl})	1	1
Corey Exponent for Gas (n_g)	2	4
Corey Exponent for Water (n_w)	2	6

4.2 Important Design Specifications

4.2.1 Radial Discretization

In designing the reservoir, two typical for the well model include the use of Cartesian coordinates in which a block is constructed, or as a radial model using cylindrical grid coordinates. This study chose to utilize a cylindrical model, because such models typically provide a more accurate representation of a single-well reservoir system. This required radially discretizing the reservoir in order to properly capture the pressure propagations with the discretizations close together near the wellbore and spreading out moving away from the wellbore. The pressure drop is most pronounced nearest to the wellbore creating the need for the radial discretizations to be closer near the wellbore and then spreading out approaching the reservoir boundary. This study utilized the procedure and equations outlined in *Petroleum Reservoir Simulation – A Basic Approach* to radially discretize the reservoir (Abou-Kassem,

Farouq Ali, Islam 2006). Equations 4.1,4.2, and 4.3 below highlight the radial discretization process discussed in the aforementioned book.

$$\alpha_{lg} = (\frac{r_e}{r_w})^{\frac{1}{n_r}} \quad \textbf{Eq. 4.1}$$

$$r_1 = r_w[\alpha_{lg} \ln\left(\frac{\alpha_{lg}}{\alpha_{lg}-1}\right)] \quad \textbf{Eq. 4.2}$$

$$r_i = r_1(\alpha_{lg}^{i-1}) \quad \textbf{Eq. 4.3}$$

where: α_{lg}= logarithmic spacing constant (dimensionless)
r_e= reservoir radius (ft.)
r_w= wellbore radius (ft.)
n_r= number of discretizations
r_1= radial constant (ft.)
r_i= radius of i-th discretization (ft.)

4.2.2 PVT Properties

As previously mentioned, it is important to accurately capture the pressure changes in the reservoir, as the pressure serves as a mechanism to drive the production of the wellbore fluids. Therefore, an important consideration in the development of the reservoir models included the specification the pressure, temperature, and volume (PVT) properties. PVT data plays a key role in determining reserve estimation, developing a production recovery plan, and identifying the quality of the reservoir fluids. Chemistry and thermodynamics provides us with the ideal gas law of PV = nRT. However, this gas law gets its name because it makes assumptions that no gas or fluid actually obeys. The ideal gas law provides its best results at high temperatures and low pressures. Because reservoirs often have varying temperatures and high pressures, a different calculation is required to provide accurate pressure data within the reservoir.

4.2.3 Gas Compressibility & Gas Expansion Factors

The answer to this problem lies with the compressibility factor, or more simply, the Z-factor. The gas compressibility factor proves to be pivotal in determining the values of other gas properties, which are important parameters within the reservoir. The gas compressibility factor can be defined as the volume of gas or liquid at actual conditions divided by the volume of gas or liquid at ideal conditions (Dandekar 2006). The Z-factor can be added to the ideal gas law, which is shown below in Equation 4.4.

$PV = ZnRT$ ……..…..**Eq. 4.4**

where: P = pressure ($psia$)
V = volume (ft^3)
Z = gas compressibility factor
n = number of moles
R = gas constant = 10.73 $\frac{\text{psia fi\^3}}{\text{lb mol °R}}$
T = reservoir temperature ($°R$)

The Z-factor can be calculated from a multitude of different equations of states or by different correlations. For natural gas systems, a widely accepted method is the Dranchuk and Abou-Kassem Z-Factor correlation (1975). The correlation expands upon Dranchuk's early work and that of Starling-Carnahan. The correlation provides better results than the Hall-Yarborough correlation, which was fitted to the Standing-Katz Z-factor chart (Dranchuk and Abou-Kassem 1975). It also provides accurate results with reduced pressure values that extend beyond 20 unlike the Hanikson et al correlation which has limitations in this regard (Dranchuk and Abou-Kassem 1975). There are several equations that make up the Dranchuk and Abou-Kassem correlation, which are outlined below.

Pseudo Critical Properties

$$T_{pc}(°R) = 168 + 325\gamma_g - 12.5\gamma_g^2 \quad \textbf{Eq. 4.5}$$

$$P_{pc}(psia) = 677 + 15\gamma_g - 37.5\gamma_g^2 \quad \textbf{Eq. 4.6}$$

Pseudo Reduced Properties

$$T_{pr} = \frac{T\ (°R)}{T_{pc}\ (°R)} \quad \textbf{Eq. 4.7}$$

$$P_{pr} = \frac{P\ (psia)}{P_{pc}\ (psia)} \quad \textbf{Eq. 4.8}$$

Reduced Gas Density

$$\rho_r = \frac{0.27P_{pr}}{ZT_{pr}} \quad \textbf{Eq. 4.9}$$

where: T_{pc} = pseudo critical temperature ($°R$)
P_{pc}= pseudo critical pressure ($psia$)
γ_g = gas specific gravity
T_{pr} = pseudo reduced temperature
P_{pr} = pseudo reduced pressure
T = reservoir temperature ($°R$)
P = reservoir pressure ($psia$)
ρ_r= reduced gas density
Z = gas compressibility factor

With this information the Z-factor can be obtained by the following equation

$$Z = 1 + \rho_r\left[A_1 + \frac{A_2}{T_{pr}} + \frac{A_3}{T_{pr}^3} + \frac{A_4}{T_{pr}^4} + \frac{A_5}{T_{pr}^5}\right] + \rho_r^2\left[A_6 + \frac{A_7}{T_{pr}} + \frac{A_8}{T_{pr}^2}\right] - A_9\rho_r^5\left[\frac{A_7}{T_{pr}} + \frac{A_8}{T_{pr}^2}\right] + \left[A_{10}(1 + A_{11}\rho_r^2)\left(\frac{\rho_r^2}{T_{pr}^3}\right)\left(e^{[A_{11}\rho_r^2]}\right)\right] \quad \textbf{Eq. 4.10}$$

where: Z = gas compressibility factor
ρ_r = reduced gas density
T_{pr} = pseudo reduced temperature
A_1=0.3265
A_2=-1.07
A_3=-0.5339
A_4=0.01569
A_5=-0.05165
A_6=0.5475

A_7=-0.7361
A_8=0.1844
A_9=0.1056
A_{1o}=0.6134
A_{11}=0.7210

This correlation contains two equations that are dependent upon each other, the Z-factor equation contains reduced gas density and the reduced gas density equation is dependent upon the Z-factor. Therefore, the Z-factor was calculated at each pressure value for each reservoir using a function in MATLAB, which allowed for the simultaneous solving for the two dependent properties. The function used for this process can be found in Appendix A. The Z-factor becomes a necessary component in not only calculating PVT properties, but also in specifying the gas formation volume factor (B_g) or its reciprocal, the gas expansion factor (E_g). For the purpose of this study the gas expansion factor (Equation 4.11) was used in the construction of the reservoir model.

$$E_g\left(\frac{scf}{bbl}\right) = 198.6\left(\frac{P(psia)}{ZT(°R)}\right) \quad \textbf{Eq. 4.11}$$

4.2.4 Gas Viscosity

The next property that required special attention in the construction of the reservoir was the gas viscosity. This property provides a measure of a substance's resistance to flow. According to Dandekar (2009), despite viscosity's dependence upon pressure, temperature, fluid density, and the composition of the fluid, there is not one distinct, well-defined equation that can relate all of the properties into one equation. Therefore, correlations and estimates have been developed much like the Dranchuk Abou-Kassem correlation for the Z-factor. In this study, the Lee-Gonzalez method (1966) for determining the gas viscosity of natural gas was used to determine the gas viscosity for each reservoir model. Lee and Gonzalez (1966) developed their correlation after compiling measurements of four pure hydrocarbon and hydrocarbon mixtures

for temperatures ranging from 100-340 degrees Fahrenheit and for pressure ranging from 100-8,000 psia. These ranges fit into the scope of this study making the results applicable for the reservoir models generated to provide data for the development of the artificial neural networks. Equations 4.12-4.16 provide the ordered process for arriving at the gas viscosity used in the reservoir models.

$$K = \frac{(9.4+0.02MW_g)T^{1.5}}{209+19MW_g+T} \quad \textbf{Eq. 4.12}$$

$$X = 3.5 + \frac{986}{T} + 0.01MW_g \quad \textbf{Eq. 4.13}$$

$$Y = 2.4 - 0.2X \quad \textbf{Eq. 4.14}$$

$$\rho_g\left(\frac{lb}{ft^3}\right) = \frac{P(psia)MW_g\left(\frac{lb}{lbmol}\right)}{ZR(psia)T(°R)} \quad \textbf{Eq. 4.15}$$

$$\mu(cp) = K * e\text{^}(X\rho_g^Y) \quad \textbf{Eq. 4.16}$$

where: T = temperature (degrees Rankine)
MW_g = molecular weight of gas (lb/lbmol)
ρ_g = gas density (ft^3/lb)
P = pressure (psia)
Z = gas compressibility factor
R = gas constant (10.73 ft^3-psi/degree Rankine-lbmol)

Given that the viscosity is dependent upon the density of the gas, which from Equation 4.15 above is dependent upon the gas compressibility factor, so too the viscosity is dependent upon the gas compressibility factor. Therefore, the gas compressibility factor must be known prior to solving for the gas viscosity.

4.2.5 Data Validation

Correlations and equations run the risk of errors given the loquaciousness each of these equations and correlations mentioned above. Therefore, prior to finalizing the reservoir construction each of these properties was validated using a reference of previous data to ensure

the equations were all running properly. Failure to validate the equations could result in erroneous data, completing eradicating the validity of the entire study. Validation runs were completed for the radial discretization, the PVT stages, the Z-factor, the gas viscosity, and the permeability data ensuring the endpoints were accurately aligned. An example of the validation process can be found in Appendix A for the Z-factor.

4.2.6 Well Information

The final consideration in constructing the reservoir model surrounded the well information. During production at a constant rate, the pressure within the reservoir will decline because the pressure serves as the drive mechanism for production. While the fluid escapes the well, the pressure begins to decline. This drop is quite pronounced during the early stages of the production of the well. For this reason, more time steps were needed to understand this decline in pressure during these early stages. In this study, the first trial considered the case where production was held steady, while pressure decreased (pressure transient analysis). In simulated time, the production began on the first day in January. In the first month of production, time steps were taken every hour during the first day, every three hours for days two through four, every six hours for days five through ten, every eight hours for days ten through fifteen, and every twenty-four hours for days fifteen to thirty-one. The initial results in stage one from this reservoir set up was not applicable for the purposes of this study. Therefore, a different well specification was put into place for the final two stages of research development.

Rather than holding the producing rate steady, a constant bottom hole pressure was specified allowing for a declining producing rate. Therefore, this study utilizes rate transient analysis. The time steps for the trial within the first month remained the same. However,

following the first month, time steps were taken at the end of each month for the following 23 months. The wells were set to produce for a total of 24 months, or two years, for this study.

4.3 Production Data

After constructing the input files for 2,000 reservoir models within the computer coding software, the information was compiled into a batch file. The batch files provided the executable data that was run in the reservoir simulation interface. After running the batch files through the CMG simulator, the production data can be extracted from the CMG simulations results. For this study, the important production parameters extracted from the results included the daily gas rate in standard cubic feet per day, the daily water rate in barrels per day, and the bottom hole pressure in pounds per square inch. This information was compiled into an RWD file to remain accessible for the inputs into the artificial neural network later in the experimental procedure.

4.4 Construction and Cleaning of Input and Output Parameters

The final stage prior to the development of the artificial neural network is to format the input and output parameters into one array of input and one array of output that can be loaded into to the neural network. Once the inputs and outputs are in this usable format, the data was cleaned to remove any and all cases which contained erroneous, invalid, or improbable data.

One limitation placed on the data eliminated cases that showed an invalid correlation of permeability versus porosity. For example, in a reservoir with minimal porosity, permeability values are also expected to be relatively low. Therefore, cases which violate this understading were eliminated from the testing data. A dissertation given by Chidambara (2009) noted a study completed by Coalson and Hartman. Their study provided an effective permeability versus porosity envelope. Any points outside of the envelope were deemed to be either impossible or highly improbable, thus they were removed from the study. Figure 8 provides an illustration of

this effective envelope. This envelope was applied to this study to eliminate invalid permeability versus porosity cases. Each case was run to ensure that its points fell within the lines bounding the envelope.

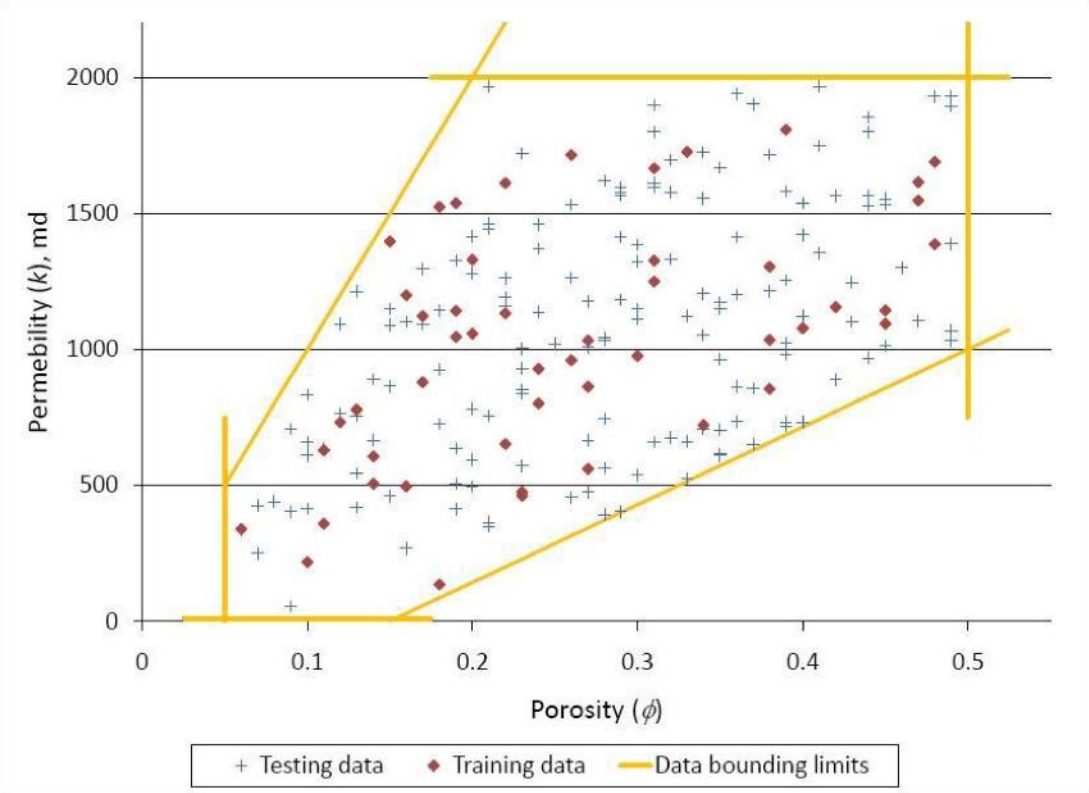


Figure 8. Coalson-Hartwell permeability and porosity envelope used to eliminate invalid cases

Another data cleaning parameter used in this study pertained to daily water rate. A heuristic approach established the water production limit per day. The data in this study pertained to tighter reservoirs with typically low water saturations. With these conditions present, it would be expected to see little water production. Therefore, the limit for water production was set to 10 barrels of water per day. Any case with production exceeding this value was eliminated from the data set.

The final parameter established to clean the data was a pressure parameter. Given that a constant bottom hole pressure was established, the reservoir production data should not have any points where the bottom hole pressure drops or rises in a significant fashion. Therefore, cases where the bottom hole pressure dropped or increased more than 10 pounds per square inches were eliminated from the data set.

Chapter 5

Artificial Neural Network Development

After generating the data sets, completing the construction of the reservoir model, and collecting the simulation production results, MATLAB's neural network toolbox was used to develop artificial neural networks. The intention of the network is to learn the functional relationship of the inputs and outputs. At this point the network will be able to demonstrate its ability to produce representative relative permeability curves for the entire drainage area of the modeled reservoirs based upon Corey's Model for relative permeability from the production data. In this chapter, the design of the feed forward network is presented and discussed in detail. The feed forward network is used to learn the non-linear relationship and uses the data to produce the representative permeability curves from the output parameters. The first section of the chapter provides an introduction to highlight an overview of the process used to develop the network. The remaining portion of the chapter discusses in more depth the structure of the network and the details behind the development of each network.

5.1 Introduction

The process to develop a satisfactory artificial neural network proves to be complex and involves a great deal of experimentation with the neural network toolbox algorithms. It also requires numerous steps of trial and error to ultimately reach the optimum network structure. Several factors play a role in the structure of artificial neural network and no one structure is the solution to every problem. The optimum network structure for one problem may differ greatly from the structure of a different problem, which explains the need for trial and error in arriving at

the optimum network structure. Gaining experience with ANNs allows the developer to understand the necessary requirements for their network and helps in applying the ideal functions and algorithms to arrive at the best case network. The following provides a set of general guidelines and methods used in the network development.

Data Normalization: Prior to developing the network, both the input and output were normalized to fit between negative one and positive one. The transfer functions within MATLAB's neural network toolbox operate at values between negative one and positive one. While a transfer function itself is capable of performing this normalization, during the transferring of the data, manually completing this prior to the network's training results in a more efficient process. With the vast amount of data entered into the network, saving the network one less step provides better results and produces a faster performing network.

Dataset Division: Once the data set was cleaned for any invalid cases (Chapter 4) the data was randomly divided into 90% for training the network, 5% for validation, and 5% for blind testing. A large majority of the data set was randomly selected for training the network to ensure ample data would be available for the network to learn the existing functional relationship between the inputs and outputs. The validation is required to ensure that during the training of the network neither under-fitting of the data nor over learning of the data takes place. The final portion of the divided data set was allocated for the blind testing of the network. This data was unused during the training of the network. The input data was introduced to the network and the output results were compared to those of the simulated results. The error between the network and the simulation data was desired to be within a $\pm 10\%$. Error exceeding this value resulted in adjustments in the network's topography and the implementation of functional links in subsequent stages of network development.

Selection of Algorithms and Functions: MATLAB's neural network toolbox offers a number of different training, learning, and transfer functions. Five different transfer functions and four different training functions were randomly tested during each stage to see which would produce the best case results for the network. The learning algorithm which performed the best and was selected for each stage of the network's development was the gradient descent with momentum weight and learning bias (*learngdm*). Lastly, the mean sum of squares of the network error with regularization performance function (*msereg*), was used to regulate the network's performance during the training of the network.

Number of Iterations (Epochs) and Validation Checks: Both of these are features that the user may establish within the MATLAB interface. It is important to establish a number of epochs that allows the network to properly learn the functional relationship between the inputs and outputs without over-learning the relationship, which can also lead to an errand network. The number of necessary iterations differs from network to network and ultimately depends upon the complexity of the problem at hand. For this study, this number was set to 100,000 to ensure the networks would learn the relationship between the inputs and outputs. However, nearly all networks in all stages completed their training in less than 10,000 iterations. Additionally, 2,000 validation checks were set. Once the iterative process does not provide a network that produces a lesser mean squared error for 2,000 iterations, the training process stops. During training, the neural network interface window (Figure 9), provides the user the ability to monitor the performance of the network during training.

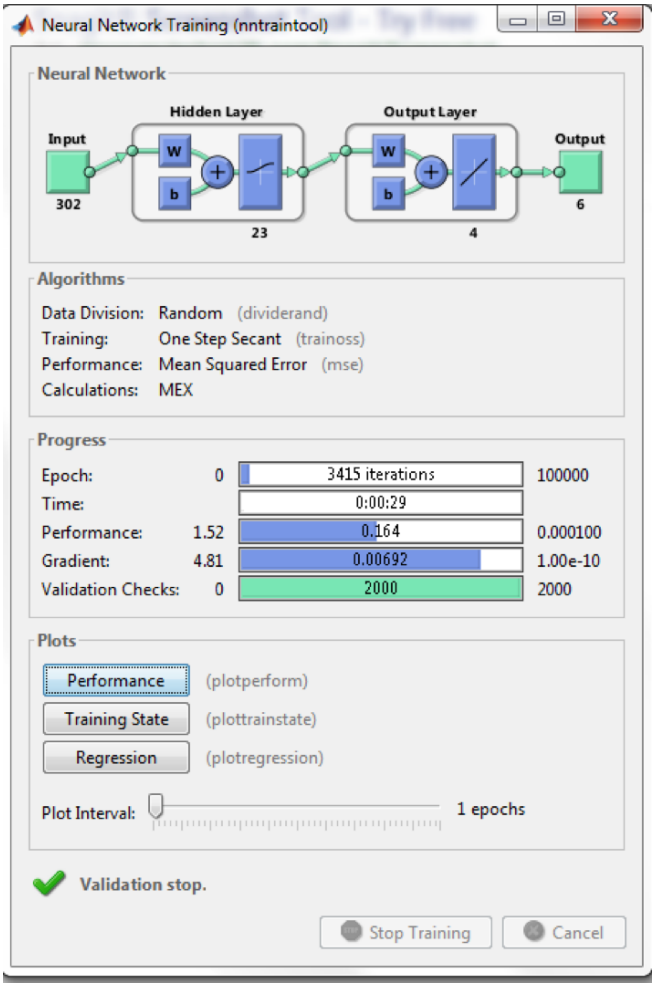


Figure 9. MATLAB Neural Network Toolbox interface during network training

Number of Hidden Layers and Neurons: These two parameters can be specified by the user or set to random numbers. However, there exists a number of hidden layers and neurons, which produces optimal results. Finding the optimum number could take numerous hours, however, in each stage of development, a one, two, and three-layer network was generated, and in each layer a random number of neurons of between 0 and 500 was generated. This allowed for each possible combination of networks to be tested during training. While each network's training took several hours to run, each stage only had to be run one time to arrive at the optimal ANN

structure. The performance error of each was analyzed during the process, and the network with the lowest performance error was taken as the optimum or "best" network.

Functional Links: After two stages of network development, functional links were found to be useful in improving the network's performance. The functional links were mathematical relationships between the input parameters used in the network. Ultimately, functional links provide more input for the network and provide a functional relationship prior to the network's attempts to learn the relationship between the inputs and outputs. Two functional links were used in the third stage of development, which will be highlighted later in this chapter.

Error Analysis: Following the training process, the network's performance was assessed by comparing the generated output from the network during blind testing to the results that were provided from the CMG simulation run. For each data point, an analysis was conducted using Equation 5.1 below.

$$\%\ Error = \left|\frac{Value_{CMG\ Simulation} - Value_{ANN\ Prediction}}{Value_{CMG\ Simulation}}\right| * 100 \quad \textbf{Eq. 5.1}$$

With a percent error for each data point for a given output parameter, the average error for that specific parameter was calculated using Equation 5.2 below. If the average error for each output parameter (Corey's Model parameter) exceeded $\pm 10\%$ a further stage of development was conducted to produce a more optimal network model.

$$Parameter\ Average\ Error\ \% = \frac{\sum_{i=N}^{N} \%\ Error\ Paremeter_N}{N} \quad \textbf{Eq. 5.2}$$

Where: N = the total number of data points per parameter

For example, each parameter in the final development stage had 60 data points during blind testing. Therefore, N is equal to 60 for all cases computed during stage three of this study.

5.2 Stage 1 of Artificial Neural Network Development

This sections provides a detailed analysis of the process used to develop the artificial neural network during the first stage of this study. The network operated as a cascade feed forward network with back propagation. Table 4 and Table 5 provide the input and output parameters for the ANN.

Table 4. The inputs consisting of the reservoir properties, PVT properties, and production information used during the first stage of the network development

Inputs for Stage 1 ANNs	
Reservoir Properties	Initial Pressure
	Specific Gas Gravity
	Formation Compressibility
	Reservoir Thickness
	Reservoir Temperature
	Porosity
	Initial Water Saturation
	Initial Gas Saturation
PVT Properties	10 Pressure Stages
	Gas Expansion Factor @ Each Pressure
	Gas Viscosity @ Each Pressure
Production Information	Producing Water Rate @ each time step
	Bottom Hole Pressure @ each time step

Table 5. The Corey's Model parameters that served as the output for each stage of network development.

Outputs for Stage 1 ANNs
Relative Permeability of Gas at Connate Liquid
Relative Permeability of Water at 100% Water Saturation
Critical Gas Saturation
Connate Water Saturation
Corey Exponent for Gas
Corey Exponent for Water

In this first stage, the data set consists of 1600 cases, after the cleaning of irrelevant cases took place, which eliminated 400 cases. From this point on, the data was divided into 96% for testing, 2% for validation, and 2% for blind testing. During each stage of this study three different networks were constructed, one with one hidden layer, one with two hidden layers, and one with three hidden layers. Different training functions and transfer functions were used during training to create the optimum network structure. In total, there were five different transfer functions tested. These included the log-sigmoid transfer function (*logsig*), the hyperbolic tangent sigmoid transfer function (*tansig*), the radial basis transfer function (*radbas*), the saturating linear transfer function (*satlin*), and the symmetric saturating linear transfer function (*satlins*). There were four training functions that were tested in this study including, the one step secant backpropagation training method (*trainoss*), the conjugate gradient backpropagation with Polak-Ribiére updates training method (*traincgp*), the conjugate gradient backpropagation with Powell-Beale restarts training method, and the gradient descent backpropagation training method (*traingd*). Each network in this study used the gradient descent with weight and bias learning function (learngdm).

5.2.1 Stage 1 Optimum Artificial Neural Network Structure

The optimum ANN structure for stage 1 was a one-layer network with 23 neurons in the hidden layer. As illustrated by Table 4 there were 250 input neurons consisting of 212 data points from the producing water rate and the bottom hole pressure at each time step. For the PVT properties, there were ten pressure stages with the inclusion of the gas expansion factor and the gas viscosity at each pressure. The final input neurons consisted of the reservoir properties mentioned in Table 4. There were six output neurons made up of Corey's relative permeability model parameters (Table 5). The optimal network used the one step secant backpropagation training method (*trainoss*), and the transfer function used in the transfer of data from neuron to neuron was the log-sigmoid transfer function (*logsig*). During the first stage of this project, the time required for the training of the networks increased with the addition of additional hidden layers. This remained true throughout the subsequent development stages.

5.2.2 Error Analysis and Results of Stage 1

The desired target error tolerance for each property was under ten percent. Typically, in engineering applications an error below twenty percent proves to be acceptable, but for the purpose of this study better results were desired to show the true capability of the network in its ability to capture the functional relationship between the inputs and Corey's relative permeability model parameters. If the parameters were not predicted with a mean error less ten percent, a new stage of network development took place.

In general, the critical gas saturation and the Corey exponent for gas showed the highest degree of error, while the error for water properties was much lower. The explanation for the phenomenon lies in the relationship between the gas parameters and water parameters. More input data with a far wider range of values was presented for gas compared to water. For

example, the gas rate fell in a range of 0-100 MMSCF/D while the water rate fell within a narrower range of 0-10 bbl/d. As a result, the gas related parameters in Corey's model underperformed compared to the water parameters.

Prior to presenting results from the optimum ANN structure displayed in Table 6, it is important to compare the results from the other ANNs developed during this stage of the study. This comparison will serve as a tool for selecting the optimum ANN structure in the subsequent stages of ANN development. A similar approach of investigation was used throughout the remaining stages to make necessary changes to build the remaining ANN models. For the first stage and second stages of development, it was observed that adding more hidden layers hindered the results of the network. The ANN with only one hidden layer outperformed the structures with multiple hidden layers in predicting the values for the parameters of Corey's relative permeability model. Table 7, Table 8 and Table 9 provide an overview of the error analysis of each of the outputs from the three different networks constructed in this stage. It was observed that increasing the number of neurons in the one-layer structure resulted in overlearning, while in the multiple layered networks exceeding 300 neurons in a given layer resulted in overlearning. Table 10 provides an overview of each networks overall performance and the amount of neurons found in the hidden layers of each network.

Table 6. Stage 1 - 1 hidden layer ANN average error results for Corey's model parameters

Stage 1 – 1 Hidden Layer	
Property	**Average Error**
Swcon	55.95
Sgcrit	148.03
krgcl	0.00
krwiro	0.00
ng	17.92
nw	18.57
Avg. Error	40.08

Table 7. Stage 1- 2 hidden layers ANN average error results for Corey's model parameters

Stage 1 – 2 Hidden Layers	
Property	**Average Error**
Swcon	65.24
Sgcrit	153.14
krgcl	0.00
krwiro	0.00
ng	72.87
nw	35.21
Avg. Error	54.41

Table 8. Stage 1- 3 hidden layers ANN average results for Corey's model parameters

Stage 1 - 3 Hidden Layers	
Property	**Average Error**
Swcon	72.24
Sgcrit	115.87
krgcl	0.00
krwiro	0.00
ng	55.24
nw	31.21
Avg. Error	45.76

Table 9. Summary of number of neurons per hidden layer and error analysis for each network in Stage 1

Stage 1 Development Summary					
	Neurons	**Absolute Summation Error**	**Average Error For Corey Parameters**	**Min Error (%)**	**Max Error (%)**
1 Layer Network	23	0.2673	40.08%	0.0823	1003.21
2 Layer Network	256-131	0.3516	54.41%	0.8732	973.78
3 Layer Network	96-341-137	0.2902	45.76%	0.7332	999.34

From these results, the best model is the one-hidden layer network. All of these table values were taken from finding the average of error between the predicted results from the ANNs and the actual results produced during the reservoir simulation runs. For a graphical representation of the predicted neural network values for Corey's relative permeability model versus the actual simulated values, Figures 10-15 show the graphs that were created with the results from the optimum network structure of stage one. There are 32 cases that were used in the blind testing representing the 2% of the data allocated for this purpose.

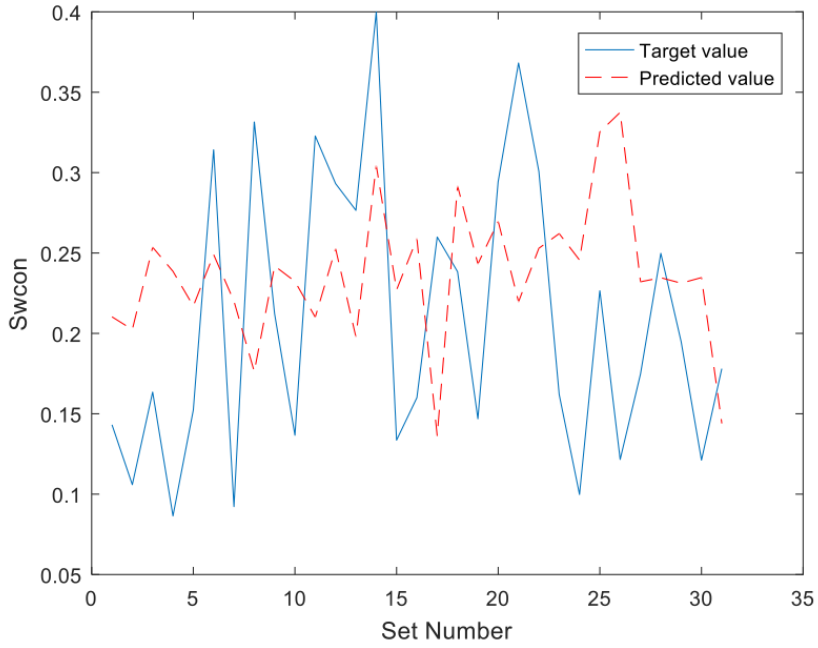


Figure 10. Results for the target connate water saturation and predicted connate water saturation by the 1-layer network in stage 1.

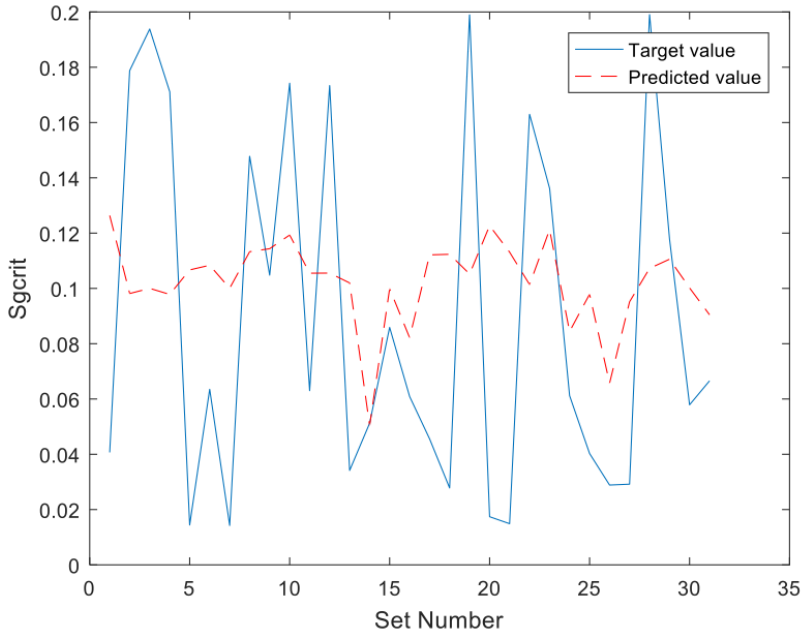


Figure 11. Results for the target critical gas saturation and predicted critical water gas by the 1-layer network in stage 1.

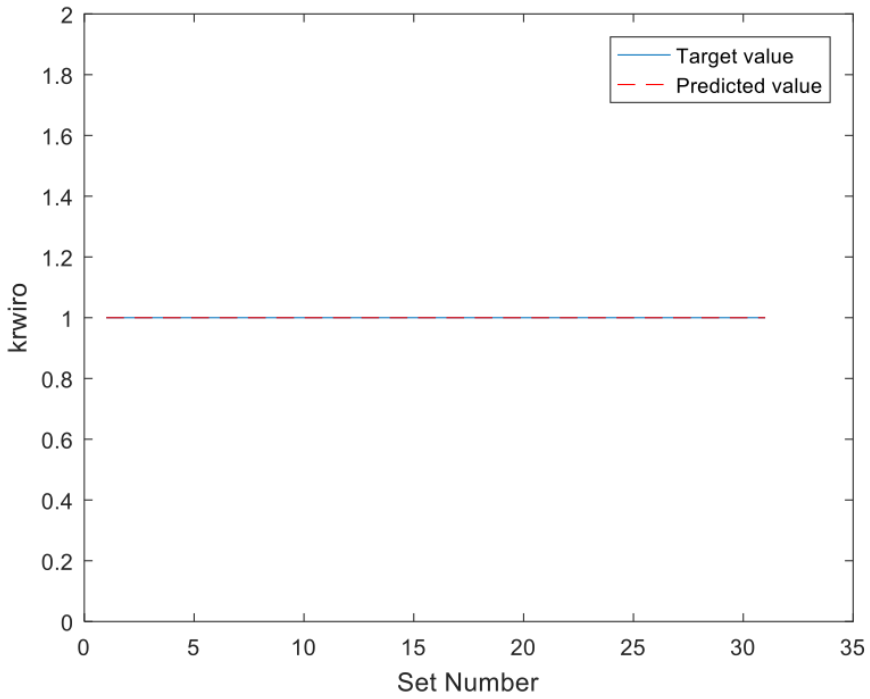


Figure 12. Results for the target relative permeability of water @ 100% water saturation and predicted relative permeability of connate water @ 100% water saturation by the 1-layer network in stage 1.

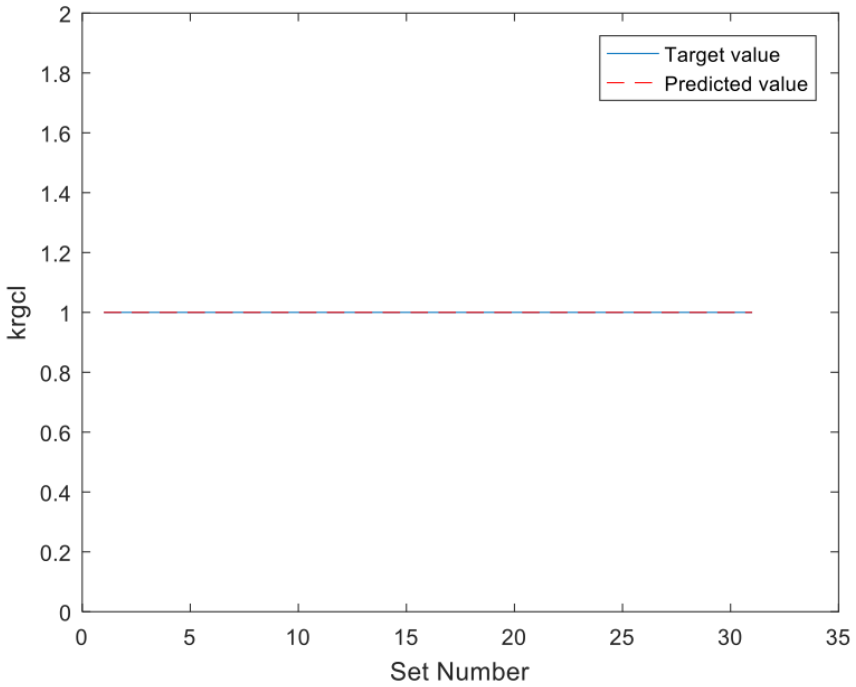


Figure 13. Results for the target relative permeability of gas at connate liquid and predicted relative permeability of gas at connate liquid by the 1-layer network in stage 1.

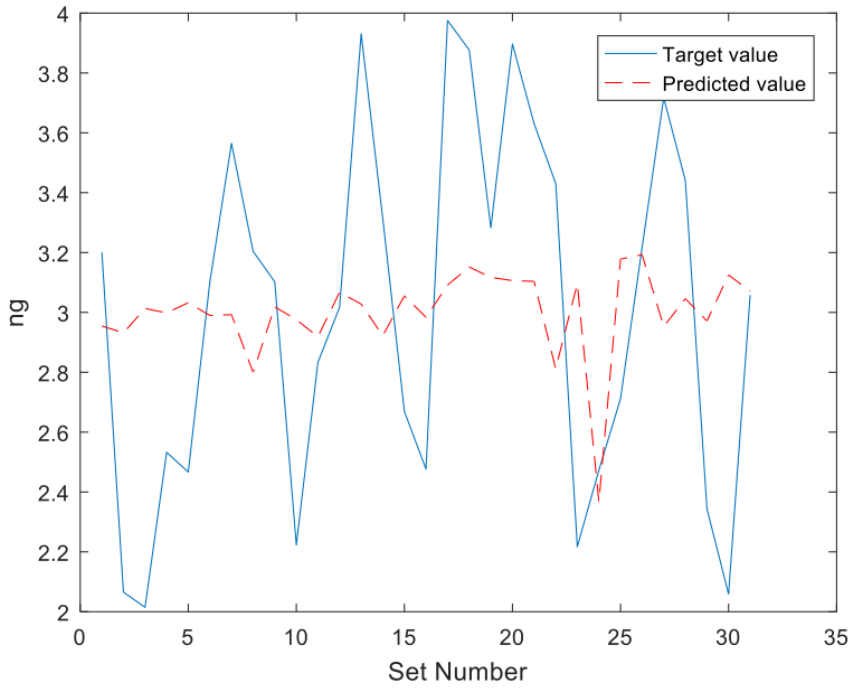


Figure 14. Results for the target Corey exponent for gas and predicted Corey exponent for gas by the 1-layer network in stage 1

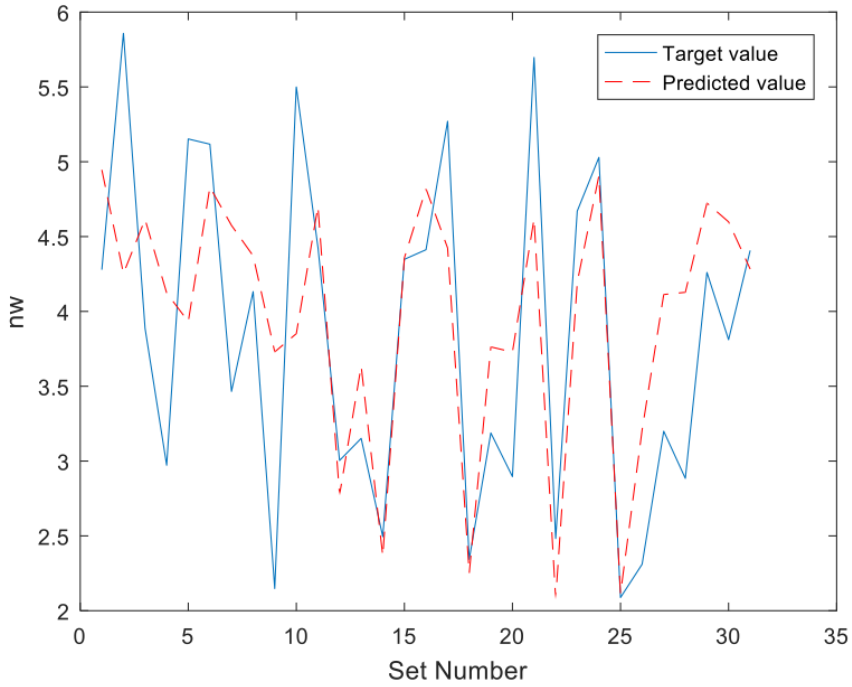


Figure 15. Results for the target Corey exponent for water and predicted Corey exponent for water by the 1-layer network in stage 1.

Based upon the error results from Table 6 and Table 9 and as evident from Figures 10-15, the optimum ANN structure for this case did not predict the Corey relative permeability model parameters with an average below $\pm$10% for the blind testing cases. Therefore, a second stage of network development was required and will be discussed in the next section.

5.3 Stage 2 of Neural Network Development

This section outlines the changes made to the network structure and the inputs and outputs used in the effort of increasing the network's performance during the second stage of development. Like the networks constructed in the first stage of network development in this study, the network operated as a cascade feed forward network with back propagation. Like the first stage of this study, there were 1,600 reservoir simulation cases after the cleaning of the data. Again, 96% of the data was used for training the network, 2% was used for validation, and 2% was used in blind testing.

However, the major change in this stage of network development was changing the data from pressure transient data to rate transient data. Instead of producing at a constant gas rate, the producing well used in the reservoir simulation operated at a constant bottom hole pressure of 100 psia. This allowed for the gas rate to decrease during the production of the well. In addition, instead of producing for just one month, the well produced gas for two years. The time steps remained unchanged in the first month of production while the rates were recorded at the end of the remaining 23 months. This change resulted in more input neurons. The second stage of network development utilized 296 inputs, while the six outputs remained unchanged. Table 10 illustrates the inputs for the second stage of network development. Because the outputs remained unchanged, Table 5 provides the outputs (Corey's relative permeability model parameters) for this stage of development.

Table 10. The inputs consisting of the reservoir properties, PVT properties, and production information used during the second stage of the network development.

Inputs for Stage 2 ANNs	
Reservoir Properties	Initial Pressure
	Specific Gas Gravity
	Formation Compressibility
	Reservoir Thickness
	Reservoir Temperature
	Porosity
	Initial Water Saturation
	Initial Gas Saturation
PVT Properties	10 Pressure Stages
	Gas Expansion Factor @ Each Pressure
	Gas Viscosity @ Each Pressure
Production Information	Producing Water Rate @ 106 time steps
	Producing Gas Rate @ 106 time steps

5.3.1 Stage 2 Optimum Artificial Neural Network Structure

The optimum ANN structure for stage two was a one-layer network with 41 neurons in the hidden layer. As was previously mentioned, there were 250 inputs neurons and 6 output neurons. The optimal network used the conjugate gradient backpropagation with Polak-Ribiére updates training method, and the log-sigmoid (*logsig)* transfer function was used to transfer the data from neuron to neuron. Similar to the first stage of development, the time required for network training increased with the addition of hidden layers to the network model. Therefore, the optimum one hidden layer network required the least amount of time to identify the functional relationship between the inputs and outputs.

5.3.2 Error Analysis and Results of Stage 2

In general, as was the case in stage one, the critical gas saturation and the Corey exponent for gas showed the highest degree of error, while the error for water properties was much lower. Like with the first network, the amount of data and the range of data affecting the gas parameters far exceeded the amount of data that affected the water parameters. This reflects in the higher degree of error of these properties for each network structure in this portion of the study. Compared to the first stage of development, the second stage of development slightly improved the results, which can be seen in Tables 11-14. The tables make it clear that the one-layer network continued to outperform the multiple hidden layered networks.

Table 11. Stage 2 - 1 hidden layer ANN average results for Corey's model parameters

Stage 2 - 1 Hidden Layer	
Property	**Average Error**
Swcon	66.56
Sgcrit	93.85
krgcl	0.00
krwiro	0.00
ng	18.33
nw	13.34
Avg. Error	32.01

Table 12. Stage 2 - 2 hidden layers ANN average results for Corey's model parameters

Stage 2 – 2 Hidden Layers	
Property	**Average Error**
Swcon	86.43
Sgcrit	65.24
krgcl	0.00
krwiro	0.00
ng	23.00
nw	24.38
Avg. Error	33.17

Table 13. Stage 2 - 3 hidden layers ANN average results for Corey's model parameters

Stage 2 – 3 Hidden Layers	
Property	**Average Error**
Swcon	60.50
Sgcrit	112.95
krgcl	0.00
krwiro	0.00
ng	18.33
nw	16.96
Avg. Error	34.79

Table 14. Summary of number of neurons per hidden layer and error analysis for each network in Stage 2.

Stage 2 Development Summary					
	Neurons	**Absolute Summation Error**	**Average Error For Corey Parameters**	**Min Error (%)**	**Max Error (%)**
1 Layer Network	41	0.1517	32.01%	0.0797	899.1784
2 Layer Network	314- 385	0.1801	33.17%	0.9600	363.0774
3 Layer Network	109-388-172	0.1557	34.79%	0.9674	824.2890

From these results, the best model is the one-layer hidden network. All of these table values were taken from finding the average error between the predicted results from the ANNs and the actual results produced during the reservoir simulation runs. For the one-layer network, the average error for all parameters was 32.01% with a minimum error of 0.0797% and a maximum error 899.17%. The best results were provided for the relative permeability of water at 100% water saturation and the relative permeability of gas at connate liquid, while the worst results were for the critical gas saturation. The overall average error decreased by 8.07% from 40.08% to 32.01% from stage one. For a graphical representation of the predicted neural network values for Corey’s relative permeability model versus the actual values from the reservoir

simulation runs, Figures 16-21 show the graphs that were created with the results from the optimum network structure of stage two. There are 32 cases that were used in the blind testing representing the 2% of the data allocated for this purpose.

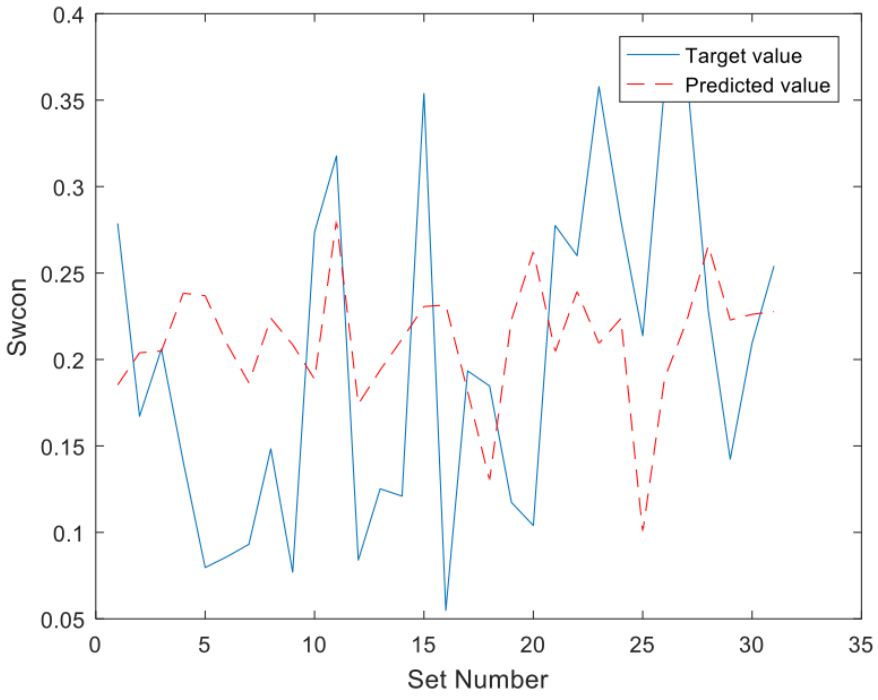


Figure 16. Results for the target connate water saturation and predicted connate water saturation by the 1-layer network in stage 2.

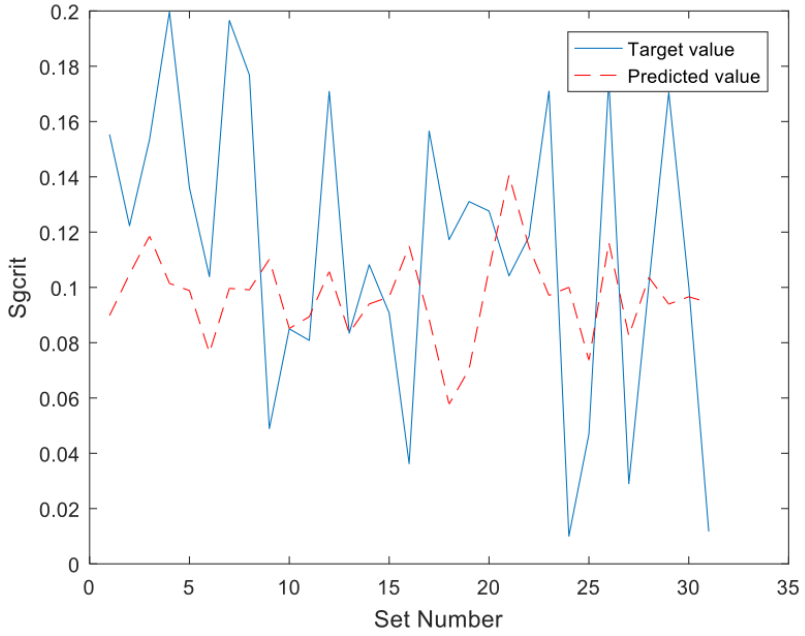


Figure 17. Results for the target critical gas saturation and predicted critical water gas by the 1-layer network in stage 2.

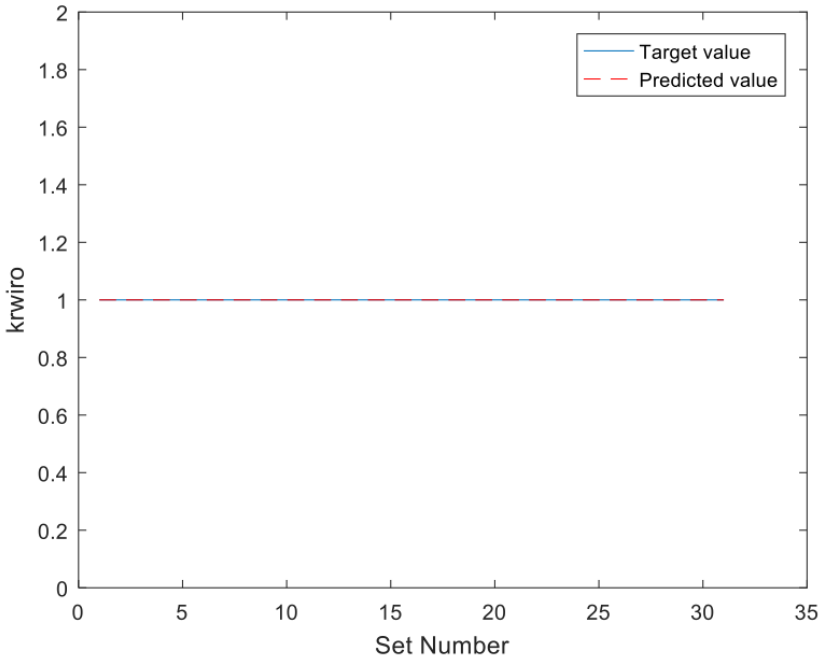


Figure 18. Results for the target relative permeability of water @ 100% water saturation and predicted relative permeability of connate water @ 100% water saturation by the 1-layer network in stage 2.

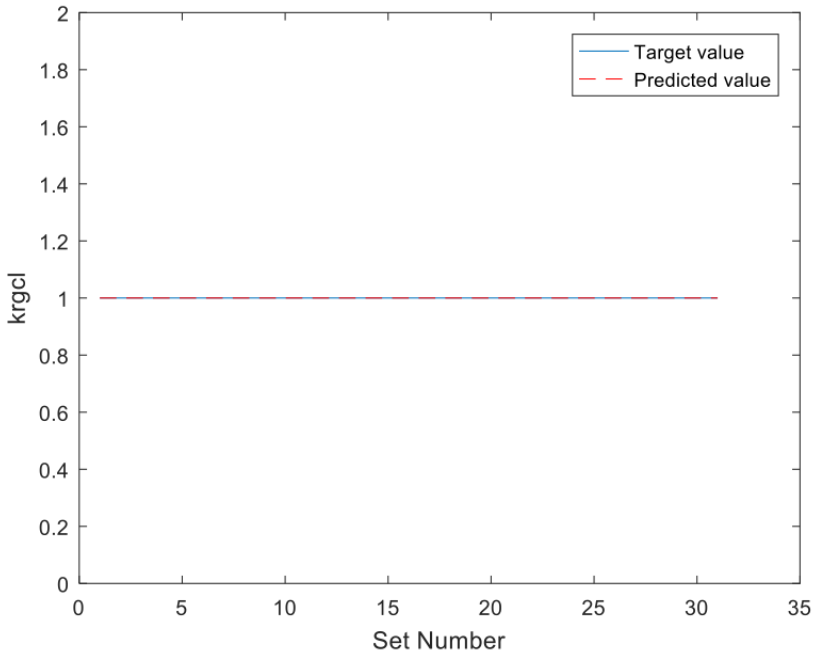


Figure 19. Results for the target relative permeability of gas at connate liquid and predicted relative permeability of gas at connate liquid by the 1-layer network in stage 2.

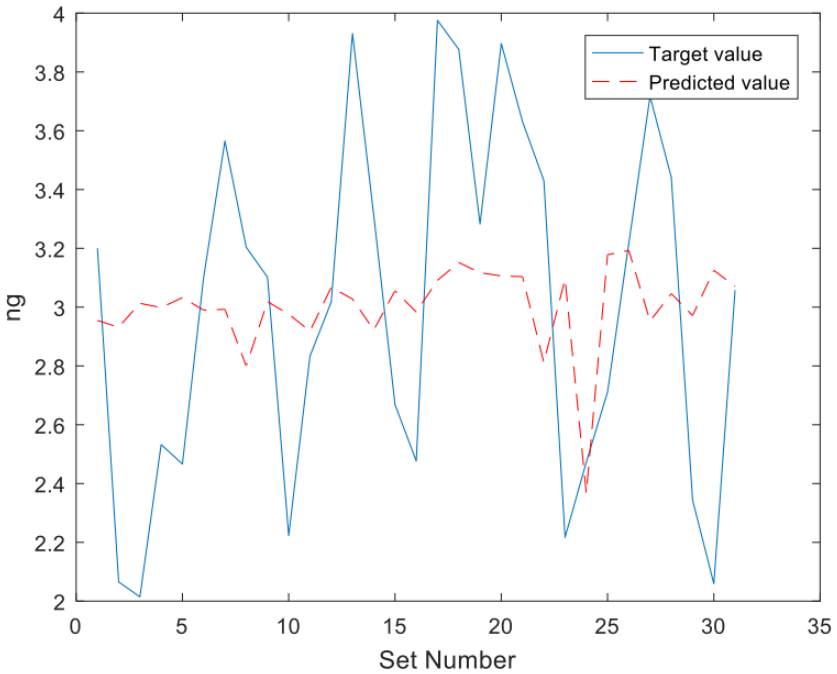


Figure 20. Results for the target Corey exponent for gas and predicted Corey exponent for gas by the 1-layer network in stage 2.

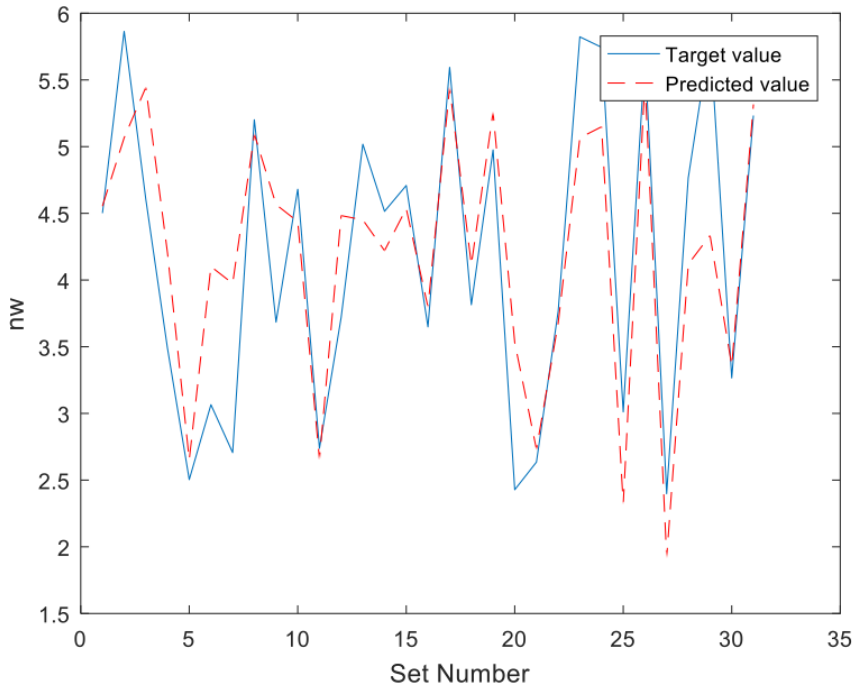


Figure 21. Results for the target Corey exponent for water and predicted Corey exponent for water by the 1-layer network in stage 2.

From the results provided by Table 11 and Table 14 in combination with graphs in Figures 15-21, the optimum structure from this stage of development did not predict the Corey parameters within a $\pm$10% average error. However, there was improvement from stage one to stage two, and as evident of Figure 21, the Corey exponent for water was closely predicted in many of the blinding testing cases meaning the steps taken to improve the model did provide the better results. Although, because of the remaining high degree of error for the rest of the properties, a third stage of network development was required in this study.

5.4 Stage 3 of Neural Network Development

This section outlines the changes made to the network structure and to the inputs and outputs used in order to increase the network's performance during the third stage of development. Like the networks in the first two stages, the three networks in this stage of development operated as cascade feed forward networks with back propagation. In addition, similar to stage two, stage three also utilized rate transient data. Because the first two stages did not produce great results, there were a several changes implemented in an effort to achieve better ANN results.

Stage three incorporated an additional cleaning parameter for irrelevant data. The new parameter required at least a 50% change in the grid block containing the producing gas well for during the production period of two years. During the life cycle of a well, it is expected that the gas saturation will decrease as the gas is extracted from the ground. Therefore, from a heuristic approach, those cases where the gas saturation did not reduce by at least 50% in the grid block containing the well were removed from the data set. More datasets were created to generate additional reservoir simulations to provide more data for the neural network training. Instead of 2,000 cases, 3,000 reservoir simulation cases were run for this stage. After cleaning the data, 2,002 cases were used for this stage of development. Lastly, two functional links were included to provide more production parameters for the network to use in an effort to improve its performance. The functional links used included the daily water rate divided by the initial daily water rate and the daily gas rate divided by the initial daily gas rate. With the addition of the functional links, this changed the number of inputs for the network. With a functional link added for 105 of the 106 time steps throughout the two-year production period, 210 inputs were added. This increased the number input neurons from 296 in stage two to 506 in stage three. Table 15

illustrates the inputs that were used for third stage of development in this study. Like the first two stage of network development, the output neurons again remained unchanged.

Table 15. The inputs consisting of the reservoir properties, PVT properties, functional links and production information used during the second stage of the network development.

Inputs for Stage 3 ANNs	
Reservoir Properties	Initial Pressure (psia)
	Specific Gas Gravity
	Formation Compressibility
	Reservoir Thickness
	Reservoir Temperature
	Porosity
	Initial Water Saturation
	Initial Gas Saturation
PVT Properties	10 Pressure Stages
	Gas Expansion Factor @ Each Pressure
	Gas Viscosity @ Each Pressure
Production Information	Producing Water Rate
	Producing Gas Rate
Functional Links	Producing Water Rate / Initial Producing Water Rate
	Producing Gas Rate / Initial Producing Gas Rate

5.4.1 Stage 3 Optimum Artificial Neural Network Structure

The optimum network structure for stage three of this study was a three-hidden layer network. The number of neurons in each of the three layers was 212, 104, and 142 respectively totaling 485 hidden layer neurons. The aforementioned number of input neurons was 506 and the number of output neurons was 6. The optimal network used the conjugate gradient backpropagation with Polak-Ribiére updates training method, and the symmetric saturating linear transfer function (*satlins*) was the transfer function used for this three-layer network. The

breakdown of the data was changed in this case to use 94% for training, 3% for validation, and 3% for blind testing. Therefore, 1,802 cases were used in the training of the network, while 60 cases were used in both the validation and blind testing of the network predictions. As in the other stages, the training time increased with the increasing number of hidden layers. Despite the fact that the three-layer network took the longest time to train and run by several minutes, it provided the best results in blind testing, which made the run time a non-factor.

5.4.2 Error Analysis and Results of Stage 3

In general, by looking at Tables 16-19 it is clear that the continued trend from the first two stages persisted in stage three; the gas related parameters of Corey's relative permeability model proved to show the highest degree of error. However, in this stage, the level of error improved to be below $\pm$10% for all parameters for the three-layer network. The overall average error for each of the models in this stage was under $\pm$10%, however, the three-layer network provided the best performance. The overall parameter error for the three-layer network was 4.38% while the two-layer network had an average error of 6.36%, and the one-layer network had an average error of 9.51%. The tables below show the superior performance of the three-layer network compared to the other two models in this stage.

Table 16. Stage 3 - 1 hidden layer ANN average results for Corey's model parameters

Stage 3 - 1 Layer	
Property	**Average Error**
Swcon	14.71
Sgcrit	18.14
krgcl	0.00
krwiro	0.00
ng	19.64
nw	4.56
Avg. Error	9.508333333

Table 17. Stage 3 - 2 hidden layers ANN average results for Corey's model parameters

Stage 3 - 2 Layer	
Property	**Average Error**
Swcon	12.14
Sgcrit	17.23
krgcl	0.00
krwiro	0.00
ng	7.30
nw	1.50
Avg. Error	6.36

Table 18. Stage 3 - 3 hidden layers ANN average results for Corey's model parameters

Stage 3 - 3 Layer	
Property	**Average Error**
Swcon	7.8
Sgcrit	9.81
krgcl	0.00
krwiro	0.00
ng	7.54
nw	1.14
Avg. Error	4.381666667

Table 19. Summary of number of neurons per hidden layer and error analysis for each network in Stage 3.

Stage 3 Development Summary					
	Neurons	**Absolute Summation Error**	**Average Error For Corey Parameters**	**Min Error (%)**	**Max Error (%)**
1 Layer Network	48	0.0542	9.51%	0	67.13
2 Layer Network	236 -153	0.0498	6.36%	0	62.95
3 Layer Network	212- 104- 142	0.0286	4.38%	0	56.08

The three-layer network performed well in determining the relationship between the inputs and Corey's relative permeability model parameters. The average error for connate water

saturation was 7.8%, 9.81% for the critical gas saturation, 0% for both the relative permeability of gas at connate liquid and the relative permeability of water at 100% water saturation, 7.54% for the Corey exponent of gas, and 1.14% for the Corey exponent for water. Figures 22-27 provide the case by case analysis for the predicted values of the three-ANN compared to the target values provided by the simulation cases. There were 60 test cases used for blind testing in this stage to represent the 3% of the 2,002 cases in the data set for stage three.

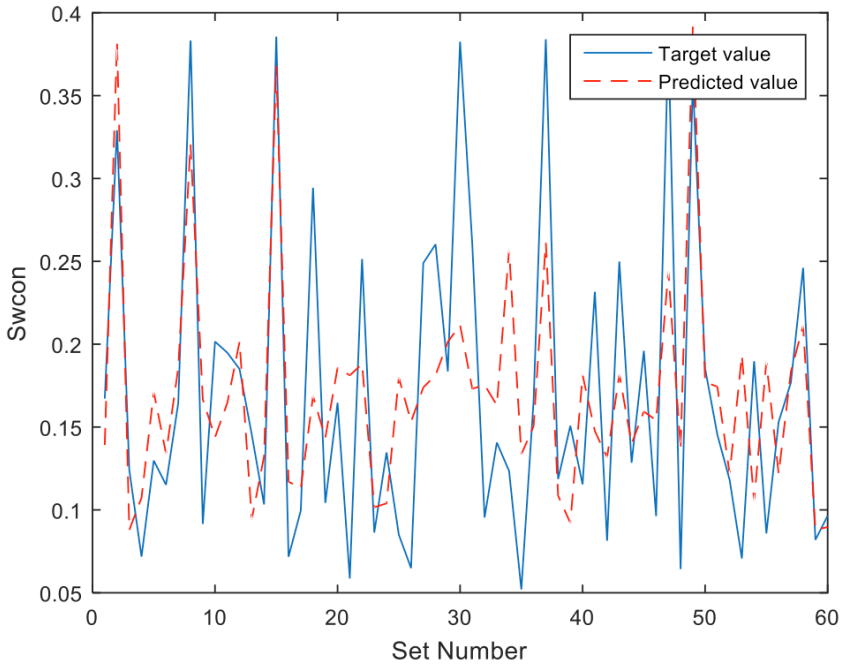


Figure 22. Results for the target connate water saturation and predicted connate water saturation by the 3-layer network in stage 3.

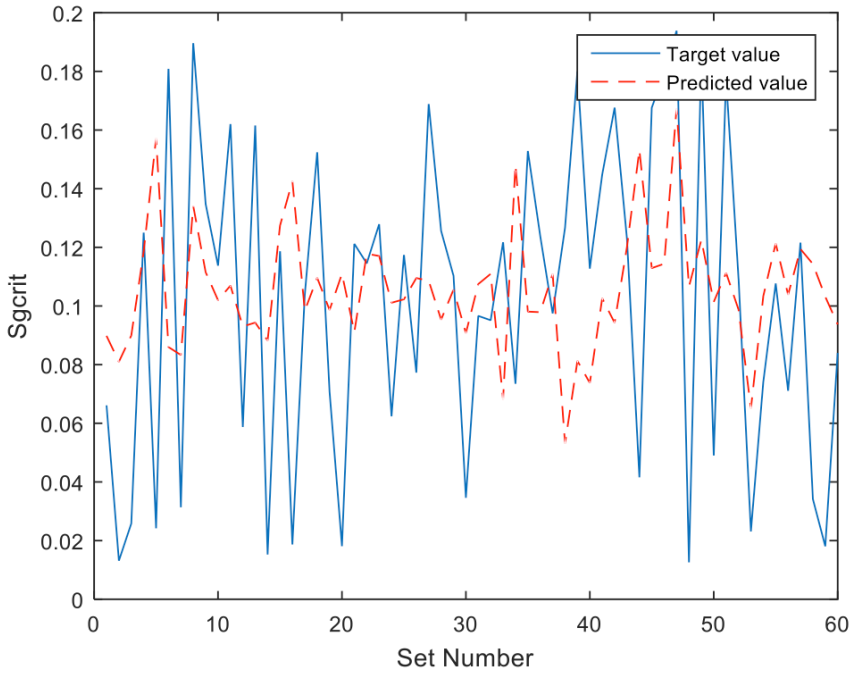


Figure 23. Results for the target critical gas saturation and predicted critical water gas by the 3-layer network in stage 3.

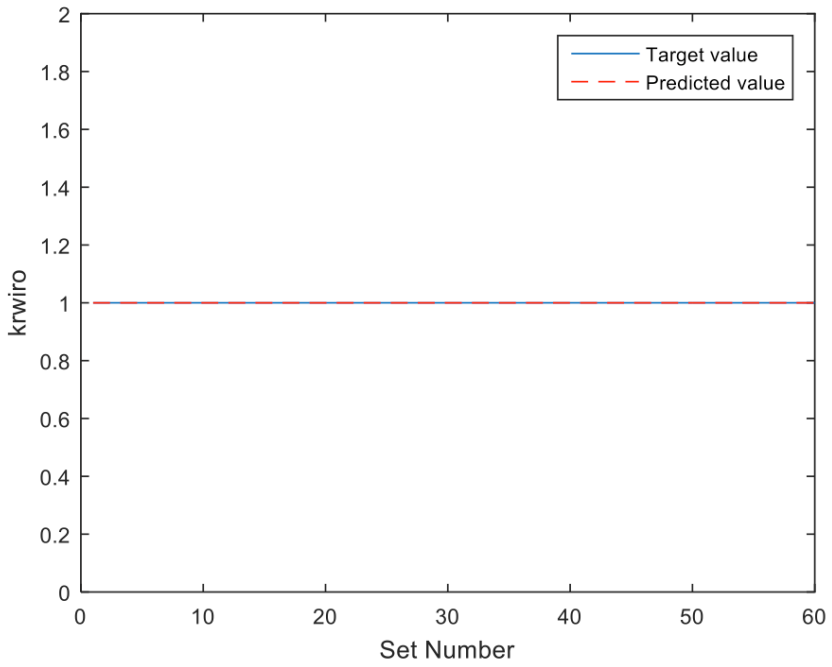


Figure 24. Results for the target relative permeability of water @ 100% water saturation and predicted relative permeability of connate water @ 100% water saturation by the 3-layer network in stage 3.

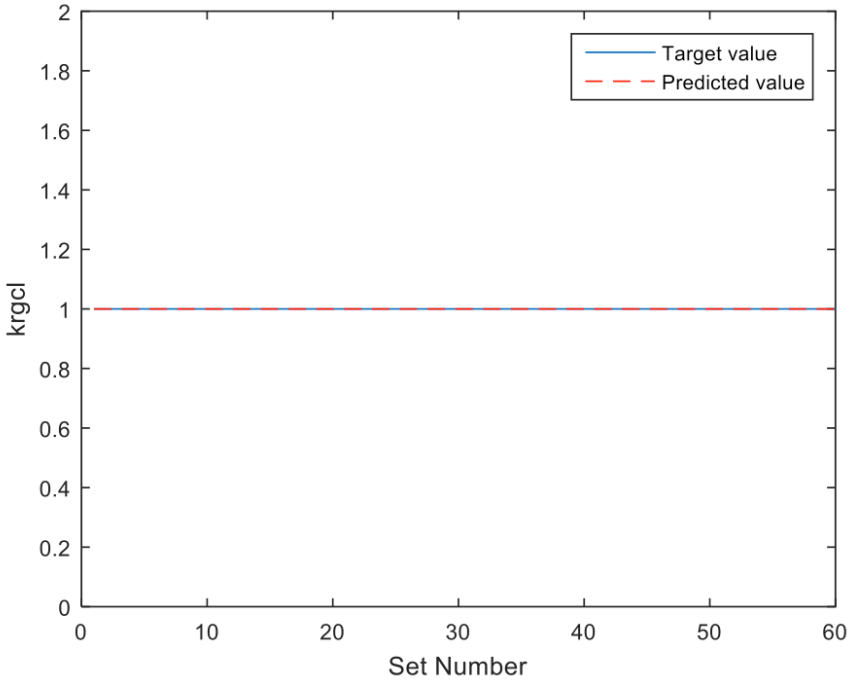


Figure 25. Results for the target relative permeability of gas at connate liquid and predicted relative permeability of gas at connate liquid by the 3-layer network in stage 3.

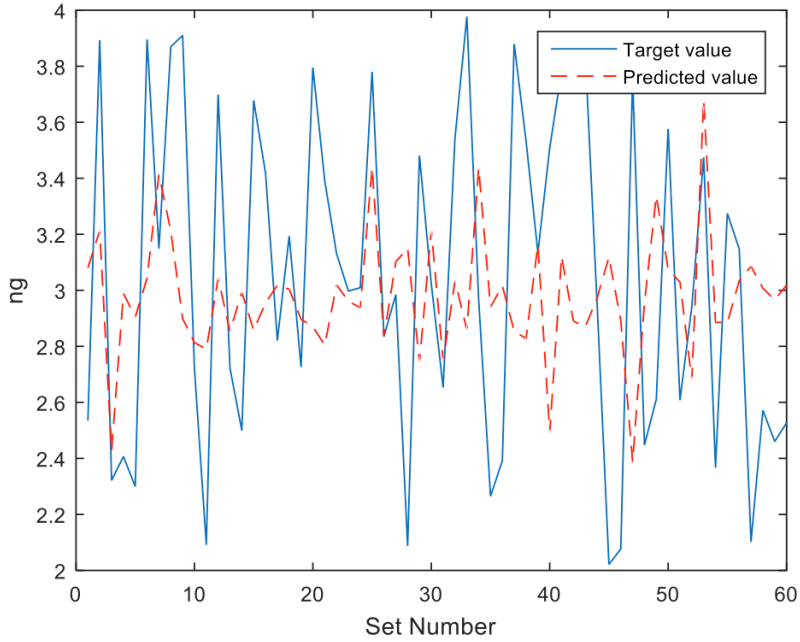


Figure 26. Results for the target Corey exponent for gas and predicted Corey exponent for gas by the 3-layer network in stage 3.

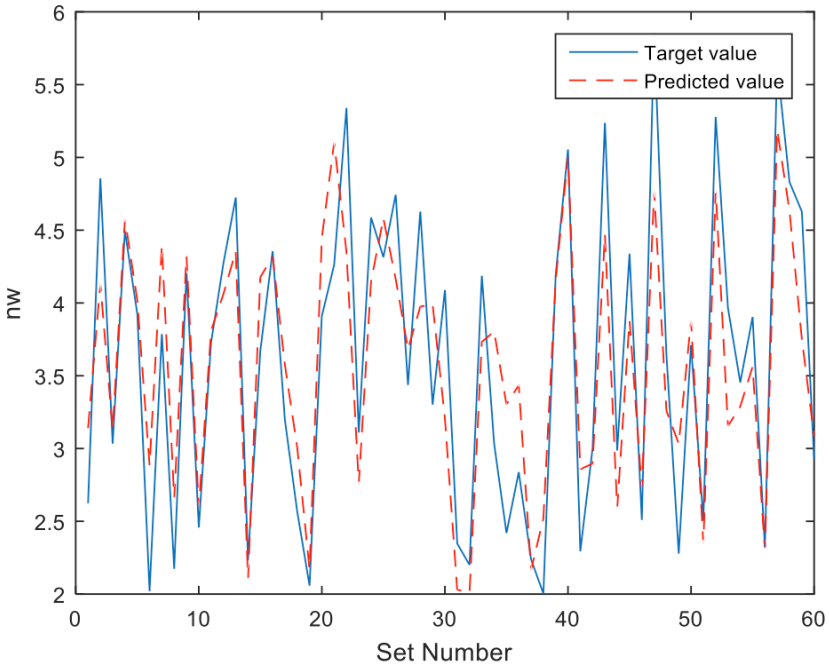


Figure 27. Results for the target Corey exponent for water and predicted Corey exponent for water by the 3-layer network in stage 3.

The results from Tables 16-29 and the graphs from Figures 22-27 illustrate the network's capability in successfully identifying the functional relationship between the inputs and outputs. Because all of the properties were predicted within a $\pm$10% error and for some of the properties with nearly no error, representative relative permeability graphs for the entire drainage area were constructed in Figures 28-37. The curves were constructed using Equations 1.3 and 1.4 from Chapter 1. The ten graphs selected represent a range of typical cases, the best cases, and the worst cases from the 60 blind testing cases. The dashed lines on the graphs show the predicted results by the three-layer ANN for both the relative permeability of gas and the relative permeability of water, while the solid lines represent the relative permeability of gas and the relative permeability of water for the reservoirs simulation cases.

The error for the relative permeability was well within the acceptable error range for engineering applications. Given that the Corey relative permeability model parameters governing the water permeability were predicted better than gas the parameters, accordingly the relative permeability of water was more accurately predicted than the relative permeability of gas. The overall average error for all testing cases for the relative permeability of water was 1.34% with a minimum error of 0% and maximum error of 6.53%. The overall average error for the relative permeability of gas was 6.47% with a minimum error 0.65% and a maximum error of 42.03%.

Table 20. The average error results for the relative permeability data for all blind testing cases.

Overall Error Analysis		
Property	krw	krg
Average % Error	1.346129598	6.466009667
Min % Error for a Case	0	0.646511397
Max% Error for a Case	6.534470168	42.0285103

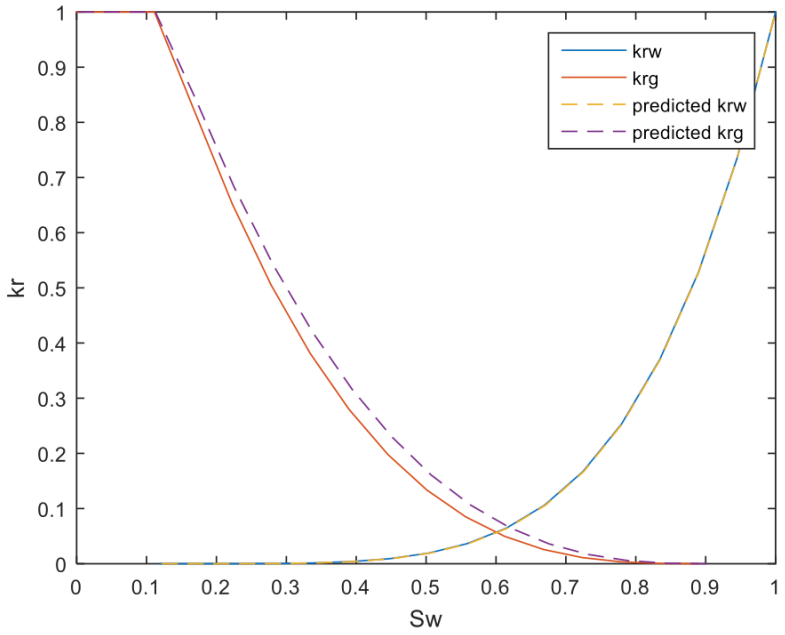


Figure 28. Blind test case 1 representative permeability curve comparing results from network prediction and simulation data (example of an average case).

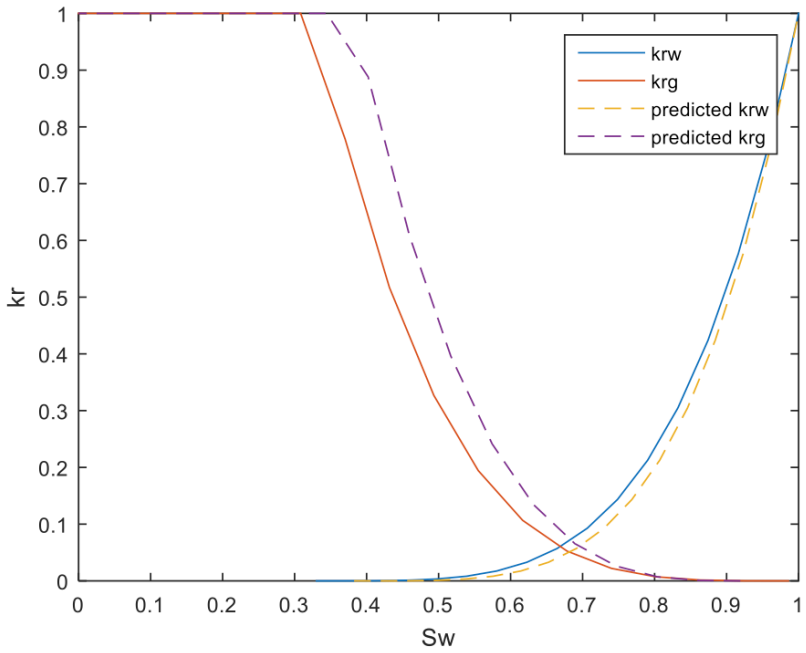


Figure 29. Blind test case 2 representative permeability curve comparing results from network prediction and simulation data (example of a worst case).

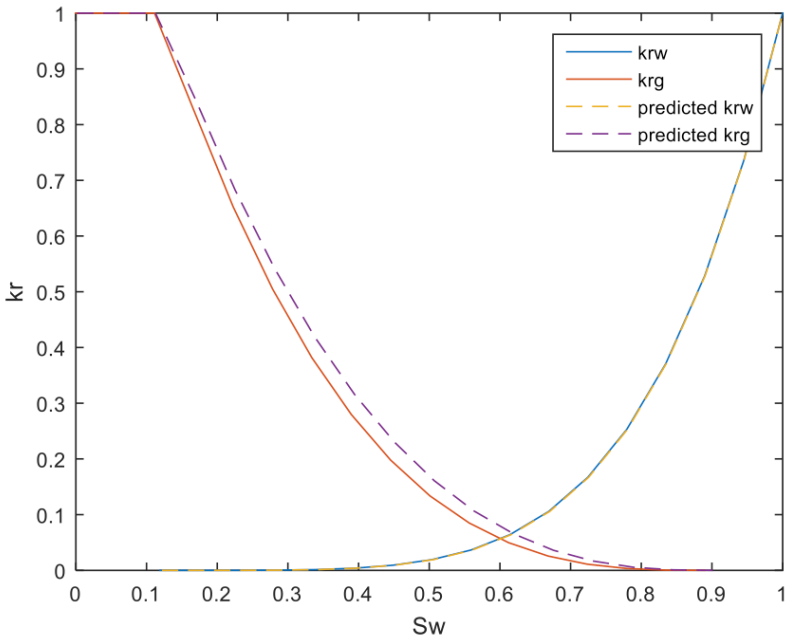


Figure 30. Blind test case 52 representative permeability curve comparing results from network prediction and simulation data (example of an average case).

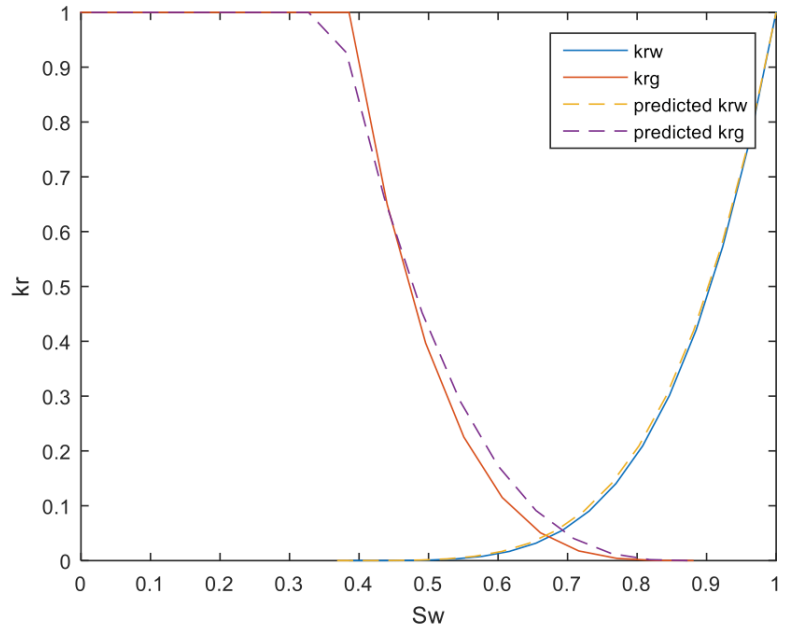


Figure 31. Blind test case 15 representative permeability curve comparing results from network prediction and simulation data (example of an average case).

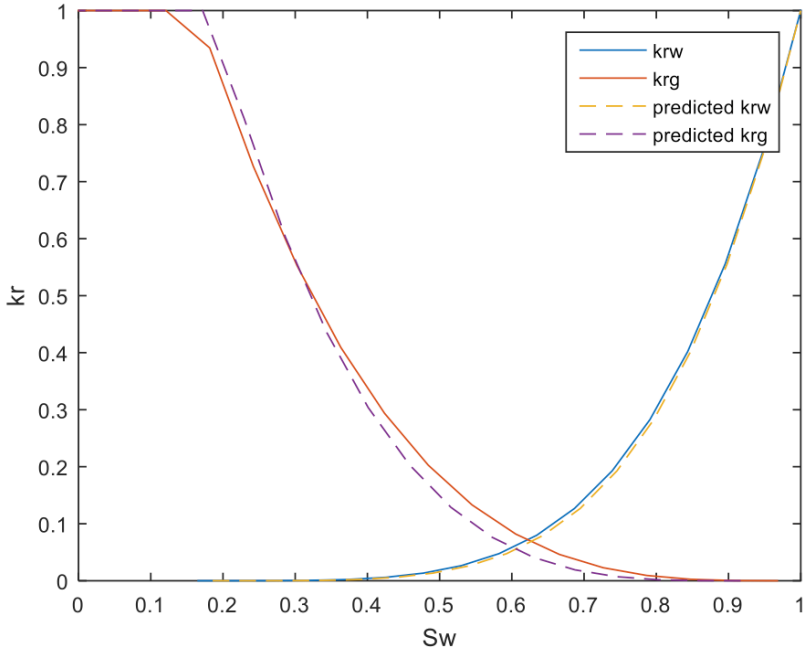


Figure 32. Blind test case 7 representative permeability curve comparing results from network prediction and simulation data (example of an average case).

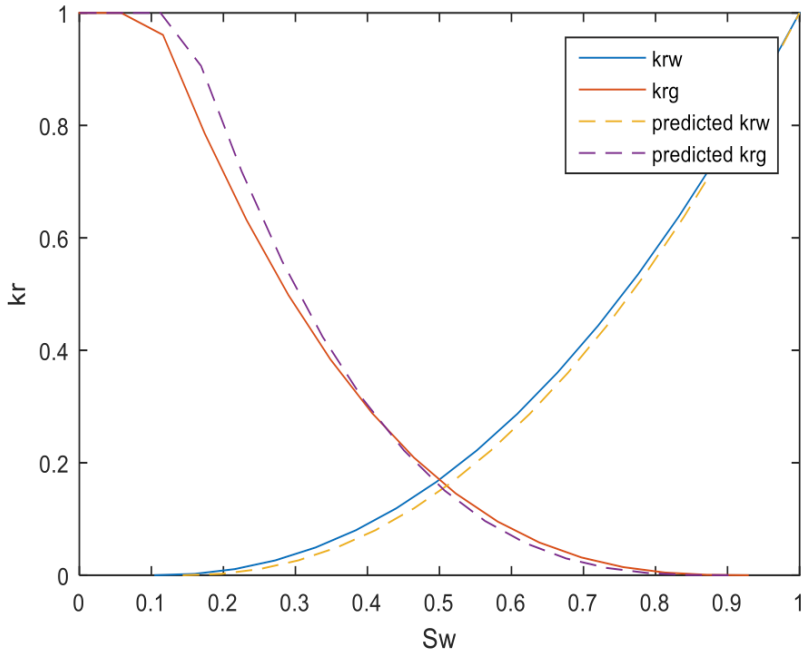


Figure 33. Blind test case 19 representative permeability curve comparing results from network prediction and simulation data (example of an average case).

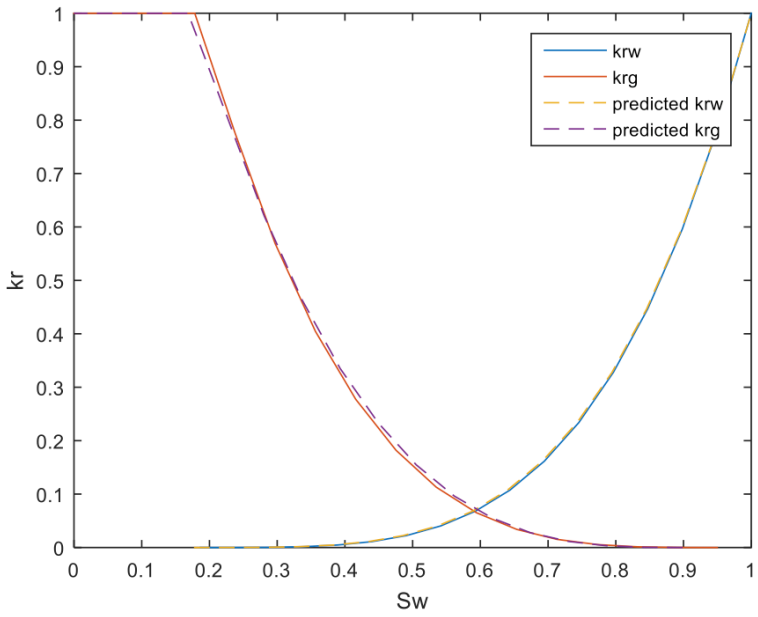


Figure 34. Blind test case 50 representative permeability curve comparing results from network prediction and simulation data (example of a best case).

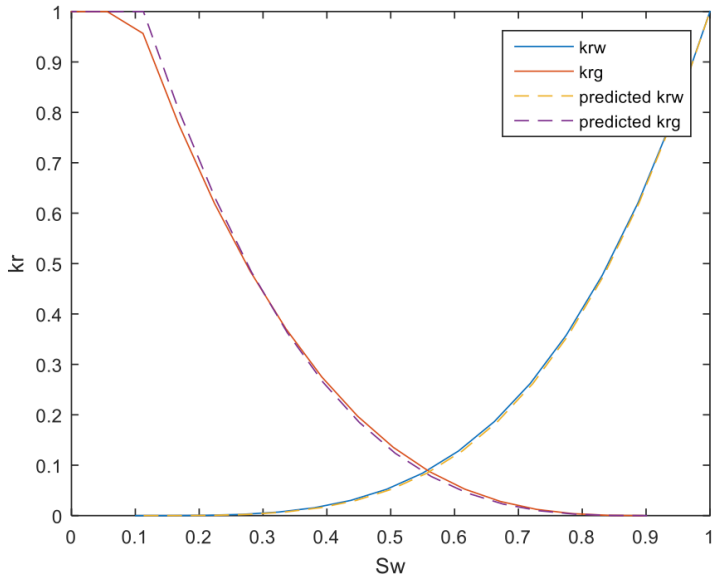


Figure 35. Blind test case 17 representative permeability curve comparing results from network prediction and simulation data (example of a best case).

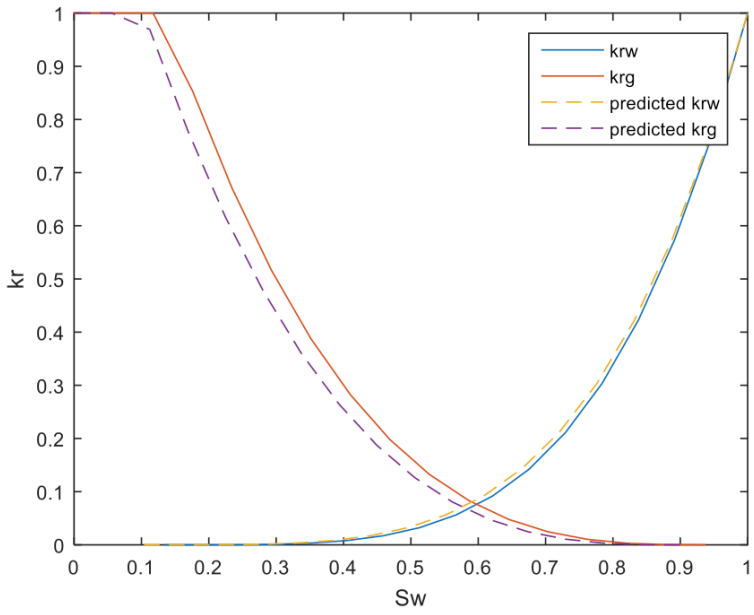


Figure 36. Blind test case 24 representative permeability curve comparing results from network prediction and simulation data (example of an average case).

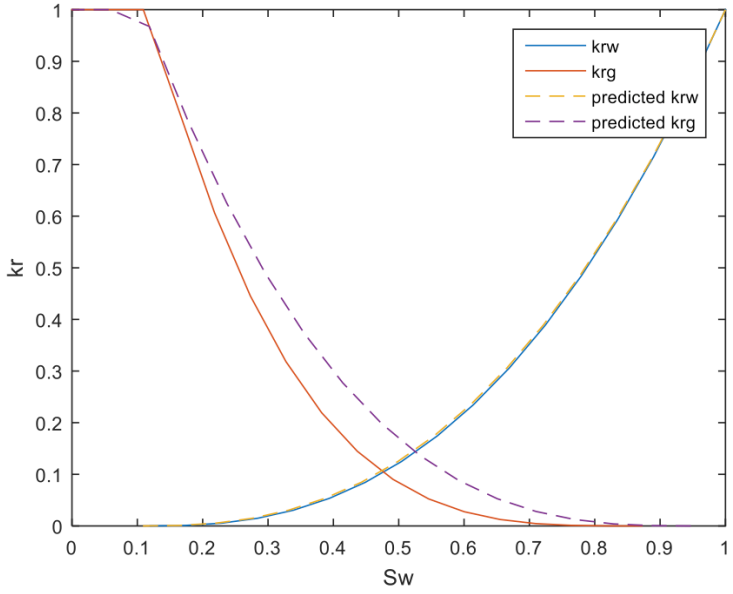


Figure 37. Blind test case 38 representative permeability curve comparing results from network prediction and simulation data (example of worst case).

By utilizing rate transient data, these permeability curves serve as representative permeability curves for the entire drainage area, due to the fact the production of the reservoir during the time production data is collected is dependent upon communication with the entire reservoir. In addition, the use of average reservoir properties instead of localized values, provides the network with data representative of the entire reservoir.

Chapter 6

Conclusions and Recommendations

Relative permeability serves as an important parameter in interpreting and understanding the flow of reservoir fluids. The most common ways of obtaining relative permeability data prove to be expensive, are limited in predictions to a localized region, and often times lead to a high degree of error in laboratory experiments. Artificial neural networks serve as a tool capable of learning functional relationships for complex problems. The main objective of this research was to develop an expert artificial neural network capable of producing representative permeability curves based upon Corey's relative permeability model for the entire drainage area of a specific reservoir given a set of inputs and outputs. A total of nine proxy models were developed in this study. There were three stages of development where three models were developed in each stage: a one-layer network, a two-layer network, and a three-layer network. During each stage, the model providing the most accurate representation of the Corey relative permeability model parameters was selected as the optimal model, however, it was not until stage three that a network was capable of reproducing the parameters within an error of $\pm 10\%$. The stages of development can be summarized as follows:

The first stage of development was based upon pressure transient data of a producing well for one month of production. The bests results from this stage were provided by a one-hidden layer network consisting of 23 neurons. The average error for all six of the Corey model parameters was 40.08% with the best results coming from the Corey exponent for gas and water at about 18% error. The worst results came from the critical gas saturation at over an average error of 100%.

The second stage of development was based upon rate transient data where daily gas rate and daily water rate were included in the input structure. This stage also changed the production time from one month to two years. Again, the best structure in this stage was a one-hidden layer network with 41 neurons. The network's performance improved in blind testing from 40.08% to 32.01%. The water related parameters of Corey's model produced the best results, while the gas parameters were predicted less precisely.

The third and final stage of development was also based upon rate transient data. This stage also included two functional links in an effort to help the network learn the functional relationship between the inputs and outputs. The networks in this stage also only considered data from simulation cases where at least a 50% or more decrease in gas saturation occurred. The optimum network for stage three was a three-hidden layer network consisting of a total of 485 neurons. The overall average error for the Corey parameters decreased from 32.01% to 4.38%. The best results were provided for the Corey exponent of water at 1.17%, while the worst results came from the critical gas saturation. The average error for this property was 9.81%.

The following section includes some specific conclusions derived from this research:

- It is important to provide a data set that includes a range of property values that pertains to the characteristics of several different reservoirs. This allows the ANNs to capture a universal relationship for gas-water systems rather than a regional relationship specific to one location.
- Using the conjugate gradient backpropagation with Polak-Ribiére updates training method with the symmetric saturating linear transfer function (*satlins*) as the functions in the hidden layers proved to be the best combination in the optimized expert system.
- The network structure heavily depends on the complexity of the problem and is unique to the amount of data and type of data for any given problem
- In this study, it was found that increasing the number hidden layers proved to improve the network's performance in the final stage of development. It appears that more hidden layers are required as the complexity of the problem increases.
- In this study, it was observed that the addition of functional links greatly improved the network's prediction capabilities.
- In this study, it was observed that providing cleaning mechanisms for the data improves the overall quality of the data and enables the network to appropriately capture the functional relationship between the inputs and Corey's relative permeability model parameters.
- In this study, it was observed that the network must be able to identify a significant change in the gas saturation during the production period of the well.

- In this study, it is evident that in all stages, the water parameters were predicted more closely to the actual values than the gas parameters. This results from the wider range of data for inputs controlling the gas parameters compared to the inputs controlling the water parameters. The restriction parameter placed upon the water production fails to include cases with higher water saturations and daily water production rates. Therefore, in comparison between the relative permeability of water and gas, the network has more difficulty accurately capturing the gas parameters resulting in less accurate relative permeability characteristics for gas.

  The following sections highlights some recommendations to further enhance this study through future actions:

  - The reservoir parameters in this study were homogenous throughout the reservoir. In reality, reservoir properties are dynamic and reservoirs are heterogeneous Implementing a reservoir model with interpolated, dynamic properties would enhance the overall accuracy and applicability of this expert ANN model.
  - Because the data sets were randomly generated and used to construct reservoir models, it is possible that some of the properties for certain cases contain a range of properties that do not make particular sense for a reservoir (i.e. for a given simulation case, some properties may apply to a shale reservoir while other property values may apply a carbonate or sandstone reservoir). In order to correct for this possibility, different reservoir models could be constructed for shale plays, carbonate plays, and sandstone plays making the simulation more accurate.
  - Future studies should expand the scope of this study to include gas-oil systems and three phase systems

- Future studies should consider implementing horizontal wells instead of vertical wells. A majority of wells drilled in today's industry are horizontal. Therefore, implementing horizontal wells increases the applicability and versatility of this tool for industry-wide use.
- This study only utilized one, two, and three hidden layer networks. Future studies should expand this study to include four, five, and hidden layer networks to see if these networks could further enhance the network's ability to capture the functional relationship between the inputs and Corey's relative permeability parameters.
- This study utilized a permeability versus porosity envelope to eliminate irrelevant cases of permeability and porosity (i.e. a case of low porosity, but high permeability). However, in reality this sort of case could exist in tight naturally, or hydraulically fractured reservoirs. Therefore, further studies should account for such cases and develop a network capable of developing relative permeability curves for fracture reservoirs.

Appendix A

Z-Factor Validation

MATLAB Code

```
%Author: Nathan Case
%z_factor_validation

pressureValues=30;
pressureRange=500:500:15000;
temp=300;
gamma=1.2;
zfactor=zeros(1,pressureValues);
for j=1:pressureValues
   if j>1
         [zfactor(j)]=fsolve(@(z)z_dranchuk(z,gamma,temp,pressureRange(j)),zfactor(j-1));
      else
         [zfactor(j)]=fsolve(@(z)z_dranchuk(z,gamma,temp,pressureRange(j)),1);
   end
end
```

Z-Factor Validation Run 1			
T=100F,γ_g=0.6			
Pressure (psi)	z-factor (Nathan)	z-factor (sheet)	% Diff.
500	0.9411	0.9413	0.0002
1000	0.8884	0.8887	0.0003
1500	0.8471	0.8475	0.0005
2000	0.8227	0.8231	0.0005
2500	0.8171	0.8175	0.0005
3000	0.8279	0.8283	0.0005
3500	0.8512	0.8516	0.0004
4000	0.8832	0.8835	0.0003
4500	0.921	0.9213	0.0003
5000	0.9629	0.9631	0.0002
5500	1.0076	1.0077	0.0001
6000	1.0541	1.0541	0.0000
6500	1.1018	1.1018	0.0000
7000	1.1504	1.1504	0.0000
7500	1.1995	1.1995	0.0000
8000	1.249	1.2489	0.0001
8500	1.2987	1.2986	0.0001
9000	1.3485	1.3483	0.0002
9500	1.3983	1.3981	0.0002
10000	1.448	1.4478	0.0001
10500	1.4977	1.4974	0.0002
11000	1.5473	1.5470	0.0002
11500	1.5967	1.5964	0.0002
12000	1.646	1.6456	0.0002
12500	1.6951	1.6947	0.0002
13000	1.7441	1.7437	0.0003
13500	1.7929	1.7924	0.0003
14000	1.8415	1.8410	0.0003
14500	1.8899	1.8894	0.0003
15000	1.9382	1.9376	0.0003

Z-Factor Validation Run 2			
T=300, γ_g=1.2			
Pressure (psi)	z-factor(Nathan)	z-factor (sheet)	% Diff.
500	0.9314	0.9315	0.0001
1000	0.8716	0.8718	0.0002
1500	0.8292	0.8295	0.0004
2000	0.8113	0.8116	0.0004
2500	0.8174	0.8177	0.0004
3000	0.8417	0.8420	0.0003
3500	0.8782	0.8784	0.0002
4000	0.9224	0.9226	0.0002
4500	0.9717	0.9718	0.0001
5000	1.0241	1.0242	0.0000
5500	1.0785	1.0785	0.0000
6000	1.1342	1.1342	0.0000
6500	1.1907	1.1906	0.0000
7000	1.2476	1.2476	0.0000
7500	1.3048	1.3047	0.0001
8000	1.3621	1.3619	0.0001
8500	1.4193	1.4192	0.0001
9000	1.4765	1.4763	0.0001
9500	1.5335	1.5333	0.0001
10000	1.5904	1.5901	0.0002
10500	1.647	1.6467	0.0002
11000	1.7035	1.7032	0.0002
11500	1.7597	1.7593	0.0002
12000	1.8156	1.8153	0.0002
12500	1.8714	1.8710	0.0002
13000	1.9268	1.9264	0.0002
13500	1.9821	1.9817	0.0002
14000	2.0371	2.0367	0.0002
14500	2.0919	2.0914	0.0002
15000	2.1464	2.1459	0.0002

Appendix B

Example of Simulator Input File for CMG from MATLAB Code

```
%Author: Nathan Case
%This code is used for the development of CMG input file for reservoir
%simulation. Will generate a batch file in the process

clc;
clear;

% Number of Radial Discretizations
nr = 30;

% Depth of the top of the formation (ft)
dTop = 1500;

% Number of Pressure Values in PVT Table
pvtRow = 16;

% Number of Saturation Values in Permeability Table
satRow = 17;

% Input Values
load indata.txt

% Print CMG File
batchFileName = ('imex_batch.bat');
fidbatch=fopen(batchFileName,'wt');

%Header of CMG File
for i=1:length(indata(1,:))
   dat_number=num2str(i);
   dat_name=['imex batch_' dat_number '.dat'];
   fid=fopen(dat_name,'wt');
   fprintf(fid,'RESULTS SIMULATOR IMEX 201410\n');
   fprintf(fid,'INUNIT FIELD\n');
   fprintf(fid,'WSRF WELL 1\n');
   fprintf(fid,'WSRF GRID TIME\n');
   fprintf(fid,'WSRF SECTOR TIME\n');
   fprintf(fid,'OUTSRF WELL LAYER NONE\n');
   fprintf(fid,'OUTSRF RES ALL\n');
   fprintf(fid,'OUTSRF GRID SO SG SW PRES OILPOT BPP SSPRES WINFLUX\n');
   fprintf(fid,'WPRN GRID 0\n');
   fprintf(fid,'OUTPRN GRID NONE\n');
```

```
fprintf(fid,'OUTPRN RES NONE\n');
fprintf(fid,'RESULTS XOFFSET          0.0000\n');
fprintf(fid,'RESULTS YOFFSET          0.0000\n');
fprintf(fid,'RESULTS ROTATION          0.0000  **$  (DEGREES)\n');
fprintf(fid,'RESULTS AXES-DIRECTIONS 1.0 -1.0 1.0\n');
fprintf(fid,'**$ *****************************************\n');
fprintf(fid,'**$ Definition of fundamental cylindrical grid\n');
fprintf(fid,'**$ *****************************************\n');
fprintf(fid,'GRID RADIAL 30 1 1 *RW        %d\n',indata(10,i));
fprintf(fid,'KDIR DOWN\n');
fprintf(fid,'DI IVAR   ');
rWidthPlus=zeros(1,nr);
rWidthMinus=zeros(1,nr);
alphaLg=(indata(1,i)/indata(10,i))^(1/nr);
% radius of the center of the gridblocks (ft)
rNode= (alphaLg*log(alphaLg)/(alphaLg-1))*indata(10,i);
for j=2:nr
   rNode(j)=rNode(1)*(alphaLg^(j-1));
end
%r+1/2
for j=2:nr-1
rWidthPlus(j)=(rNode(j+1)-rNode(j))/(log((rNode(j+1))/rNode(j)));
end
rWidthPlus(30)=indata(1,i);
%r-1/2
for j=2:nr
   rWidthMinus(j)=(rNode(j)-rNode(j-1))/(log((rNode(j))/rNode(j-1)));
end
fprintf(fid, '%d ',indata(10,i)); %establishes the beginning of the radial system at the radius of
the wellbore
for j=2:nr
   fprintf(fid,'%d ',rWidthPlus(j));
end
fprintf(fid,'\n');
fprintf(fid, 'DJ JVAR  360\n');
fprintf(fid, 'DK ALL\n');
fprintf(fid, '%d*%d\n',nr,indata(2,i));
fprintf(fid, 'DTOP\n');
fprintf(fid, '%d*%d\n',nr,dTop);
fprintf(fid, 'NETPAY CON       %d\n', indata(2,i));
fprintf(fid,'NULL CON           1\n');
fprintf(fid,'**Property: Porosity  %d\n',indata(3,i));
fprintf(fid,'POR CON         %d\n',indata(3,i));
fprintf(fid,'**$ Property: Permeability I (md)  %d\n',indata(7,i));
fprintf(fid,'PERMI CON          %d\n',indata(7,i));
fprintf(fid,'**$ Property: Permeability J (md)  %d\n',indata(7,i));
```

```
fprintf(fid,'PERMJ CON          %d\n',indata(7,i));
fprintf(fid,'**$ Property: Permeability K (md)  %d\n',indata(6,i));
fprintf(fid,'PERMK CON          %d\n',indata(6,i));
fprintf(fid,'PINCHOUTARRAY CON          1\n');
fprintf(fid,'CPOR %d\n',indata(9,i));
fprintf(fid,'MODEL GASWATER\n');
fprintf(fid,'TRES %d\n',indata(5,i));
fprintf(fid,'PVTG EG 1\n');
fprintf(fid,'**$    p        Eg          visg\n');
pPVT=(indata(4,i)-14.696)/(pvtRow-1)*(0:(pvtRow-1))+14.696;
Eg=zeros(1,pvtRow);
visG=zeros(1,pvtRow);
zfactor=zeros(1,pvtRow);
densityGas=zeros(1,pvtRow);

%for calculating the Gas Expansion Factor (Eg) and Viscosity of Gas
%(visG)
% Required in viscosity calulations (Using Lee Method)
tempRan=indata(5,i)+459.67;
      mwG=indata(8,i)*28.97; %gas gravity multiplied by molecular weight of air
      K=((0.00094+(mwG*2e-06))*((tempRan)^1.5))/(209+(19*mwG)+tempRan);
      X=3.5+(986/tempRan)+(mwG*0.01);
      Y=2.4-(0.2*X);

for j=1:pvtRow
   fprintf(fid,'%d  ',pPVT(j));
   if j>1
      [zfactor(j)]=fsolve(@(z)z_dranchuk(z,indata(8,i),tempRan,pPVT(j)),zfactor(j-1));
   else
      [zfactor(j)]=fsolve(@(z)z_dranchuk(z,indata(8,i),tempRan,pPVT(j)),1);
   end

   %Eg calculation
   Eg(j)=198.6*(pPVT(j))/(zfactor(j)*tempRan);
   fprintf(fid,'%d    ',Eg(j));

   %Viscosity Calculation (Using Lee Method)
      densityGas(j)=0.00149406*((pPVT(j)*mwG)/(zfactor(j)*(tempRan)));
      visG(j)=K*exp(X*(densityGas(j)^Y));
      fprintf(fid,'%d    \n',visG(j));
end
fprintf(fid,'GRAVITY GAS %d\n',indata(8,i));
fprintf(fid,'REFPW 14.696\n');
fprintf(fid,'DENSITY WATER 62.1797\n');
fprintf(fid,'BWI 1.01056\n');
fprintf(fid,'CW 3.15633e-006\n');
```

```
fprintf(fid,'VWI 0.62582\n');
fprintf(fid,'CVW 0.0\n');

%Rock Fluid Properties (permeability and saturation relations)
fprintf(fid,'ROCKFLUID\n');
fprintf(fid,'RPT 1\n');
fprintf(fid,'**$     Sw     krw\n');
fprintf(fid,'SWT\n');
fprintf(fid,' SMOOTHEND OFF\n');

%Process using Sw
sw=0:0.0625:1;
for j=1:satRow
   krw(j) = indata(15,i)*(((sw(j)-indata(12,i))/(1-indata(12,i)))^indata(18,i));
   if sw(j)-indata(12,i)<0
      krw(j)=0;
   end
end


%Process using Sg
sg=0:0.0625:1;
for j=1:satRow
   krg(j)=indata(16,i)*(((sg(j)-indata(14,i))/(1-indata(14,i)-indata(11,i)))^indata(17,i));
   if sg(j)-indata(14,i)<0
      krg(j)=0;
   end
end


%Checking endpoints to ensure that values do not exceed 1
for j=1:length(krw)
   if krw(j)>1
     krw(j)=1;
   end
end
for j=1:length(krg)
   if krg(j)>1
     krg(j)=1;
   end
end


for j=1:satRow
   fprintf(fid,'%d ',sw(j));
   fprintf(fid,'%d\n ',krw(j));
```

```
end

fprintf(fid,'**$    Sg    krg\n');
fprintf(fid,'SGT\n');

for j=1:satRow;
   fprintf(fid,'%d ',sg(j));
   fprintf(fid,'%d\n ',krg(j));
end


%Initial Conditions
fprintf(fid,'INITIAL\n');
fprintf(fid,'VERTICAL DEPTH_AVE WATER_GAS EQUIL NOTRANZONE\n\n');
fprintf(fid,'REFDEPTH   %d\n',dTop+((indata(2,i))/2));
fprintf(fid,'REFPRES   %d\n',indata(4,i));
fprintf(fid,'DWGC     %d\n',dTop+indata(2,i));
fprintf(fid,'**$ Property: Water Saturation\n');
fprintf(fid,'SW CON    %d\n',indata(20,i));
fprintf(fid,'**$ Property: Initial Water Saturation\n');
fprintf(fid,'SWINIT CON   %d\n',indata(11,i));
%Numerical Conditions
fprintf(fid,'NUMERICAL\n');
fprintf(fid,'DTMIN  1e-010\n');
fprintf(fid,'MAXSTEPS 1000000\n');
fprintf(fid,'RUN\n');
fprintf(fid,'DATE 2015 1 1\n');
fprintf(fid,'DTWELL 1e-009\n');

%Well Info
fprintf(fid,'WELL   "Well-1"\n');
fprintf(fid,'PRODUCER  "Well-1"\n');
fprintf(fid,'OPERATE MAX STG %d CONT\n', indata(19,i));
fprintf(fid,'**$        rad  geofac  wfrac  skin\n');
fprintf(fid,'GEOMETRY K  0.25 0.37     1.    0.\n');
fprintf(fid,'PERF  GEOA  "Well-1"\n');
fprintf(fid,'**$ UBA   ff  Status   Connection\n');
fprintf(fid,'   1 1 1  1.  OPEN    FLOW-TO  "SURFACE"\n');

%output 24 points per day for the first day

outputTime1=1+.041666*(1:24);
for j=1:24
   fprintf(fid,'DATE 2015 1 %d\n',outputTime1(j));
end
```

```
%output 8 points per day for days 2,3,and 4
outputTime2=2+.125*(1:8);
for j=1:8
   fprintf(fid,'DATE 2015 1 %d\n',outputTime2(j));
end
outputTime3=3+.125*(1:8);
for j=1:8
   fprintf(fid,'DATE 2015 1 %d\n',outputTime3(j));
end
outputTime4=4+.125*(1:8);
for j=1:8
   fprintf(fid,'DATE 2015 1 %d\n',outputTime4(j));
end

%output 4 points per day for days 5-10
outputTime5=5+.25*(1:4);
for j=1:4
   fprintf(fid,'DATE 2015 1 %d\n',outputTime5(j));
end
   outputTime6=6+.25*(1:4);
for j=1:4
   fprintf(fid,'DATE 2015 1 %d\n',outputTime6(j));
end
outputTime7=7+.25*(1:4);
for j=1:4
   fprintf(fid,'DATE 2015 1 %d\n',outputTime7(j));
end
   outputTime8=8+.25*(1:4);
for j=1:4
   fprintf(fid,'DATE 2015 1 %d\n',outputTime8(j));
end
   outputTime9=9+.25*(1:4);
for j=1:4
   fprintf(fid,'DATE 2015 1 %d\n',outputTime9(j));
end
% output 3 points per day for days 10-15
   outputTime10=10+.3333*(1:3);
for j=1:3
   fprintf(fid,'DATE 2015 1 %d\n',outputTime10(j));
end
   outputTime11=11+.3333*(1:3);
for j=1:3
   fprintf(fid,'DATE 2015 1 %d\n',outputTime11(j));
end
   outputTime12=12+.3333*(1:3);
for j=1:3
```

```
    fprintf(fid,'DATE 2015 1 %d\n',outputTime12(j));
  end
    outputTime13=13+.3333*(1:3);
  for j=1:3
    fprintf(fid,'DATE 2015 1 %d\n',outputTime13(j));
  end
    outputTime14=14+.3333*(1:3);
  for j=1:3
    fprintf(fid,'DATE 2015 1 %d\n',outputTime14(j));
  end

  %Output 1 point per day from days 15-31
  fprintf(fid,'DATE 2015 1 16.0000\n');
  fprintf(fid,'DATE 2015 1 17.0000\n');
  fprintf(fid,'DATE 2015 1 18.0000\n');
  fprintf(fid,'DATE 2015 1 19.0000\n');
  fprintf(fid,'DATE 2015 1 20.0000\n');
  fprintf(fid,'DATE 2015 1 21.0000\n');
  fprintf(fid,'DATE 2015 1 22.0000\n');
  fprintf(fid,'DATE 2015 1 23.0000\n');
  fprintf(fid,'DATE 2015 1 24.0000\n');
  fprintf(fid,'DATE 2015 1 25.0000\n');
  fprintf(fid,'DATE 2015 1 26.0000\n');
  fprintf(fid,'DATE 2015 1 27.0000\n');
  fprintf(fid,'DATE 2015 1 28.0000\n');
  fprintf(fid,'DATE 2015 1 29.0000\n');
  fprintf(fid,'DATE 2015 1 30.0000\n');
  fprintf(fid,'DATE 2015 1 31.0000\n');
  fprintf(fid,'*STOP');
  fprintf(fid,'\n**$');
  fprintf(fid,'\n**$');
  fclose(fid);
  fprintf(fidbatch,'%s','call "C:\Program Files
(x86)\CMG\IMEX\2014.10\Win_x64\EXE\mx201410.exe" -f "imex batch_');
  fprintf(fidbatch,num2str(i));
  fprintf(fidbatch,'.dat"\n');
end
fclose(fidbatch);
```

Appendix C
Example of 3-Hidden Layer ANN MATLAB Code

```
clc;
clear;
%% Input
load input.txt;
load output.txt;

input=input(:,1:3000);
output=output(:,1:3000);

%% 3-layer
n=5000;
%upper limit of number of neurons in one layer
neuron_no_upper_limit=500;
NNeu=zeros(n,3);
NNeu(:,1)=unidrnd(neuron_no_upper_limit,n,1);
NNeu(:,2)=unidrnd(neuron_no_upper_limit,n,1);
NNeu(:,3)=unidrnd(neuron_no_upper_limit,n,1);
transfer_cell=cell(n,3);
transfer_function_index=randint(n,1,5);
for i=1:n
   for j=1:3
      switch transfer_function_index(i,j)
          case 0
             transfer_cell{i,j}='logsig';
          case 1
             transfer_cell{i,j}='tansig';
          case 2
             transfer_cell{i,j}='radbas';
          case 3
             transfer_cell{i,j}='satlin';
          case 4
             transfer_cell{i,j}='satlins';
      end
   end
end
%Training function initialization
 train_cell=cell(n,1);
 train_function_index=randint(n,1,4);
  for i=1:n
       switch train_function_index(i)
          case 0
```

```
            train_cell{i}='trainoss';
        case 1
            train_cell{i}='traincgp';
        case 2
            train_cell{i}='traincgb';
        case 3
            train_cell{i}='traingd';
    end
 end
%Train ANNs
min_error=1/eps;
 for i=1:n
   [min_error] = training( NNeu(i,:),{transfer_cell{i,:}},input,output,min_error, train_cell{i});
 end
```

BIBLIOGRAPHY

(1.) Al-Fattah, S.M, and Al-Naim, H.A. 2009. Artificial-Intelligence Technology Predicts Relative Permeability of Giant Carbonate Reservoirs. SPE Reservoir Evaluation & Engineering. 2 (4): 96-103. SPE-109018-PA.

(2.) Almousa, T. 2013. Development and Utilization of Integrated Artificial Expert Systems for Designing Multi-lateral Well Configurations, Estimating Reservoir Properties and Forecasting Reservoir Performance. Dissertation, The Pennsylvania State University, University Park, Pennsylvania (24 June 2013).

(3). Beale, M.H., Hagan, M.T., & Demuth, H.B. 2014. Neural Network Toolbox: User's Guide R2014a. Natick, MA: The MathWorks Inc.

(4.) Bradley, Howard B., Gipson, Fred W. 1987. Petroleum Engineering Handbook. Richardson, TX, U.S.A: Society of Petroleum Engineers.

(5.) Chaudhary, Anish Singh (2011). Shale Oil Production Performance from a Stimulated Reservoir Volume. Thesis, Texas A&M University, College Station, Texas (22 October 2012).

(6.) Chidambaram, P. 2009. Development and Testing of an Artificial Neural Network Based History Matching Protocol to Characterize Reservoir Properties. Dissertation, The Pennsylvania State University, University Park, Pennsylvania (2 February 2009).

(7.) Cladbaugh, Caroline, Myszewski, Dave, & Pang, Jimmy. 2000. Neural Networks. Stanford University. https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/index.html. (accessed 25 January 2017).

(8.) Commonwealth of Pennsylvania. 2013. Marcellus Shale Development. Pennsylvania Department of Environmental Protection. October 2013.

(9.) Corey, A.T. 1954. Interrelation Between Gas and Oil Relative Permeabilities. Producers Monthly.

(10.) Crain, E.R. 2015. Compressibility of Rocks. Crain's Petrophysical Handbook. https://www.spec2000.net/09-compress.htm. (accessed 11 June 2015).

(11.) Dandekar, Abhijit Y. 2006. Petroleum Reservoir Rock and Fluid Properties. Boca Raton, FL: CRC/Taylor & Francis.

(12.) Dewitt, Hank. 2008. Marcellus Shale Overview. Chesapeake Energy Eastern Division. http://thefriendsvillegroup.com/2008_Investor_and_Analyst_Meeting-Marcellus.pdf. (accessed 18 June 2015).

(13.) Dranchuk, P.M, and Abou-Kassem, J.H. 1975. Calculation of Z Factors for Natural Gases Using Equations of State. The Journal of Canadian Petroleum. (July-September 1975): 34-36.

(14.) Effendi, H.S. 2011. Relative Permeability Curves Using Corey Method. Petroleum Support. http://petroleumsupport.com/relative-permeability-curves-using-corey-method (accessed 30 July 2015).

(15.) Esmaili, S. Mohagegh, S.D. 2014. Full Field Reservoir Modeling of Shale Assets Using Advanced Data-Driven Analytics. Geoscience Frontiers. http://shahab.pe.wvu.edu/Publications/Pdfs/GF2015.pdf (accessed 18 June 2015).

(16.) Fast Welltest. 2012. Formation Compressibility (cf). Feteke Associates Inc. http://www.fekete.com/SAN/TheoryAndEquations/WellTestTheoryEquations/Formation_Compressibility.htm. (accessed 11 June 2015).

(17.) Fetter, C.W., 1994. *Applied Hydrogeology*, 3rd ed., Macmillan Publishing, New York, NY.

(18.) Guler, B., Ertekin, T., and Grader, A.S. 2003. An Artificial Neural Network Based Relative Permeability Predictor. Journal of Canadian Petroleum Technology. 42 (4): 49-57.

(19.) Katz, Donald L. 1959. Introduction to the Natural Gas Industry. In Handbook of Natural Gas Engineering. 27-32. New York: McGraw-Hill.

(20.) Katz, Donald L. 1959. "Properties of Reservoir Rock. In Handbook of Natural Gas Engineering. 42. New York: McGraw-Hill.

(21.) KED Interests LLC. 2015. Eagle Ford Shale Geology. Eagle Ford Shale. http://eaglefordshale.com/geology/. (accessed 18 June 2015).

(22.) King, Hobart M. Ph.D. 2005-2015. Utica Shale-The Natural Gas Giant below the Marcellus. Geology.com. http://geology.com/articles/utica-shale/. (accessed 23 June 2015).

(23.) Lee, A.L, Gonzalez, M.H, Aime, J.M. et al. 1966. The Viscosity of Natural Gases. Journal of Petroleum Technology. (August 1966): 997-1000

(24.) Lyons, William C., and Joseph Zaba. 1996. Reservoir Engineering: Basic Principles, Definitions, and Data. In Standard Handbook of Petroleum & Natural Gas Engineering, ed. William C. Lyons and Gary J. Plisga. 29-30+. Houston, TX: Gulf Pub.

(25.) Marcellus-Shale.us. 2009-2015. Marcellus Shale Gas Production. Marcellus-Shale.us. http://www.marcellus-shale.us/Marcellus-production.htm. (accessed 23 June 2015).

(26.) Onur, Mustafa. 2007. PET467E A Note on Rock Compressibility. Istanbul Teknik Universitesi. Spring 2007. http://web.itu.edu.tr/~palabiyiky/PET467E_A_Note_on_Rock_Compressibility.pdf (accessed 11 June 2015).

(27.) Saturation and Relative Permeability End-Points Scaling. 2013. Petrofaq. 29 January 2013. http://petrofaq.org/wiki/Saturation_and_relative_permeability_end-points_scaling. (accessed 22 January 2017).

(28.) Reingold, Eyal. History of Neural Networks. Artificial Neural Networks Technology. The University of Toronto. http://www.psych.utoronto.ca/users/reingold/courses/ai/cache/neural4.html. (accessed 25 January 2017).

(29.) Santizábal, H. and Perez-Uribe, A. 2013. TORCS, Supervised Learning. March 2013. http://ape.iict.ch/teaching/AIGS/AIGS_Labo/Labo4-Supervised. (accessed 4 August 2015).

(30.) Silpngarmlers. N., Guler, B., Ertekin, T. et al. 2002. Development and Testing of Two-Phase Relative Permeability Predictors Using Artificial Neural Networks. SPEJ 7 (3): 299-308. SPE-79547-PA. DOI: 10.2118/79547-PA.

(31.) Relative Permeability 2016. PetroWiki. SPE International. 19 January 2016. http://petrowiki.org/Relative_permeability. (accessed 21 January 2017).

(32.) Dry Gas Reservoirs. 2014. PetroWiki. SPE International. 17 September 2014. http://petrowiki.org/Dry_gas_reservoirs. (accessed 17 June 2015).

(33.) Petroleum Fluids and Their Characteristics. 2013. PetroWiki. SPE International. 27 March 2013. http://petrowiki.org/File%3AVol5_Page_0897_Image_0001.png. (accessed 11 June 2013).

(34.) Suzuki, K. 2011. Artificial Neural Network – Methodological Advances in Biomedical Applications. Rijeka, Croatia. InTech.

(35.) U.S Energy Information Administration. 2015. Drilling Productivity Report for Key Tight Oil and Shale Gas Regions. U.S Energy Information Administration Independent Statistics & Analysis. June 2015. http://www.eia.gov/petroleum/drilling/pdf/dpr-full.pdf. (accessed 23 June 2015.)

(36.) Wang, Xiuli, and Michael J. Economides. 2009. Natural Gas Basics. In Advanced Natural Gas Engineering. Chap. 1. 4-6. Houston, TX: Gulf Pub.

(37.) Yegnanarayana, B. 1999. Artificial Neural Networks. New Delhi, India: Prentice-Hall of India.

(38.) Zimmerman, Robert Wayne. 1991. Part One: Compressibility and Stress. In Compressibility of Sandstones. Amsterdam: Elsevier.

*Note many of the sources listed in the above reference section were used to select the acceptable range of reservoir parameter values.

ACADEMIC VITA

Academic Vita of Nathan J. Case
njc5215@psu.edu
303 Fraternity Row State College, PA 16801

EDUCATION: THE PENNSYLVANIA STATE UNIVERSITY, College of Earth and Mineral Sciences
Graduation: *May 2017*
Major: *Bachelor of Science in Petroleum and Natural Gas Engineering*
Minor: *Energy, Business, and Finance*
Schreyer Honors College Scholar | Dean's List: 7 semesters

EXPERIENCE:

EAST MANAGEMENT SERVICES/JKLM ENERGY; Sewickley, PA
Petroleum Engineering Intern **May 2016 – August 2016**
- Conducted engineering studies on Marcellus and Utica wells through the analysis of well logs and the collection/organization of completions and production data in the region
- Obtained rig experience in observing and participating in different drilling and completion operations (drilling, well logging, casing, tubing, cementing, fracking)
- Expanded knowledge in various aspects of the oil/gas industry working and developing relationships with employees in the geology, land management, and law departments

SHELL DRILLING & PRODUCTION CAMP; Robert, LA **May 2016**
- One of 40+ students from various universities selected to attend weeklong instructional camp with deep water focus on drilling, production, and HUET safety practices
- Attended professional presentations regarding different disciplines within Shell

PENN STATE UNIVERSITY DEPARTMENT OF ENERGY & MINERAL ENGINEERING
Undergraduate Research Internship May – August 2015
- Developed artificial neural network to predict representative relative permeability curves using production parameters and reservoir properties for dry gas reservoirs
- Established proficiency in MATLAB Neural Network Toolbox and CMG Reservoir Simulation and presented project to Department Head using SPE guidelines
- Expanded project into Undergraduate Thesis for Schreyer Honors College requirements

WEGMANS FOOD MARKETS; Allentown, PA **June 2013 – May 2015**
- Acquired praise for demonstrating leadership and the ability to work well with co-workers of diverse backgrounds at #1 rated supermarket in the U.S.

LEADERSHIP:

SIGMA PI FRATERNITY
President **November 2015-December 2016**
- Managed fraternity of 100+ brothers and oversaw daily operations of 43 in house brothers
- Responsible for social event risk management and for community outreach/relations
- Organized campaign to lead house through largest renovation in chapter's 105-year history
- Created IFC/National award winning philanthropy benefitting the Amazing Day Foundation

Chapter Accreditation Chair **Fall 2014-Fall 2015**
- Completed all fraternity documents that were submitted to both the IFC and our national governing board ensuring chapter's compliance to requirements and rules

THON Chair **Spring 2015-Spring 2016**
- Collaborated with other THON chairs in the organization of fundraising activities to achieve our goal of over $150,000 contributed to Four Diamonds at Hershey Medical
- Developed online database to send monthly newsletters to almost 2,000 contacts

SIGMA PI UNIVERSITY; Cumberland, Tennessee **July 2015**
- Selected to attend 5-day leadership conference sponsored by Undergraduate Inter Fraternity Institute to educate chapter leaders and complete community service projects

ACTIVITIES:

Phi Kappa Phi Honor Society	**Spring 2016-Present**
Society of Petroleum Engineers	**Fall 2013-Present**
Penn State Cello Choir	**Fall 2013-Present**

AWARDS:

Chevron Corporation Petroleum and Natural Gas Engineering Scholarship (2015 &2016), George L. Ellis Scholarship (2015 &2016), John and Willie Leone Family Undergraduate Scholarship (2016), Sigma Pi Academic Foundation Scholarship 2016, C. Jake Phipps Scholarship 2014, President's Freshman Award (4.0 GPA) Spring 2014, Lion's Club International Scholarship 2013