SANDIA REPORT

SAND2003-2950
Unlimited Release
Printed June 2004

Evolution of Neural Networks for the Prediction of Hydraulic Conductivity as a Function of Borehole Geophysical Logs: Shobasama Site, Japan

Paul C. Reeves and Sean A. McKenna

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2003-2950
Unlimited Release
Printed June 2004

Evolution of Neural Networks for the Prediction of Hydraulic Conductivity as a Function of Borehole Geophysical Logs: Shobasama Site, Japan

Paul C. Reeves
Data Exploitation Department

Sean A. McKenna
Geohydrology Department

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0735

Abstract

This report describes the methodology and results of a project to develop a neural network for the prediction of the measured hydraulic conductivity or transmissivity in a series of boreholes at the Tono, Japan study site. Geophysical measurements were used as the input to a feed-forward neural network. A simple genetic algorithm was used to evolve the architecture and parameters of the neural network in conjunction with an optimal subset of geophysical measurements for the prediction of hydraulic conductivity.

The first attempt was focused on the estimation of the class of the hydraulic conductivity, high, medium or low, from the geophysical logs. This estimation was done while using the genetic algorithm to simultaneously determine which geophysical logs were the most important and optimizing the architecture of the neural network. Initial results showed that certain geophysical logs provided more information than others- most notably the "short-normal", micro-resistivity, porosity and sonic logs provided the most information on hydraulic conductivity. The neural network produced excellent training results with accuracy of 90 percent or greater, but was unable to produce accurate predictions of the hydraulic conductivity class.

The second attempt at prediction was done using a new methodology and a modified data set. The new methodology builds on the results of the first attempts at prediction by limiting the choices of geophysical logs to only those that provide significant information. Additionally, this second attempt uses a modified data set and predicts transmissivity instead of hydraulic conductivity. Results of these simulations indicate that the most informative geophysical measurements for the

prediction of transmissivity are depth and sonic log. The long normal resistivity and self potential borehole logs are moderately informative. In addition, it was found that porosity and crack counts (clear, open, or hairline) do not inform predictions of hydraulic transmissivity.

Acknowledgements

This work was done under funding from the Japanese Nuclear Cycle Development Institute (JNC) and the authors are indebted to Mr. Nakano, Mr. Takeuchi and Mr. Saegusa of the JNC Tono Geoscience Center for initiating the funding and for support and guidance during this project. Reviews of this report by Hirotaka Saito and Ray Finley improved the final version.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94-AL-85000

Table of Contents

Table of Figures

Table of Tables

Introduction

The goal of this project was to develop a simulation tool capable of predicting hydraulic conductivity, or transmissivity, from a set of geophysical measurements. Such a tool could help minimize costs associated with site characterization through the identification of informative surrogate measures that could replace expensive hydraulic tests. It is recognized that there will always be a need for down-hole hydraulic tests, but it might be possible to reduce the number of tests necessary, or to better locate the tests that are taken by using a predictive neural network.

Feed-forward neural networks are capable of performing high-fidelity classification and function approximation, among other things (e.g., *Principe et al.*, [2000].). We attempted to develop a multi-layer perceptron (MLP), a form of feed-forward neural network that could perform hydraulic conductivity prediction using borehole geophysical log measurements as input features for the network.

In the past fifteen years, a number of applications of neural networks to borehole geophysical problems have been reported. Most of these applications have been focused on the estimation of facies types from geophysical logs. For example, *Baldwin et al.* [1989 and 1990] used a neural network to identify eight different lithofacies in a sedimentary sequence from eight different geophysical log measurements. Additional work on the identification of lithofacies in a sedimentary sequence was reported by *Rogers et al.* [1992] who used three geophysical logs: gamma, neutron and density, to identify the occurrence of four different lithofacies.

A more difficult problem than that of facies identification is that of permeability estimation. More recently, neural networks have also been applied to the problem of estimating permeability from geophysical log measurements. *Mohaghegh et al.* [1996] used three log responses as well as X,Y and depth coordinates and geological interpretation to estimate downhole permeability in a set of wells within a sandstone gas reservoir. The results of the permeabilities estimated with the neural network agreed favorably with the permeabilities measured on core-plug samples within the wells. *Wong and Shibli* [1998] applied an "interpolation neural network" to estimate permeabilities measured on core-plugs in a petroleum reservoir. They used seven geophysical logs and a facies description as input to the neural network. The results reported by *Wong and Shibli* [1998] indicate that the neural network was able to reproduce the measured permeability values and, most importantly, accurately predict the highest of the measured permeability values.

In the current work, the problem of estimating hydraulic conductivity, or transmissivity, in a fractured rock is attempted. This is a different problem from the previous work mentioned above in that the scale of the hydraulic conductivity or transmissivity measurement (meters) is very different from that of the geophysical logs (centimeters). Additionally, the accurate measurement and estimation of hydraulic conductivity or transmissivity in fractured rocks using borehole geophysical logs is very difficult, and to the authors' knowledge has not yet been reported.

Neural Networks

Multilayer perceptrons (MLPs), a form of feed forward neural networks, are useful for predicting classification or for performing functional approximation. Figure 1 depicts a two hidden-layer MLP that has been configured to predict three classes of events (note the three outputs) as a function of four measured properties (features). Their calculations are relatively simple. Values are fed into the input layer and these are multiplied by weights. The resulting values are fed into the next layer where "perceptrons" are located at each node in the network. At each perceptron, all inputs are summed and put through a rescaling function (typically a logistic function, sigmoid function, or hyperbolic tangent function). This continues through to the output layer. The perceptron at the output with the largest value corresponds to the predicted class of the event.

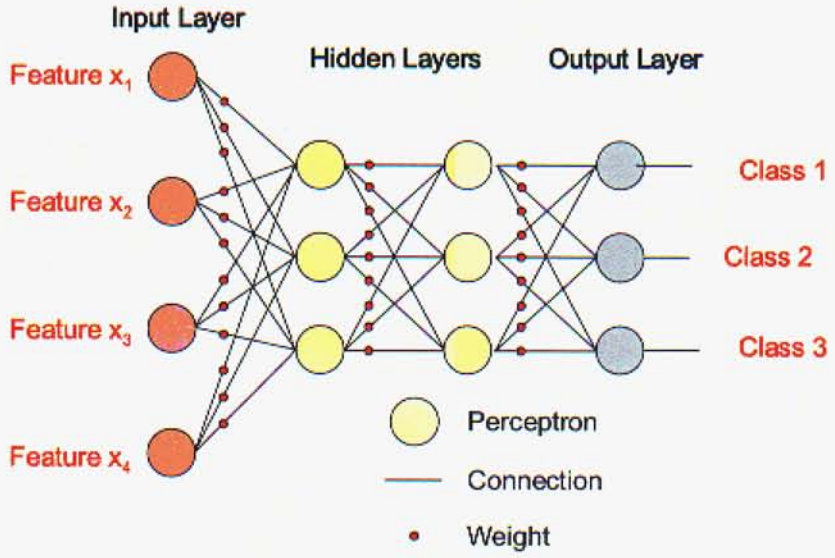

Figure 1: Schematic diagram of a multilayer perceptron neural network. This example has four inputs and three outputs with two hidden layers.

For functional approximation only a single output perceptron is used and its functional form is simply linear. Thus, the network performs a transform that results in a functional approximation of $f(x_1, x_2, ...x_n) = y$. It should be noted that MLPs with two hidden layers are capable of universal functional approximation making them a very powerful predictive algorithm.

Because the number of data points are limited, the neural network had to be relatively small (*i.e.*, with few weights). This necessitated a search for a few critical measurements in the suite of geophysical logs that best inform the prediction of hydraulic conductivity or transmissivity. Determining the best geophysical logs to use to estimate hydraulic conductivity is not a simple problem. Examining the linear correlation of each geophysical log measurement against the hydraulic conductivity is not effective as this only examines one log at a time and it is the combination of different geophysical log

responses that have the greatest chance of predicting the hydraulic conductivity. For this work, a simple genetic algorithm was used to evolve a population of such neural networks (*i.e.*, their parameters, architecture, and a subset of geophysical logs to use as inputs) in a search for an optimal one.

Phase I Simulations: Predicting Hydraulic Conductivity Classes

Algorithm Methodology

Genetic algorithms (See, for example, *Goldberg* [1989]) are a subset of evolutionary algorithms particularly suited to optimization problems involving combinatoric difficulty. They evolve a population of solutions that compete based on their ability to solve the problem at hand. A single solution can be considered as an individual within a population of solutions. Each individual's problem-solving ability is measured and used to assign it a "fitness" value. Fitness values determine the number of offspring each individual solution produces for the next generation, in a simple approximation of Darwinian evolution. Each individual in the population can be considered as a combination of parameters that produce a particular solution. The parameter values in each individual are encoded as a chromosome that encodes its particular solution to the problem. "Mating" between offspring, combined with mutation, is then used to explore the search space in the next generation through a recombination of chromosomes.

In this work, the genetic algorithm is used for "feature selection". Feature selection is a technique that has been developed to attempt to identify an optimal subset of features that can predict another quantity, for example, with a neural network. In this application, we are trying to determine an optimal subset of geophysical logs that can predict hydraulic conductivity. This method works well if there exists a set of measured features that inform the prediction problem. If such features do not exist, then the measured values will not be sensitive to any of the features used in the predictions. This method was particularly suited for our problem of estimating hydraulic conductivity from geophysical logs, since we were constrained to work with a subset of the geophysical measurements (see "Data and Data Constraints").

We used a simple genetic algorithm, POBBLE [*Reeves*, 2001a] to evolve a feed-forward neural network, JUMBLIE [*Reeves*, 2001b] to solve the hydraulic conductivity prediction problem. The general algorithm methodology consisted of running a simple genetic algorithm where each individual in the population specified the parameters and architecture of a feed-forward neural network. Performance measures such as mean-squared error, confusion-matrix values, and neural-network architecture were used to assign fitness values to each individual. The genetic algorithm then evolved the population of potential solutions towards an optimal one. Specific issues regarding the algorithmic methodology are addressed below.

Data and Data Constraints

The data consist of twelve distinct geophysical measurements (Depth, Elevation, Natural Gamma, Long Normal Resistivity, Short Normal Resistivity, Micro 1 Resistivity, Micro 2 Resistivity, Porosity determined from neutron logging, Self Potential, Sonic, Temperature, and Neutron) along nine boreholes (DH-5, DH-6, DH-7, DH-9, DH-12, DH-13, MIU-1, MIU-2, and MIU-3). The geophysical logs are collected on a 10 cm sampling interval, which is considerably smaller than the length of the packer intervals used to measure hydraulic conductivity. This discrepancy in scale is addressed here by

defining the geophysical log measurements within each hydraulic testing zone by the minimum, maximum, mean, and standard deviation of each log across the hydraulic test interval. The focus of this project is to use the geophysical logs collected at a fine spatial resolution to predict hydraulic conductivity or transmissivity measured at a much larger scale. Not only is the vertical dimension of the hydraulic test zone much larger than the vertical dimension of the geophysical log measurements, but the volume of the rock that is investigated by the hydraulic test may be much larger than that investigated by the geophysical logging tools.

In order to gain some appreciation of the volume of rock investigated by the geophysical logging tools, Figure 2 was constructed to provide an idea of the vertical resolution of the logs as a function of the radius of investigation of the logs. The vertical resolution is the length of the rock parallel to the borehole over which the geophysical log response is averaged. The radius of investigation is the distance into the rock orthogonal to the borehole length to which the geophysical log can measure properties. The best possible geophysical log would have a very fine vertical resolution with a deep radius of investigation. Such a log would plot in the lower right hand corner of Figure 2. As can be seen from Figure 2, very few logging tools (e.g., the deep laterlog) exist that have a radius of investigation that is greater than the vertical resolution

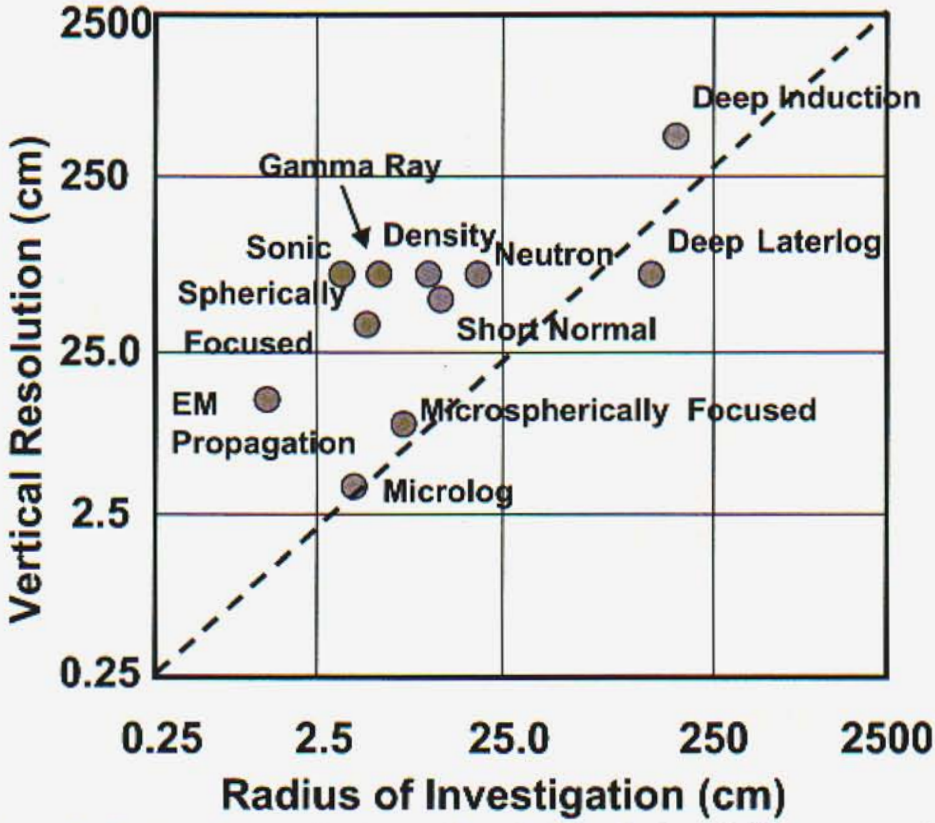

Figure 2. Radius of investigation and vertical resolution of different geophysical logs (After Doveton and Olea, 2001)

Figure 2 shows the vertical resolution and radius of investigation for a number of the geophysical logs used at the Shobasama site (the deep induction, EM propagation and deep laterlog are not used at the Shobasama site). Generally these logs have a vertical resolution of approximately 3-4 cm (microlog resistivity) to 1 meters and a radius of investigation between 1 cm and 20 cm (neutron). It is noted that the results shown in Figure 2 were developed for carbonate and sandstone rocks (see Doveton and Olea, 2001) and may not apply exactly for the fractured granites at the Shobasama site.

In addition to the geophysical logs, a borehole televiewer was used to count the number of "clear", "open", and "hairline" cracks on one-meter intervals down the borehole. For each hydraulic test interval, the sum of the different types of cracks and the total number of cracks are recorded and used as input to the hydraulic conductivity prediction problem.

For each hydraulic test interval, a packer test had been performed to measure hydraulic conductivity. The typical hydraulic test interval length is 6.5 meters. The hydraulic conductivity data set consisted of 137 such measurements, ("events" in neural network terms), with 51 measured geophysical log summary statistics ("features" in neural network terms) per event. The 51 possible features are the minimum, maximum, mean and standard deviation of the individual log response within each hydraulic test zone for each of the twelve different logs plus the three different crack counts.

Borehole MIU-1, which had 20 events, was set aside as a test set to be used in the validation phase, and the remaining 117 events were used as a training set to determine the optimal input well log variables and to train the neural network. MIU-1 was chosen because it has a broad range in hydraulic conductivities that could be used to stress the neural network. Thus, the first goal of this study is to develop a neural network that can predict membership of hydraulic conductivity in one of a small number of classes in a new well.

The cumulative distribution of measured hydraulic conductivity values are shown in Figure 3. Based on this plot, three different classes of hydraulic conductivity were defined using the following boundaries (in m/s):

low	$K \leq 2.0E-10$
medium	$2.0E-10 < K < 1.0E-07$
high	$1.0E-07 \leq K$

The number of events in each class, broken down by training and testing (MIU-1) sets is given in Table 1.

	Training	Testing
Low K	26	6
Medium K	71	11
High K	20	3

Table 1: Number of each hydraulic conductivity class for the training and test sets.

Ideally, there should be equal numbers of events in each of the classes for the training set. For this training set, we had the option of removing some of the medium conductivity measurements, but due to the already limited number of data, we chose not to decrease the size of the training set.

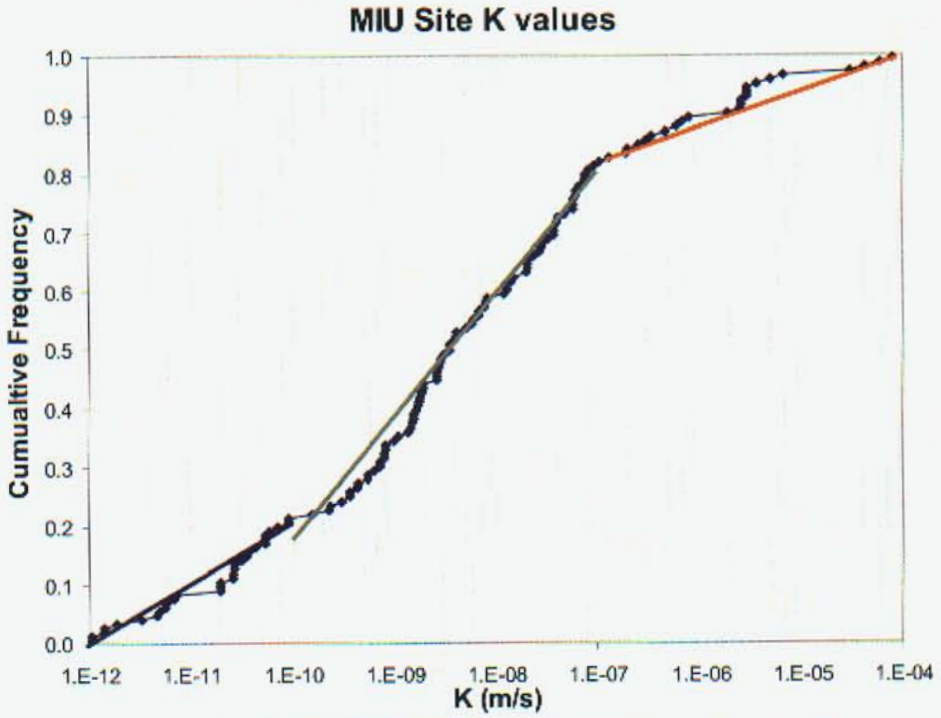

Figure 3: Measured hydraulic conductivity for the entire data set. The colored lines indicate the bounds of the three different classes.

The limited number of data points also creates a constraint for the neural network architecture. A general rule of thumb is that 5 to10 training events are required for each weight in a neural network. For a fully connected neural network having two hidden layers, the total number of weights is given by:

$$N_{tot}^{Weights} = \left(N_{in}^{PE} \cdot N_{layer1}^{PE}\right) + \left(N_{layer1}^{PE} \cdot N_{layer2}^{PE}\right) + \left(N_{layer2}^{PE} \cdot N_{output}^{PE}\right)$$

where,

$N_{tot}^{Weights}$	Total Weights in the Neural Net
N_{in}^{PE}	Number of Input Perceptrons
N_{layer1}^{PE}	Number of Perceptrons in Hidden Layer 1
N_{layer2}^{PE}	Number of Perceptrons in Hidden Layer 2
N_{out}^{PE}	Number of Perceptrons the Output Layer

The rule of thumb indicates we can only use about 24 weights if we want to develop a network that is not over trained to the point where it can give very good results on the training data but has almost no predictive power. More weights than this can be expected to produce a network that performs well on the training set but fails on the test set. Ideally we would like to have twice this amount of training events since this is the lower limit of only 5 training events per weight.

This observation implies that only a subset of the measurements can be used as input to the neural network, since each event has 51 features. Because of this constraint, the core of this project focused on a search for an optimal subset of measured features that could be used to predict hydraulic conductivity. A simple genetic algorithm was used to solve this combinatoric problem of determining the optimal subset of measured features.

Algorithm Details

A multilayer perceptron, a subset of feed-forward neural networks was used in this study. This is in contrast to the radial basis formulation used by *Wong and Shibli* [1998], who tried to develop a functional approximation for permeability from well-log measurements. We chose a multi-layer perceptron because they can be shown to be universal function approximators, and because they are ideal for the solution of classification problems.

Neural network parameters that were held fixed are given in Table 2. The individuals in the genetic algorithm population encoded additional parameters used to define the neural network architecture, specific learning and momentum parameters, the annealing schedule, and the subset of geophysical measurements to use as inputs to the neural network (see below).

Parameter	***Formulation***
Perceptron Function	Logistic Function
Training Methodology	Standard Back Propagation w/ Momentum
Back Propagation Update Rate	After Each Epoch
Learning Rate	Annealed
Weight Initialization	Random Initial Weights

Table 2. Neural network parameters and formulation.

The backpropagation algorithm is used to adjust weights during the training phase. For standard backpropagation with momentum learning we adjust weights after each training event has been processed (aka a single epoch) according to (See Figure 4):

$$w_{ij}(n+1) = w_{ij}(n) + \eta(n)\delta_i(n)y_j(n) + \alpha\left[w_{ij}(n) - w_{ij}(n-1)\right]$$

$$\Delta w_{ij} = \eta(n)\delta_i(n)y_j(n) + \alpha\left[w_{ij}(n) - w_{ij}(n-1)\right]$$

A single epoch is a complete iteration of the training process where the error in estimating each hydraulic conductivity measurement is backpropogated through the neural network and used to update the weights. Therefore a single epoch results in assessment of all of the training events and a single updating of the weights in the neural network.

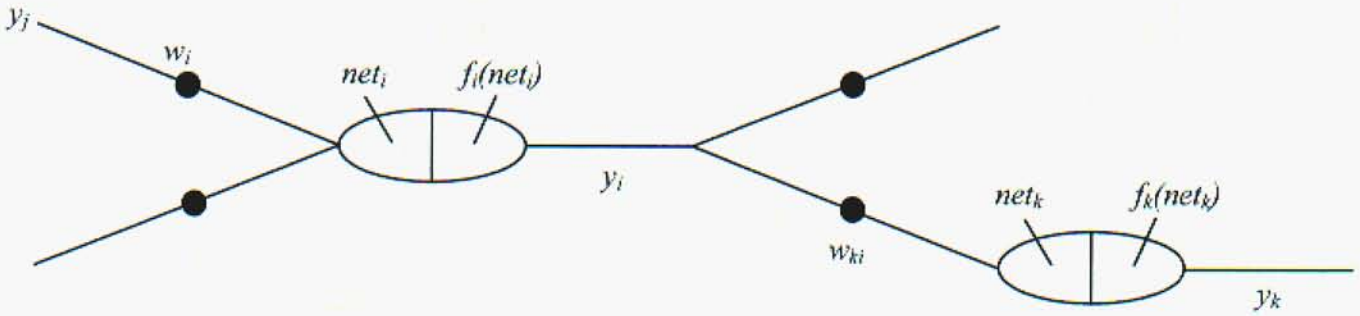

Figure 4. Definition of quantities pertaining to weight adjustments during training.

where:

$\eta(n)$ Learning Rate at epoch n. This value is usually adjusted downward as the simulation progresses according to the annealing rule:

$$\eta(n) = \frac{\eta_0}{1 + \frac{n}{n_0}} \qquad \begin{cases} \eta \approx \text{constant} & \text{for } n << n_0 \\ \eta \to 0 & \text{for } n >> n_0 \end{cases}$$

η_0 Initial Step Size

n_0 Annealing Constant

α Momentum Constant , Typically $0.5 < \alpha < 0.9$

The momentum constant aides in overcoming local minima. The primary values affecting the weight adjustment are the "local error", $\delta_i(n)$, and the outputs of the local perceptrons, $y_j(n)$. These values are defined differently depending on which layer of the neural network they correspond to:

Input Layer

$$y_i(n) = feature_i(n)$$

$$\delta_i(n) = none$$

Hidden Layer

$$y_i(n) = f_i(net_i(n))$$

$$\delta_i(n) = f_i'(net_i(n)) \sum_{downstream,k} \delta_k(n) w_{ki}(n)$$

Output Layer

$$y_i(n) = f_i(net_i(n))$$

$$\delta_i(n) = -f_i'(net_i(n))[d_i(n) - y_i(n)]$$

and where

$$net_i(n) = \sum_{upstream,j} w_{ij}(n) y_j(n)$$

$f_i(net_i(n))$ Perceptron Function. A logistic function was used: $f_i(net_i(n)) = \dfrac{1}{1+\exp(net_i(n))}$

A simple genetic algorithm using standard mutation, selection, and crossover operators was used to evolve the free parameters of the neural network. The genetic algorithm formulation is shown in Table 3 and Figure 5. The values in Table 3 are taken from previous experience and values published in other studies.

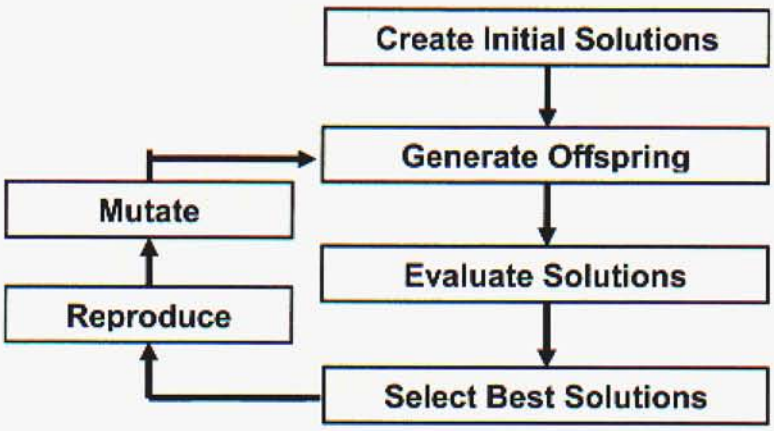

Figure 5. Schematic flow chart of the genetic algorithm process.

It should be stressed that these simulations are very computationally expensive due to the cost of computing the objective function of a single individual: constructing, initializing and then training a unique neural network. One significant advantage of genetic algorithms is that they can continue to make progress using an approximation to a true objective function. This capability proved extremely valuable for this research. For example, it is possible to train the neural network for a "small" number of epochs and get a sense of the value of the neural network setup through the resulting mean squared error. These shortcuts are valid within a genetic algorithm framework, since we do not seek the optimal setup, rather, we wish to maintain a population with potential for improvement in later generations.

Parameter	*Value*
Population Size	400
Chromosome Length	73 (bits)
Probability of Mutation	0.001
Selection Scheme	Tournament Selection w/ Replacement
Tournament Size	4
Crossover Scheme	Uniform
Probability of Crossover	0.8

Table 3. Genetic algorithm parameters and formulation.

A major factor in training the neural network is the number of epochs of back propagation applied. Simulations showed that the neural network could continue to improve even after a few thousand iterations. However, a full genetic algorithm run using 2000 iterations per neural network requires in excess of 6 hours per neural network architecture formulation on a Sun Fire UltraSparc 2 workstation. This led us to apply a simple scheduling to the first round of simulations (Table 4).

Generation	*Number of Epochs of Back Propagation*
0-4	200
4-8	400
9-12	800

Table 4. Scheduling of the number epochs of back propagation used for the first round of baseline simulations.

Chromosomal Encoding

Chromosomes of the individuals in the genetic algorithm encoded the free parameters of the neural network. These consisted of:

- Initial Learning Rate (0.0 – 4.0, 8 Bits)
- Learning Rate Decay Parameter (1-64 x Maximum Epochs, 6 Bits)
- Momentum Parameter (0.4 - 0.9, 8 Bits)
- Subset of Geophysical Measurements to Input to the Neural Network (51 Bits)

The chromosomes were encoded as binary [0;1] strings for a total of 73 bits per chromosome. A Gray code, an alternative binary encoding that only requires a single bit change between successive integers, was used for all the integer and real parameters to avoid Hamming cliffs where changes between one integer and the next require all bits to change (*e.g.*, for a four bit string, changing from 7 to 8 requires changing from 0111 to 1000). The subset of measurements to use as inputs to the neural net was chosen using the bits as true/false values.

Objective Function Formulation

Three objective functions were ultimately used to quantify the difference between the predicted and measured hydraulic conductivity classes. The first objective function was based on the mean-squared error computed for the entire training set after a given fixed period of training. The mean-squared error is defined as:

$$MSE = \frac{\sum_{events}\left(\sum_{outputs}\left(c_d - c_p\right)\right)^2}{N_{events}}$$

where,

c_d	Desired Class (1 if correct, 0 otherwise)
c_p	Predicted Class ($0 \le c_p \le 1$) (arbitrary units)
N_{events}	Total Number of Training Events

Note that we have three distinct outputs representing the three distinct classes. A penalty was applied to the MSE to define objective function 1

$$Objective\ Function\ 1 = MSE + MSE \cdot N_{excess}^{weights} / N_{max}^{weights}$$

where,

$N_{excess}^{Weights}$	Number of Weights in Excess of the Maximum Allowed
$N_{max}^{weights}$	Maximum Number of Allowable Weights $\left(N_{max}^{weights} = 24\right)$

The second term is simply the ratio of the number of excess weights to the maximum number of weights. The maximum allowable number of weights was set at 24 based on the data constraints discussed above. If there are no excess weights, the penalty term is zero and the objective function is simple equal to the MSE.

A second objective function was based on the "confusion matrix". For classification problems, the "confusion matrix" shows predicted versus known classes, *e.g.*, consider the hypothetical confusion matrix below:

		Known Class		
		low	medium	high
Predicted Class	low	**16**	1	4
	medium	2	**89**	7
	high	19	3	**34**

In this example we see that 16 low conductivity, 89 medium conductivity events, and 34 high conductivity events were correctly predicted. The values away from the main diagonal indicate erroneous predictions. We see 2 low conductivity events were mistakenly predicted to be medium conductivity while 7 high conductivity events were predicted to be only medium conductivity. We used the confusion matrix results to define Objective Function 2 as follows:

$$Objective\ Function\ 2 = N_{confused} + N_{confused} \cdot N_{excess}^{weights} / N_{max}^{weights}$$

where,

$N_{confused}$ Number of Confused Events:

$$N_{confused} = \sum_{i=1}^{3}\sum_{j=1}^{3} abs(i-j)N_p \qquad i \neq j$$

The number confused is simply the sum of the off-diagonals with the outer values counted twice (*i.e.*, low conductivity predicted to be high and vice versa). Again, excess weights are penalized in the same fashion as with MSE.

The third objective function used the MSE and two penalty terms that combine the number of excess weights and the confusion matrix:

$$Objective\ Function\ 3 = MSE + MSE \cdot N_{excess}^{weights} / N_{max}^{weights} + MSE \cdot N_{confused} \cdot P$$

where,

P Parameter for Penalizing Confused Values $(P = 0.01)$

The first penalty term is the same as for Objective Function 1, while the second accounts for confused event classes. In effect, the second term increases the MSE by 1% for every confused event. The value of P was chosen based on examination of the results of several trial simulations.

Results and Discussion

A broad range of simulations were performed in a search for a neural network that could be trained with the limited data set available, while still being capable of predicting an unseen training set. The simulations were run on a Sun Blade Unix workstation operating at 950 MHz.

Baseline Simulations

A baseline set of simulations was run to explore the impact of neural network architecture on performance. These simulations varied the number of hidden layers (one or two) and the number of perceptrons in the hidden layers. Figures 6a and 6b show convergence behavior for Objective Function 1. Likewise, Figures 7a,b and Figures 8a,b show analogous results for Objective Functions 2 and 3, respectively. These simulations took approximately 3 days of CPU time. The results are instructive:

1) Independent of the objective function used, we see similar convergence behavior. All of the simulations demonstrate that the algorithm is making progress throughout the simulation and that none of the configurations appears to stall within 12 generations.

2) For a network with a single hidden layer, the optimal number of perceptrons (PE) in the hidden layer appears to be three.

3) For a network with two hidden layers, the optimal numbers of perceptrons in the upstream and downstream hidden layers appear to be two and three (2-3), respectively. The only exception is for Objective Function 3 in the two hidden layer case (Figure 8b), that shows improved performance for 2-4 and 3-3 perceptrons in the upstream-downstream hidden layers.

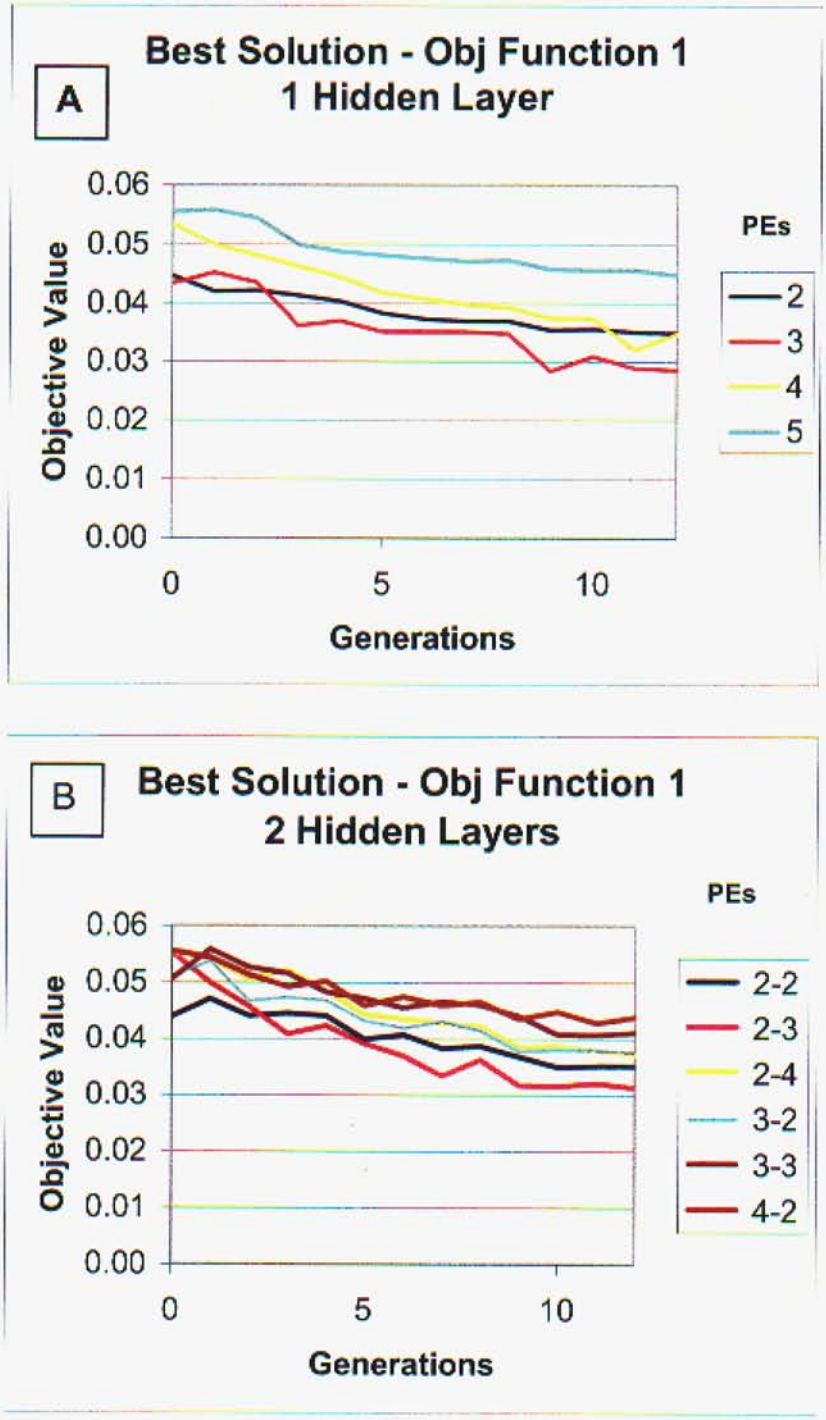

Figure 6. Objective Function 1 – Baseline Simulation. Best Solution vs. Genetic Algorithm Generation. MLP with 1 Hidden Layer (A) and 2 hidden layers (B). The different curves correspond to different numbers of perceptrons in the hidden layer.

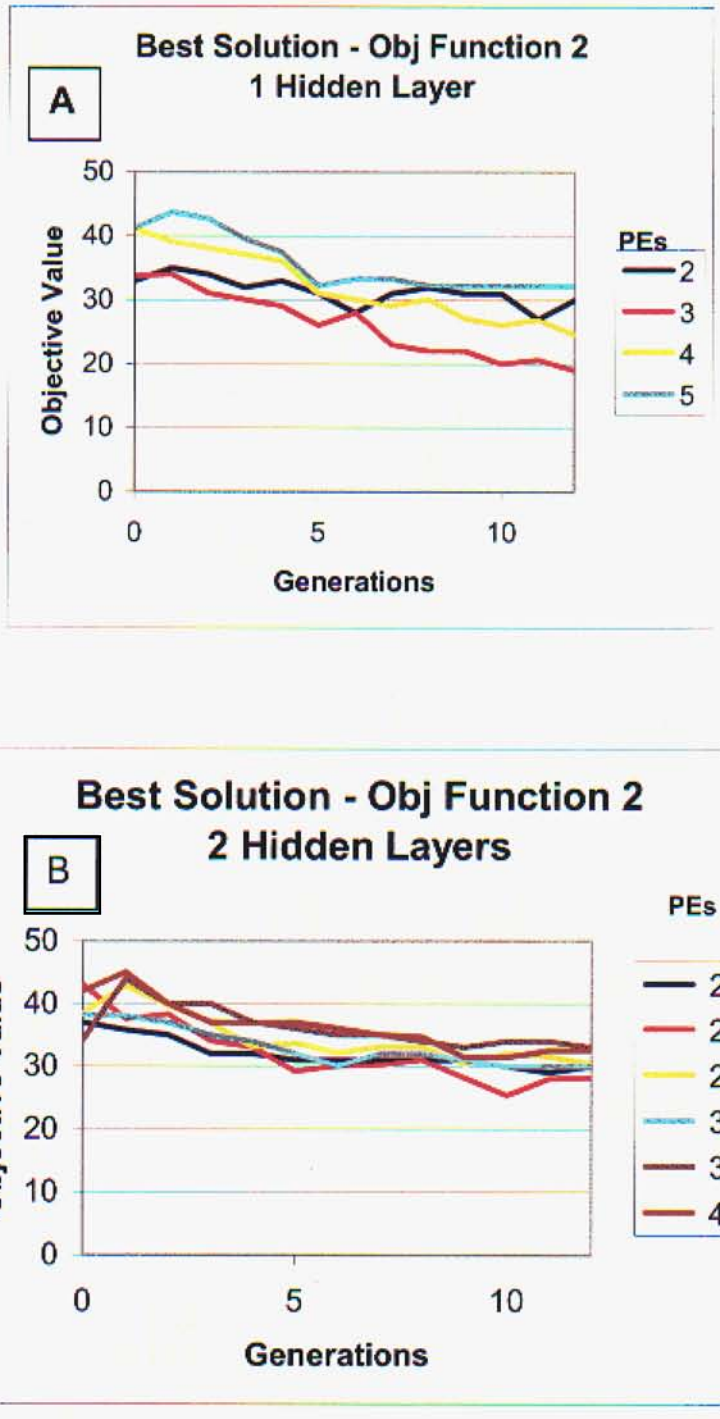

Figure 7: Objective Function 2 – Baseline Simulation. Best Solution vs. Genetic Algorithm Generation. MLP with 1 hidden layer (A) and 2 hidden layers (B).

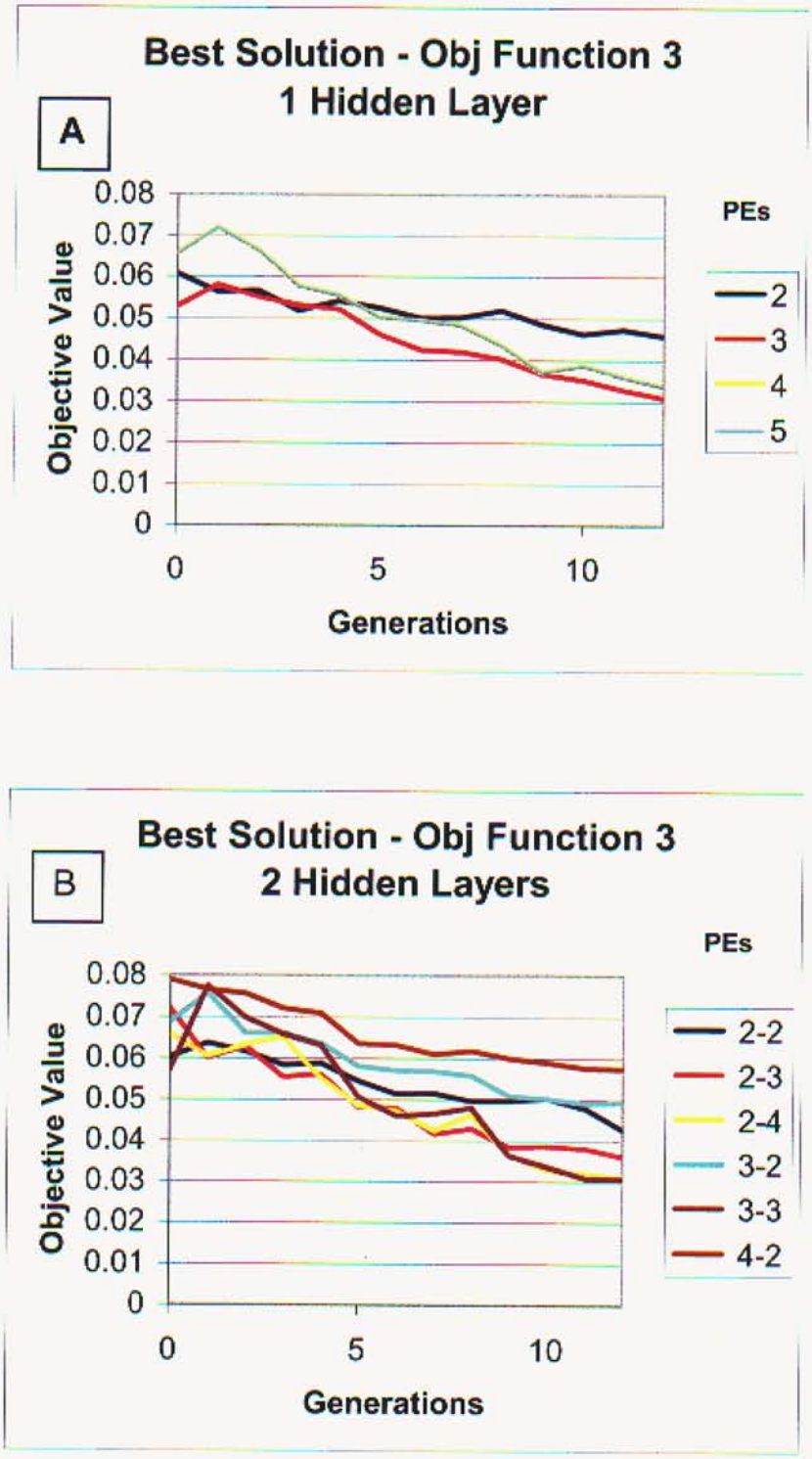

Figure 8. Objective Function 3 – Baseline Simulation. Best Solution vs. Genetic Algorithm Generation. MLP with 1 hidden layer (A) and 2 hidden layers (B).

A closer examination of the results underpinning Figure 8b shows that these low objective function values reflect neural networks that significantly exceed the maximum number of weights. After about eight generations we find that the penalty imposed by exceeding the maximum number of weights is smaller relative to the penalty imposed by the number of confused classes. After that point, the population of solutions evolved by the genetic algorithm favors large networks since they have more degrees of freedom with which to fit the training data. We see lower and lower MSE values, but at the expense of too many weights in the system.

Figures 9a,b, 10a,b, and 11a,b show the details of the best solution (*i.e.*, the fittest individual) from each of the baseline simulations, while Table 5 gives the corresponding numerical values for a subset of the results. Note that in these figures, not all solutions have excess weights.

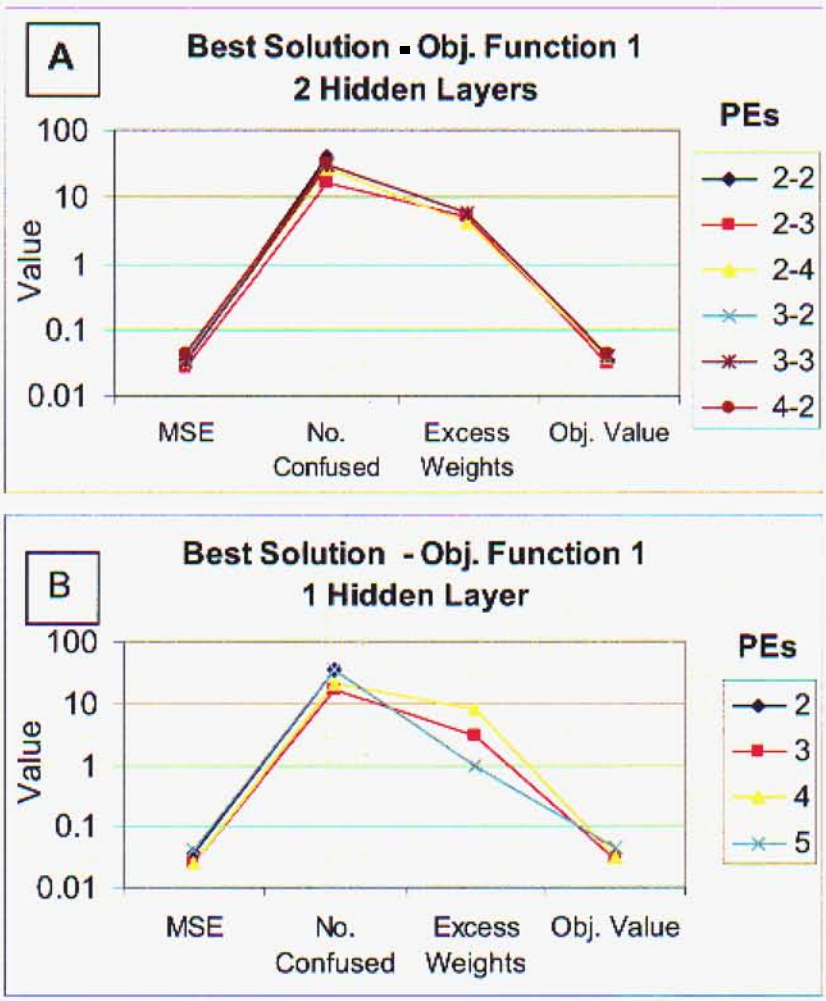

Figure 9: Objective Function 1 – Baseline Simulation. Best Solution Details. MLP with 1 hidden layer (A) and with 2 hidden layers (B). Zero values are omitted.

From Figure 10, which summarizes the details of the best solutions for Objective Function 1, we see that for Objective Function 1only the configuration with 2 perceptrons does not need to rely on excess weights to achieve its best solution. With two hidden layers, only the 3-2 and 4-2 configurations achieve their results without excess weights. We also note that two layers do not lead to improved results, which is counterintuitive. These results most likely reflect the fact that the same number of epochs was used, regardless of the number of hidden layers, and that the weights connecting the input layer to the upstream hidden layer respond more slowly to back propagation than the weights connecting the two hidden layers. Thus, it may be that a network with two hidden layers is a superior configuration, but this advantage may be masked by the slower training speed of such configurations.

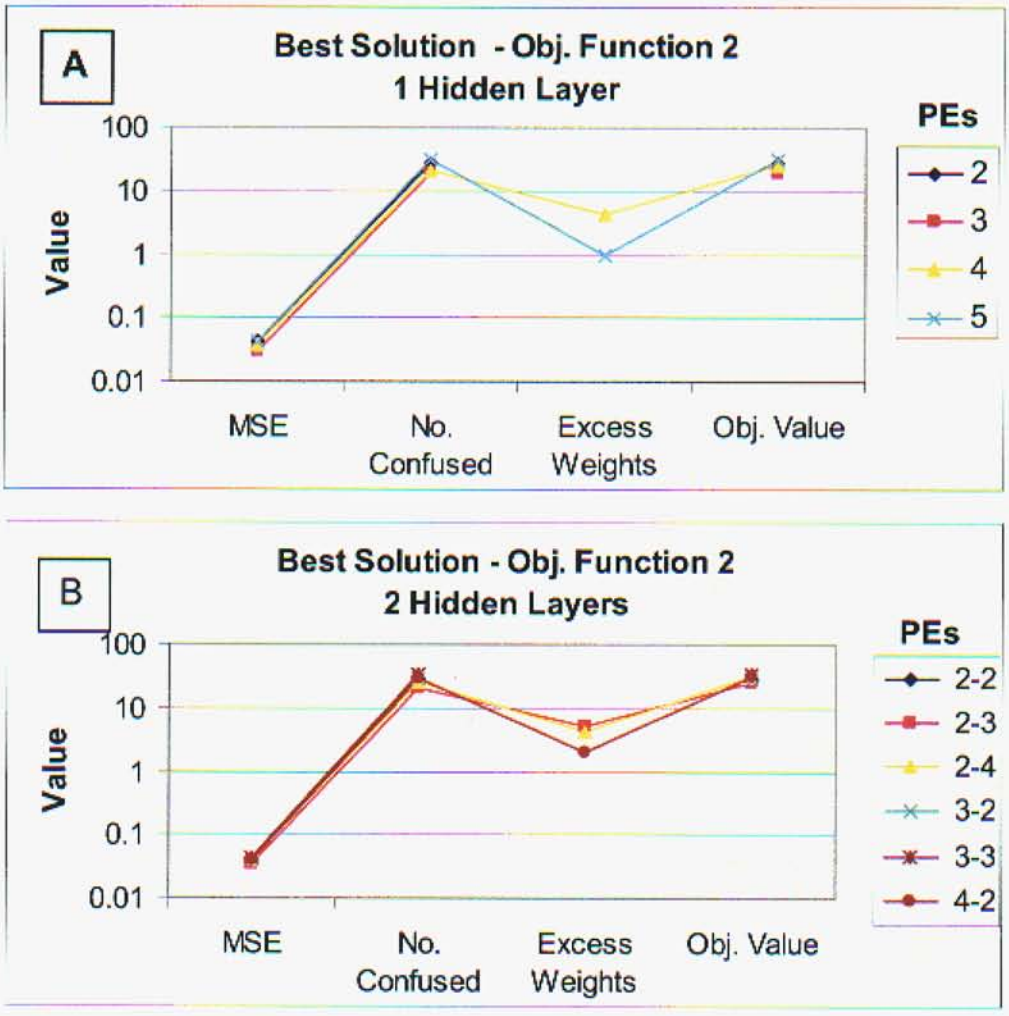

Figure 10. Objective Function 2 – Baseline Simulation. Best Solution Details. MLP with 1 hidden layer (A) and 2 hidden layers (B). Zero values are omitted.

The behavior of Objective Function 2 is similar to that of Objective Function 1, with the single hidden layer configurations using fewer excess weights than two hidden layer architectures. Again, a single hidden layer configuration performs best with 3 PEs and a two hidden layer system does best with a 2-3 PE configuration. The same results are seen with Objective Function 3. Overall, these baseline simulation configurations produced neural networks that incorrectly predicts between 12-21 of the training event. This is a relatively large percentage of errors (10-18%)

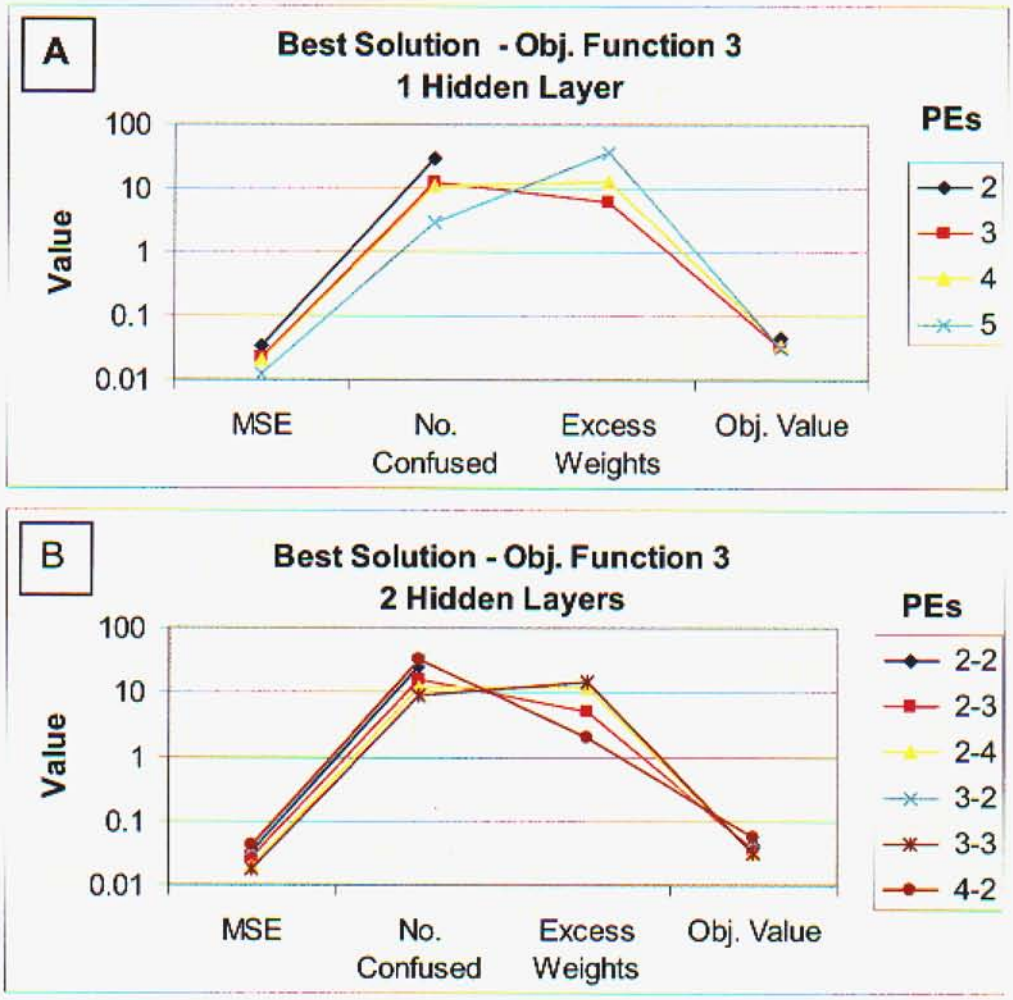

Figure 11. Objective Function 3 – Baseline Simulation. Best Solution Details. MLP with 1 Hidden Layer (A) and 2 hidden layers (B). Zero values are omitted.

	Objective Function 1		*Objective Function 2*		*Objective Function 3*	
	1 Layer	2 Layer	1 Layer	2 Layer	1 Layer	2 Layer
	3	2-3	3	2-3	3	2-3
MSE	0.02529	0.02583	0.03001	0.03433	0.02246	0.02628
No. Confused	16	17	19	21	12	16
Excess Weights	3	5	0	5	6	5
Obj. Value	0.02845	0.03122	19	25.375	0.03077	0.03596

Table 5. Summary of Best Solution Details. Only results for 1 hidden layer (3 PEs) and 2 hidden layer (2-3 PEs) configurations are presented.

Focused Simulations

The baseline results led us to retry the best configurations (1 hidden layer with 3 PEs and 2 hidden layers with 2-3 PEs) with more generations (18) and more epochs of back propagation training for the neural networks. The scheduling for these longer simulations is given in Table 6.

Figures 12, 13, and 14 show the results for the focused simulations. All objective functions show that the genetic algorithm is making very little progress by 18 generations. However, we do see lower objective function values than the shorter simulations. Like the shorter, baseline simulations, we see little improvement when two hidden layers are used, despite the extended neural net training. Summaries of these results are given in Table 7. Here we see that all of the simulations ultimately shifted toward larger networks and used more than 24 weights to achieve their minimum objective function values.

Generation	*Number of Epochs of Back Propagation*
0-4	200
4-8	400
9-18	2000

Table 6. Scheduling of the number epochs of back propagation used for the focused simulations.

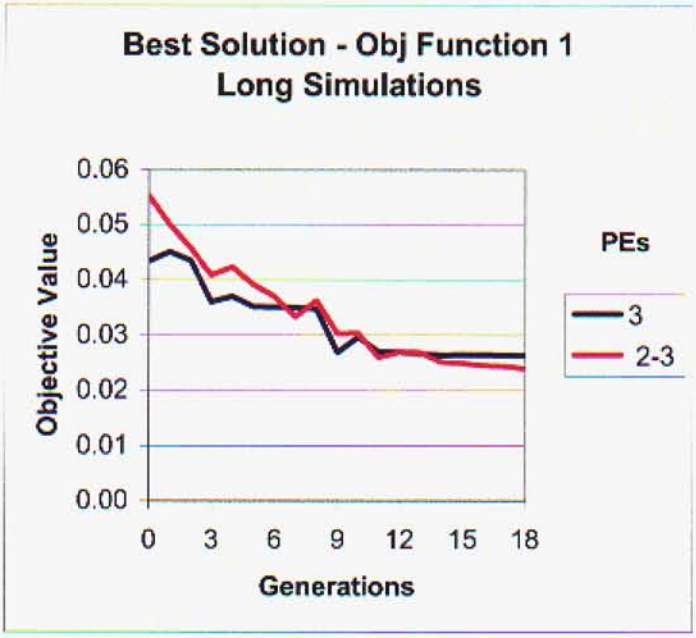

Figure 12. Objective Function 1 – Focused Simulation. Best Solution of the Generation for Two Configurations.

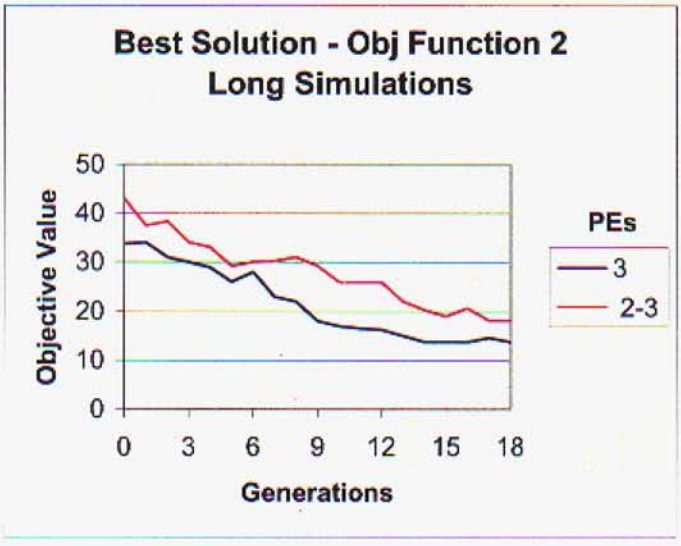

Figure 13. Objective Function 2 – Focused Simulation. Best Solution of the Generation for Two Configurations.

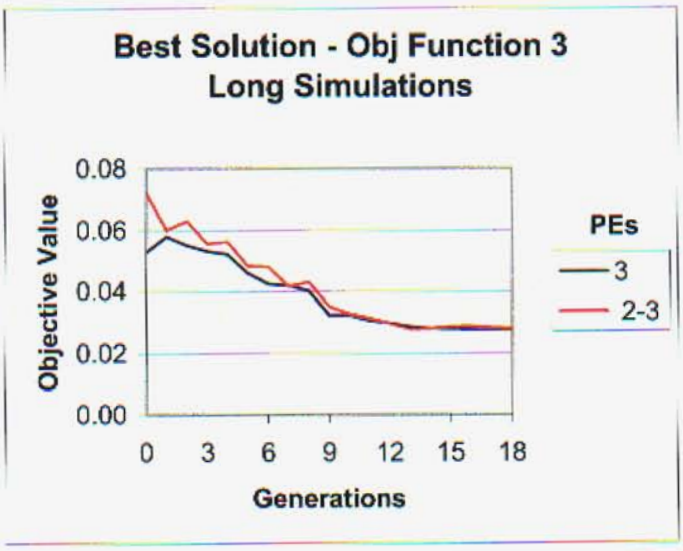

Figure 14. Objective Function 3 – Focused Simulation. Best Solution of the Generation for Two Configurations.

	Objective Function 1		*Objective Function 2*		*Objective Function 3*	
	1 Layer	2 Layer	1 Layer	2 Layer	1 Layer	2 Layer
	3	2-3	3	2-3	3	2-3
MSE	0.02343	0.01737	0.02089	0.02558	0.02011	0.01762
No. Confused	13	11	10	14	12	11
Excess Weights	3	9	9	7	30	35
Obj. Value	0.02636	0.02388	13.750	18.083	0.02755	0.02763

Table 7. Summary of Best Solution Details for the Focused Simulations. Results are presented for 1 hidden layer (3 PEs) and 2 hidden layer (2-3 PEs) configurations.

Table 8 shows the features selected as inputs to the neural networks of the best solutions from the longer focused simulations. There is some amount of variation between the features selected, but there is also a great deal of commonality. All or nearly all of the neural networks selected use the following measurements:

- Short Normal – Standard Deviation
- Micro 1 - Maximum
- Micro 1 – Standard Deviation
- Porosity – Maximum
- Self Potential – Minimum
- Sonic – Standard Deviation

We can infer that these measurements provide the best information for the hydraulic conductivity classification problem. It is notable that Depth, Gamma, Long Normal, Temperature, and Crack Counts were virtually ignored by all solutions.

As mentioned above, the long, focused simulations all resulted in neural networks with excess weights (*i.e.*, greater than 24). These networks run the risk that they will be overtrained and not be capable of predicting a new data set (*i.e.*, a data set that was not included in the training).

	Depth			Elevation			Gamma			Long	Short Normal		Micro 1			Micro 2		Self Potential		Sonic		Porosity		Temperature		Cracks		Neutron Strength	
Obj	1	1	0000	0000	0000	0000	0001	0101	0000	0100	1000	0001	0000	0000	000													
Obj	1	2	0000	0000	0001	0000	0001	0101	0010	0100	1000	0001	0000	0100	000													
Obj	2	1	0000	1000	0000	0000	0001	0101	0100	0100	0010	0001	0000	0000	000													
Obj	2	2	1000	0000	0000	0000	0100	0010	0100	0100	1000	1001	0000	0000	000													
Obj	3	1	0000	0000	0000	0000	0001	0101	0000	0100	1000	0001	0010	0000	000													
Obj	3	2	0000	0001	0000	0000	0001	0100	0100	1100	1001	0001	0000	1000	000													

Table 8: Features selected as inputs to the neural network for the best solutions from the long simulations. Columns for depth, elevation, and other geophysical measurements are minimum, maximum, mean, and standard deviation for the zone, left to right. Crack columns are "clear", "open", and "hair", respectively. The column immediately following the objective function refers to whether the results are for the single hidden layer (1) or two hidden layer (2) configurations.

Predictive Capabilities

We have run extensive testing on the optimal solutions arrived at from the focused tests. Unfortunately, none of these solutions is capable of accurately predicting even 50% of the hydraulic conductivity classes from the borehole that was withheld for testing purposes. We tried training to various numbers of epochs in case the networks had been over trained in that sense. We tried eliminating weights from the networks via weight decay methods (e.g., "optimal brain damage"). We tried training with random noise added to the measured features in an attempt to find a new globally optimal minimum. None of these efforts was successful.

A typical example of this behavior is given below. We extracted a solution from the Objective Function 2, single hidden layer simulation that used only 24 weights. This selection minimizes the potential that the network is over parameterized. Table 9 shows the parameters that were evolved for this network together with results for training with 4000 epochs of back propagation. Table 10 gives the features selected as inputs for this neural network. They represent a subset of the dominant features selected by the optimal solutions shown in Table 8, so this solution is consistent with the optimal solutions. The dominant features for this simulation are:

- Short Normal – Standard Deviation
- Micro 1 - Maximum
- Micro 1 – Standard Deviation

- Porosity – Maximum
- Sonic – Standard Deviation

The resultant confusion matrix for training is shown in Table 11. A total of 16 events are mis-classified. This is still a relatively high number (13.7%). All but one of the predictions is within one class of its known class. Figure 15 shows the training results for this simulation. The errors are concentrated in the first 2/3 of the training set.

Parameter	*Value*
Random Seed	0.004989
Initial Learning Rate	2.635290
Half Life	60000 epochs
Momentum Parameter	0.884375
Maximum Epochs	4000
Final MSE	0.02742448
Input PEs	5
Hidden Layers	1
Hidden Layer PEs	3
Output PEs	3
Total Number of Weights	24

Table 9. Objective Function 2 – Confusion Matrix for Solution "X" Individual

```
          Depth                    Short Normal          Self Potential      Cracks
               Elevation                Micro 1               Sonic
                    Gamma                    Micro 2                Temperature
                         Long Normal              Porosity                Neutron Strength
Obj 1 1 0000 0000 0000 0000 0001 0101 0000 0100 0000 0001 0000 0000 000
```

Table 10 Features selected as inputs to the neural network for one of the solutions taken from Objective Function 2, single hidden layer (3 PE), long simulation. Columns for depth, elevation, and other geophysical measurements are minimum, maximum, mean, and standard deviation for the zone, respectively. Crack columns are "clear", "open", and "hair", respectively.

		Known Class		
		low	**medium**	**high**
Predicted Class	**low**	**20**	6	0
	medium	4	**65**	2
	high	1	3	**16**

Table 11. Objective Function 2 – Confusion Matrix for the Training Set.

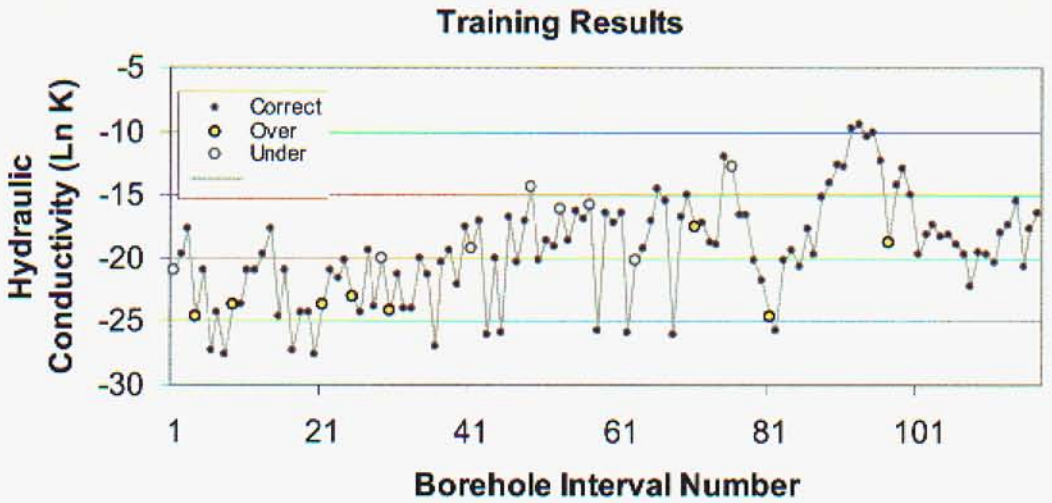

Figure 15. Typical Training Results. Predicted values are shown in colors. Correct values are in black, over predictions are in yellow and under predictions are blue. "Borehole Interval" refers to the hydraulic test interval number in the entire training set.

The last 1/3rd of the training set corresponds to the MIU-2 and MIU-3 boreholes. The poor training results may indicate that MIU-2 and MIU-3 should not be lumped together with the DH boreholes, though this conclusion is only qualitative.

Table 12 and Figure 16 give the predictive results for this neural network. Recall that borehole MIU-1 is the test set. The performance is very poor with 11 events misclassified and only 8 classified correctly. No predictions in the high permeability class were made which is no better than random chance. To emphasize, these results are typical of all the neural networks we investigated. Training success was only marginal and no network could be developed that was more than 90% accurate in training. None of the networks were capable of adequately predicting hydraulic conductivity classes of the test borehole (MIU-1).

		Known Class		
		low	**medium**	**high**
Predicted Class	**low**	**3**	6	2
	medium	2	**5**	1
	high	0	0	**0**

Table 12. Objective Function 2 – Confusion Matrix for the Test Set. (Borehole MIU-1)

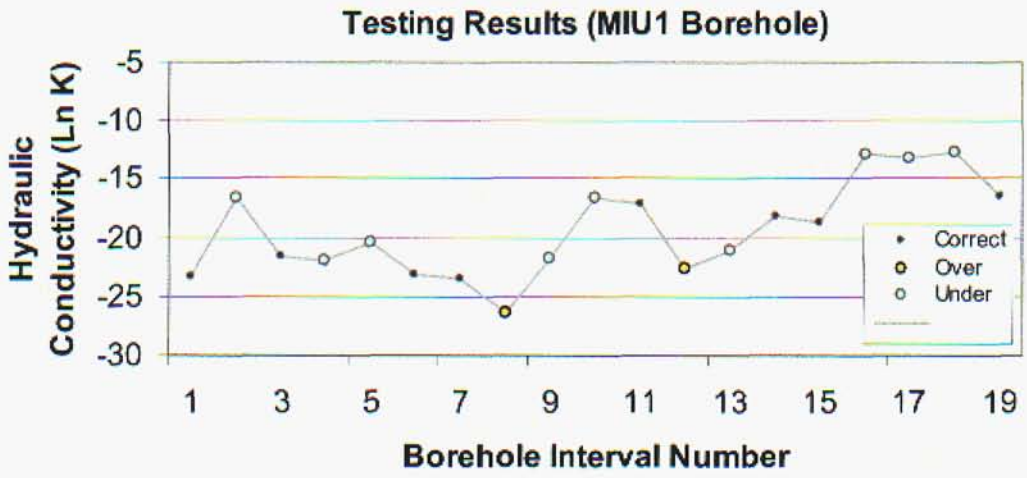

Figure 16. Typical Testing Results. Note: The test set is Borehole MIU-1. Correct values are in black, over predictions are in yellow and under predictions are blue. "Borehole Interval" refers to the interval number, from top to bottom, of the hydraulic tests done in granite for Borehole MIU-1

Phase II Simulations: Transmissivity Estimation

The initial part of this report (Phase I) described an algorithm that attempted to evolve an optimal set of input data for the neural network (downselected from the set of 51 possible measurements) while simultaneously attempting to optimize a set of parameters for the resulting neural net. Each geophysical borehole log yielded minimum, maximum, mean, and standard deviation values for each hydraulic test interval. The Phase 1 simulations were designed to predict the class: high, medium or low, of hydraulic conductivity.

The Phase 1 approach was not successfully completed since we failed to develop a neural network that could be used to predict the hydraulic conductivity measured in a borehole that had not been included in the training set. We hypothesize that this failure may stem from two sources of difficulty involving:

Data: Minimum and maximum values of the geophysical logs may be too noisy to provide informative measurements. Likewise, hydraulic conductivity does not account for variations in the length of measurement intervals and all of the testing data were taken from the same borehole.

Algorithm: Rates of convergence in neural network training are impacted by parameter settings, architecture (number of hidden layers, number of inputs, and numbers of perceptrons in each layer), and the predictive value of the chosen inputs. It is probable that selection of the neural network parameters and architecture are dominating the genetic algorithm and causing it to select neural networks based on these factors while ignoring the value of which geophysical measurements are being used as input to the neural networks in the population.

We addressed these two issues in turn, first by modifying the data set and then modifying the numerical algorithm. With the exception of the depth and elevation logs where the minimum and maximum of the interval are still used, we replaced minimum and maximum values of the other geophysical measurements with 5th and 95th percentiles. Additionally, transmissivity values were computed as a replacement for hydraulic conductivity in order to incorporate hydraulic test interval length.

The new data set was run through the original algorithm with virtually identical results: it was not possible to develop a neural network that was capable of predicting both a training set and an unknown test set (borehole MIU-1). This result confirmed that the algorithm also needed modification.

Algorithm Modification

We decoupled the simultaneous search for optimal neural network parameters and architecture from the search for informative geophysical logs. This was done by breaking the simulations into two distinct tasks: (task a) identifying informative geophysical measurements and then (task b) optimizing neural network parameters and architecture using the geophysical logs identified in the first task (task a). This section presents results for Phase II, task a.

To identify important geophysical logs, we fixed neural network parameters (i.e., initial learning parameter, half-life, momentum parameter and epochs of training) and constrained architectures to a limited range of one and two hidden layer networks with 4, 5, 6, 7 or 8 inputs. Thus, the chromosome used in the genetic algorithm only encoded which of the 51 geophysical measurements to use as inputs for a given individual (neural network) in the population. We performed a series of such simulations using 4, 5, 6, 7, and 8 inputs and with either one or two hidden layer neural networks.

Results and Discussion

Results of the Phase II simulations showing the reduction in the training MSE as a function of the number of GA generations are shown in Figure 17. Figure 17 shows that generally the larger the number of inputs the lower the training MSE becomes. The differences in MSE between the one layer and two layer neural network results are less significant than the differences caused by varying the number of inputs.

Figure 18 shows two examples of the selection of the geophysical logs as a function of the number of GA generations. The progress of geophysical log selection for a neural network with 4 inputs and 1 hidden layer is shown in the upper image and the log selection process for a neural network with 8 inputs and 2 hidden layers is shown in the lower image of Figure 18. The images in Figure 18 clearly show when a certain geophysical measurement displaces another as better and better neural networks are evolved. The results of the log selection process are summarized in Tables 13 and 14, which are for the one and two hidden layer neural networks, respectively.

For one hidden layer, the depth and sonic log values are always selected as inputs. Results of the different calculations show that these measurements are always selected early during the simulations, indicating that their information content with respect to transmissivity prediction is relatively high. It is notable that the minimum depth for an interval is always selected; whereas the 5th percentile, and standard deviation of sonic log values are chosen, depending on the number of neural network inputs. Long normal and self-potential logs are also chosen in four out of five simulations. However, the specific attribute of the long-normal and self potential geophysical logs that is selected is more variable than in the single-layer neural network results. Generally, the 95th percentile of these two logs within a hydraulic test zone is selected. Porosity, temperature, and crack counts are never selected.

For two hidden layers, we also see that depth and sonic log values are consistently chosen. Self-potential again appears in four out of five simulations, but selection of the

long normal log occurs less often than the single hidden layer. In general, with two hidden layers the range of measurements selected is broad. Whereas temperature was ignored in the single hidden layer simulations, it appears in 3/5 of the two hidden layer simulations. However, similar to the single-layer case, porosity and crack counts are not selected.

Table 15 presents the selection frequency summaries for Phase IIa simulations. From these results, we can conclude that (a) depth and sonic logs are the most informative measurements, (b) long normal and self potential are moderately informative, (c) porosity and crack counts do not inform the transmissivity prediction, and (d) elevation, gamma short normal, micro logs, temperature, and neutron strength do not appear to contain significant informative value, but cannot be completely ignored.

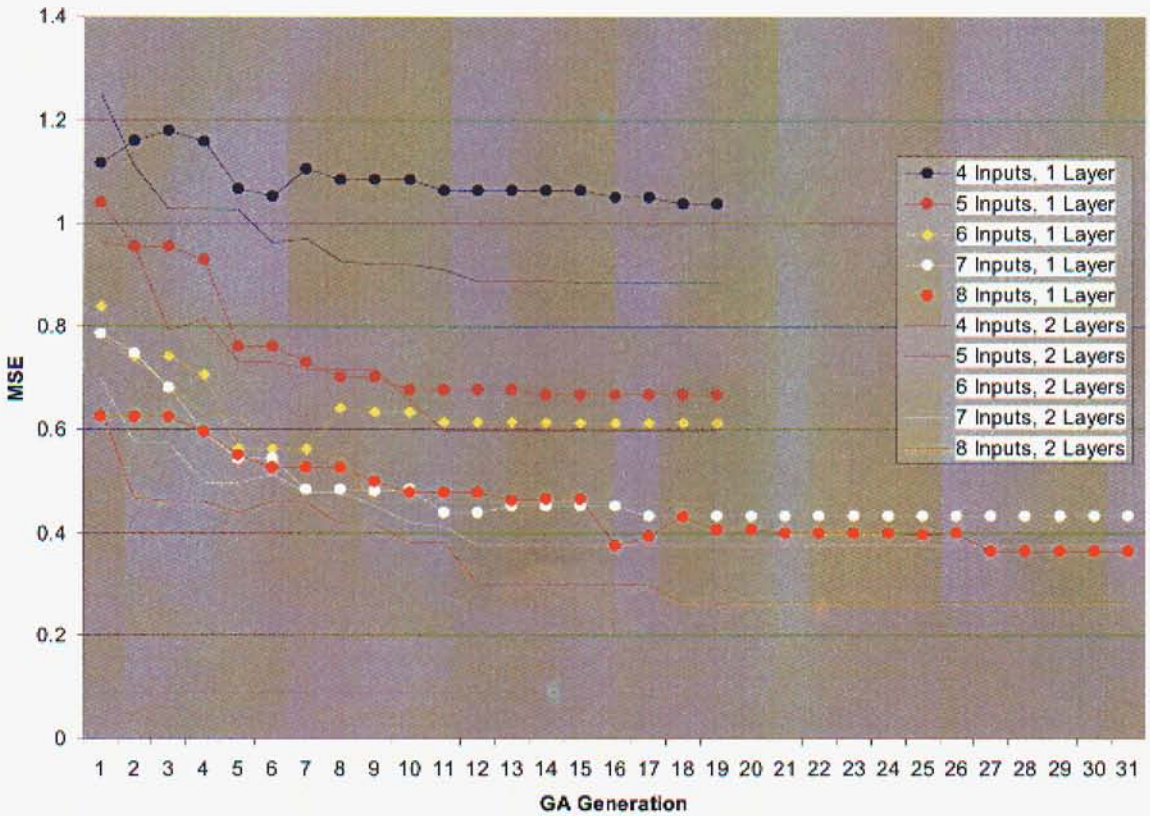

Figure 17. The training MSE for different neural network architectures as a function of the number of GA generations.

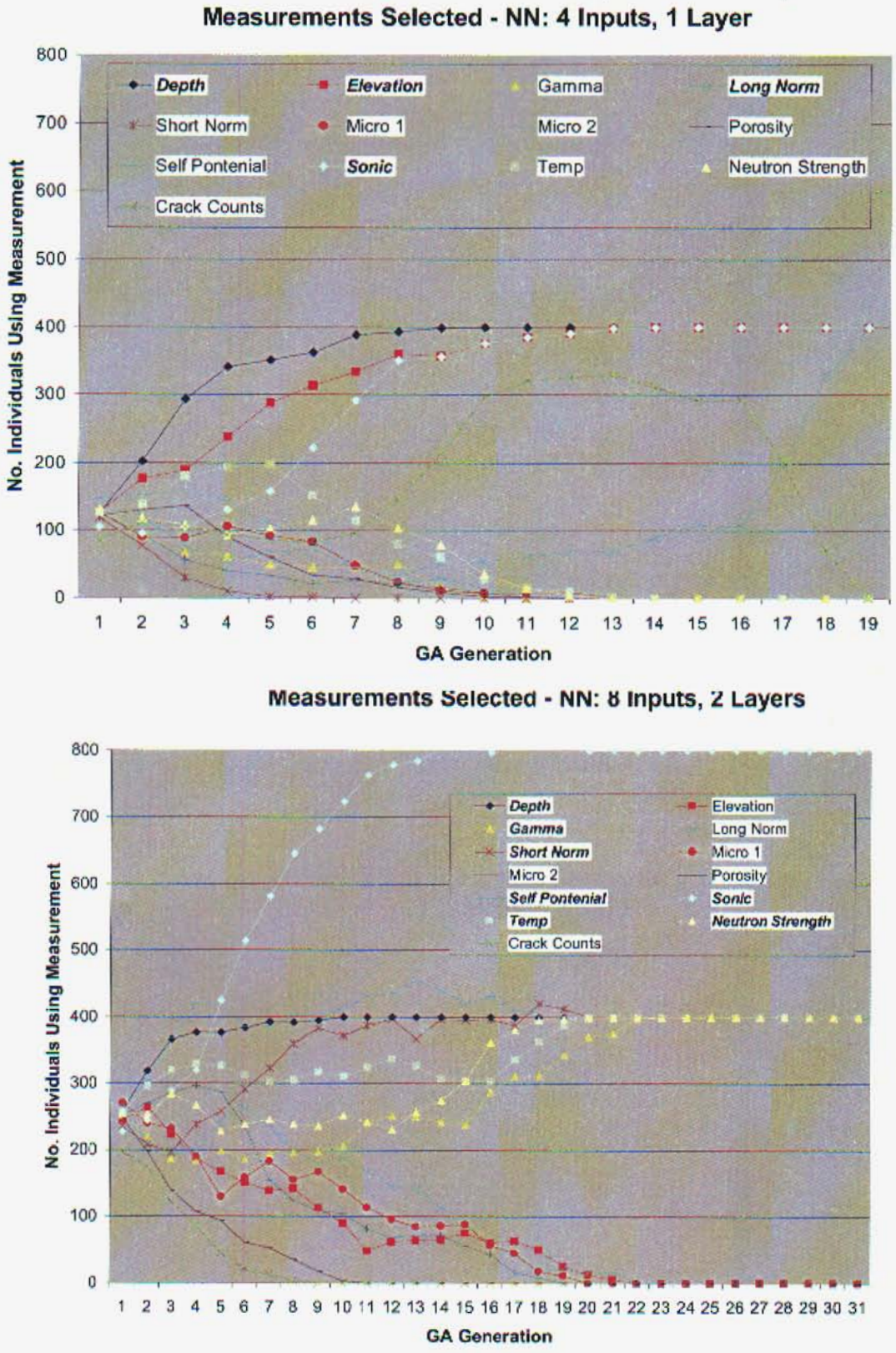

Figure 18. Example results of geophysical log selection using a GA. The population size of the GA is 400.

	1 Layer Neural Network				
	4 Inputs	**5 Inputs**	**6 Inputs**	**7 Inputs**	**8 Inputs**
Depth	min	min	min	min	min
Elevation	min		mean		
Gamma		5th			5th
Long Norm	95th	95th	std dev		95th
Short Norm				95th	std dev
Micro 1			mean		
Micro 2					mean
Porosity					
Self Potential		95th	95th	5th / 95th	5th
Sonic	5th	std dev	5th	5th / std dev	std dev
Temp					
Neutron Strength				std dev	mean
Crack Counts					

Table 13: Results of the geophysical measurement selections for the single-hidden layer neural networks.

	2 Layer Neural Network				
	4 Inputs	**5 Inputs**	**6 Inputs**	**7 Inputs**	**8 Inputs**
Depth	min	min	min	min	mean
Elevation	min				
Gamma		5th			5th
Long Norm	95th	95th		std dev	
Short Norm			95th		95th
Micro 1			5th		
Micro 2				95th	
Porosity					
Self Potential		95th	95th	5th	5th
Sonic	mean	std dev	95th	95th / std dev	5th / 95th
Temp			95th	mean	95th
Neutron Strength					std dev
Crack Counts					

Table 14. Results of the geophysical measurement selections for the two-hidden layer neural networks.

	Selection Frequency		
	1 Hidden Layer	2 Hidden Layers	Total
Depth	5/5	5/5	**10/10**
Elevation	2/5	1/5	3/10
Gamma	2/5	2/5	4/10
Long Norm	4/5	3/5	**7/10**
Short Norm	2/5	2/5	4/10
Micro 1	1/5	1/5	2/10
Micro 2	1/5	1/5	2/10
Porosity	0/5	0/5	0/10
Self Potential	4/5	4/5	**8/10**
Sonic	5/5	5/5	**10/10**
Temp	0/5	3/5	3/5
Neutron Strength	1/5	1/5	2/10
Crack Counts	0/5	0/5	0/10

Table 15. Summary of selection frequency of geophysical measurements for Phase IIa simulations.

Table 16 presents correlation coefficients for the data. These results indicate that raw correlation coefficients between individual geophysical log measurements and transmissivity are insufficient indicators of predictive value, since a set of measurements is needed for the functional approximation problem. A blind reliance on correlation being above 0.10 would have chosen both elevation and porosity logs, which were demonstrated to have no informative value in the neural network.

	Depth	Elevation	Gamma	Long Norm	Short Norm	Micro1	Micro2	Porosity	SP	Sonic	Temp	Neutron
Elevation	-0.969											
Gamma	-0.114	0.046										
Long Norm	0.229	-0.216	-0.118									
Short Norm	0.217	-0.198	-0.143	0.933								
Micro1	0.349	-0.346	-0.151	0.500	0.542							
Micro2	0.363	-0.354	-0.180	0.513	0.571	0.988						
Porosity	-0.356	0.214	-0.158	-0.305	-0.395	-0.290	-0.319					
SP	0.418	-0.431	-0.137	-0.426	-0.416	-0.269	-0.275	0.225				
Sonic	0.111	-0.076	0.139	0.383	0.383	0.251	0.255	-0.556	-0.282			
Temp	0.839	-0.888	0.140	0.253	0.214	0.447	0.441	-0.298	0.304	0.245		
Neutron	0.294	-0.246	-0.018	0.565	0.690	0.428	0.456	-0.715	-0.297	0.571	0.247	
Transmissivty	**-0.166**	**0.114**	-0.061	-0.025	-0.036	0.007	-0.019	0.206	0.192	**-0.141**	0.084	**-0.161**

Table 16. Correlation coefficients for the various geophysical measurments. Correlation was computed between mean values of geophysical logs.

Phase III Simulations: Predictive Neural Networks

Phase II simulations identified optimal subsets of the geophysical measurements for inputs into neural networks (Tables 13 and 14). The focus of Phase III was to attempt to develop an accurate, predictive neural network that could use any of the optimal subsets as input. Each of the subsets was tested.

The data were divided into training and testing sets chosen to reflect a broad range of measured T values. The test set is comprised of 20 sets of geophysical logs and transmissivities taken from a variety of wells at varying depths. The test data set was not chosen by random. Rather, the picks were made to give a wide range of transmissivity values. Table 17 lists the wells and data intervals chosen for the test data. The "Well Interval" number is the hydraulic test interval for the individual borehole counting from the top down and only including the test intervals in the granite. The "Testing Interval" is the sequential number of the interval in the entire test data set and is used for graphing the results.

Well	Well Interval	Transmissivity ($log10(m^2/sec)$)	Testing Interval
DH-12	2	-2.28	1
DH-12	3	-2.91	2
DH-12	5	-3.43	3
MIU-2	4	-4.36	4
MIU-1	17	-4.77	5
MIU-3	15	-5.32	6
DH-9	7	-5.70	7
MIU-3	4	-6.04	8
MIU-3	7	-6.73	9
DH-5	11	-7.08	10
MIU-3	8	-7.17	11
DH-7	7	-7.64	12
DH-8	18	-7.92	13
DH-8	2	-8.01	14
DH-6	11	-8.27	15
DH-5	5	-8.29	16
DH-5	9	-9.42	17
DH-6	17	-9.55	18
DH-6	19	-9.68	19
DH-8	13	-10.33	20

Table 17. Well intervals chosen for the test data set.

Simulation Parameters and Setup

The Phase II simulations were run using the fixed training parameters listed in Table 18. Each optimal subset of geophysical logs (Tables 13 and 14) was used as inputs to a neural network. For each measurement set, a suite of neural network architectures were tried. These included a single hidden layer with 2, 3, 4, 5, 6, or 7 perceptrons, as well as two hidden layer networks with a wide variety of perceptrons in the upstream or downstream layers (*i.e.*, 2-2, 2-3, 2-4, 2-5, 3-2, 3-3, 3-4, 3-5, 4-2, 4-3, 4-4, 4-5, 5-2, 5-3, 5-4, 5-5). Thus, for *each* optimal geophysical log subset, a total of 22 neural network architectures were trained. After each training attempt, the resulting weights were fixed and then the test (unseen) data was run through the network to see what predictive value it had.

Ideally, we are seeking a network that can train to a low mean squared-error value, on the order of 1.0 (1 order of magnitude) or less and then achieve a similar value during testing.

Parameter	*Value*
Random Seed	0.0771695
Initial Learning Rate	0.005
Half Life	50000 epochs
Momentum Parameter	0.9
Maximum Epochs	2000

Table 18. Fixed parameter settings for Phase III simulations.

Results and Discussion

The upper image of Figure 19 presents a typical result for the single hidden layer scoping simulations. The MSE of the training data can be seen to decrease with an increasing number of weights. This behavior is expected, since a network can better fit the data with more parameters to adjust. However, the testing behavior displays a concurrent increase in MSE with network size. This demonstrates that the training network is becoming overtrained such that it is tuned to the training data, but cannot extrapolate to new data. In any case, the MSE achieved by training remains near 4.0, or a mean error (not squared) of 2 orders of magnitude and does not approach a smaller value. This behavior was fairly typical of all the single layer simulations. Simulations with more input values achieved better MSE's during trainng, but this simply reflected the ever-increasing number of weights with which to fit the training data. In all cases the networks were overtrainined and unable to accurately predict the testing data.

The lower image of Figure 19 presents results for the two hidden-layer simulations using 4 inputs. The results are similar to the single layer case, but with more variation in the MSE trends. There is still a significant MSE after training, as well as a decrease in MSE with the number of weights in the network. Likewise, the MSE for testing is higher still and tends to increase with the number of weights in the network. This behavior is typical across all the two hidden-layer simulations.

Of note are two simulations where the training and testing MSE are comparable. These correspond to 2-4 and 2-5 perceptrons in the upstream and downstream hidden layers. These neural networks had 20 and 23 weights respectively. These values are at the upper limit of allowable weights (see discussion above) corresonding to the lower limit of 5 training events per weight. Another series of simulations were performed to see how the 2-4 PE network performed as the number of epochs of training varied. These results are shown in Figure 20. Unfortunately, we cannot achieve significantly lower MSE values with further training. The mean error (not squared) of the training data in Figure 20 is approximately 1.4 orders of magnitude.

Figure 21 presents predicted transmissivity values of the test set as well as the known values for 1000 and 2000 epochs of training. In both cases we see that there is conditional bias in the predictions with high transmissivity values being underpredicted and low values being overpredicted. Figure 22 presents the corresponding training set values. This figure shows that the network is capturing "mean" behavior and losing fidelity at the extremes of the distribution. These results may indicate that the training set contains too few extreme values so that the central values dominate training. Also, the neural network is an estimation algorithm and, as such, provides estimates that are a smoothed representation of the original input data.

Despite concerted efforts in varying training parameters, network architecture and input data sets, we were unable to develop a neural network that could accurately predict all values of an unseen data set.

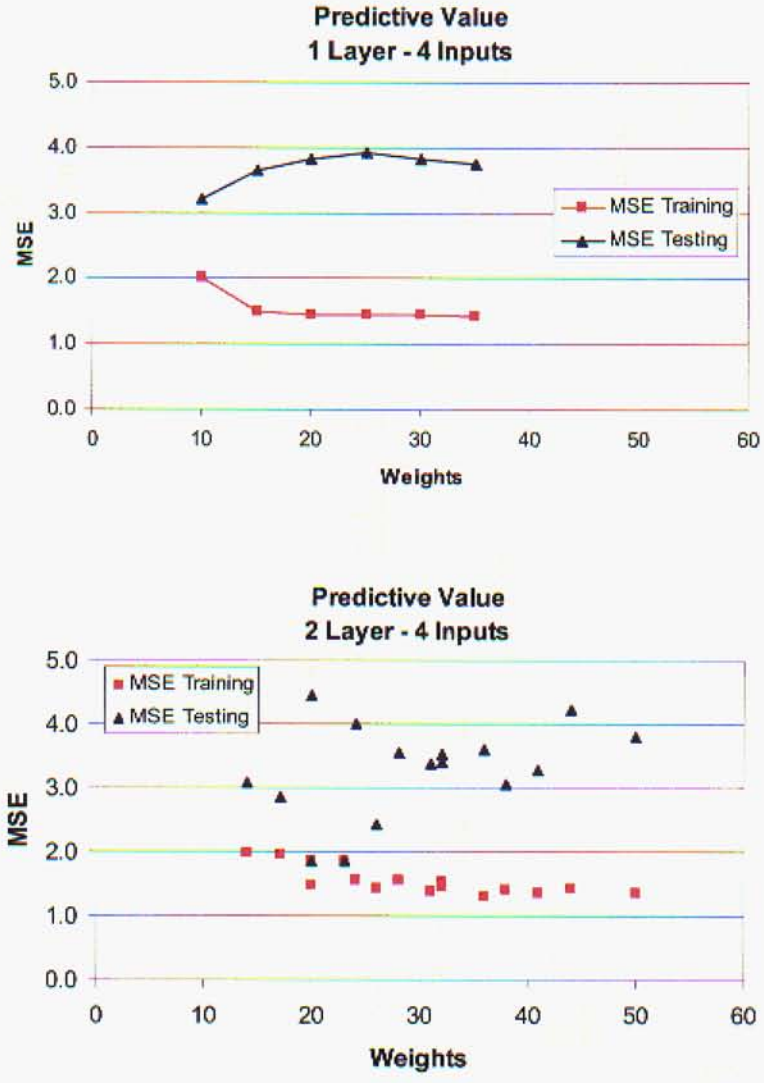

Figure 19. Two examples of the change in MSE for the training and testing calculations as a function of the number of weights in the neural network.

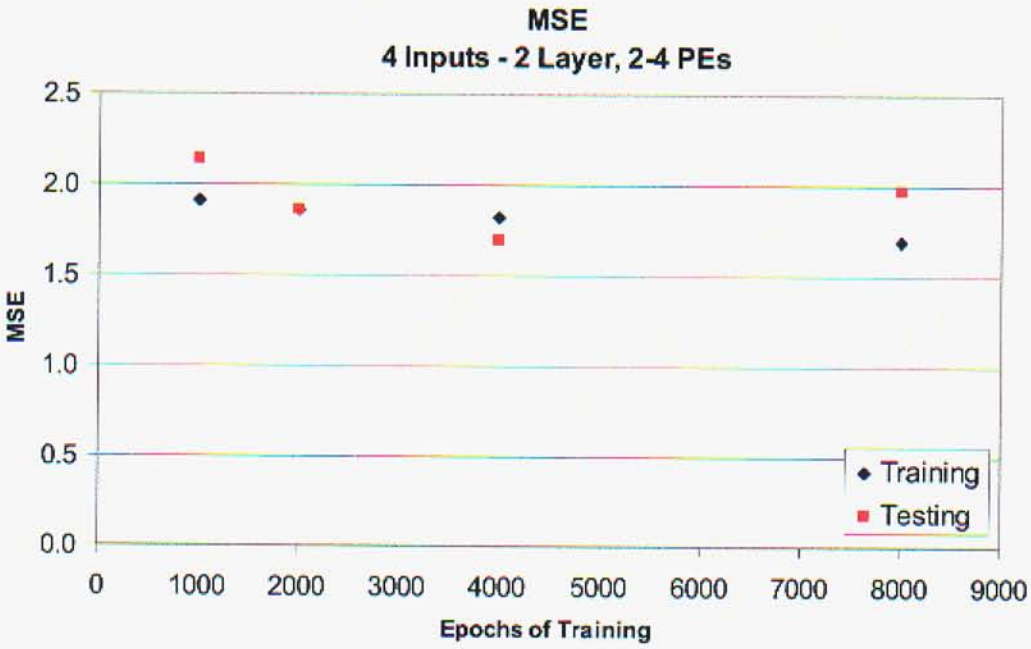

Figure 20. MSE as a function of the number of epochs of training for the 4-input, 2-Layer, 2-4 PE network.

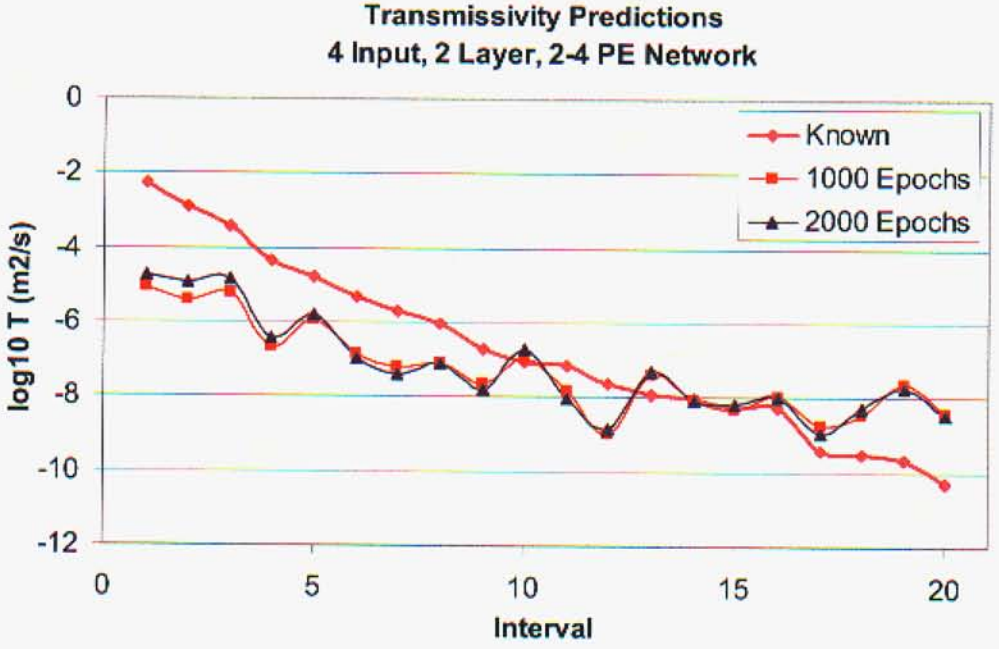

Figure 21. Predicted transmissivity values for the 4-Input, 2-Layer, 2-4 PE network after 1000 and 2000 epochs of training.

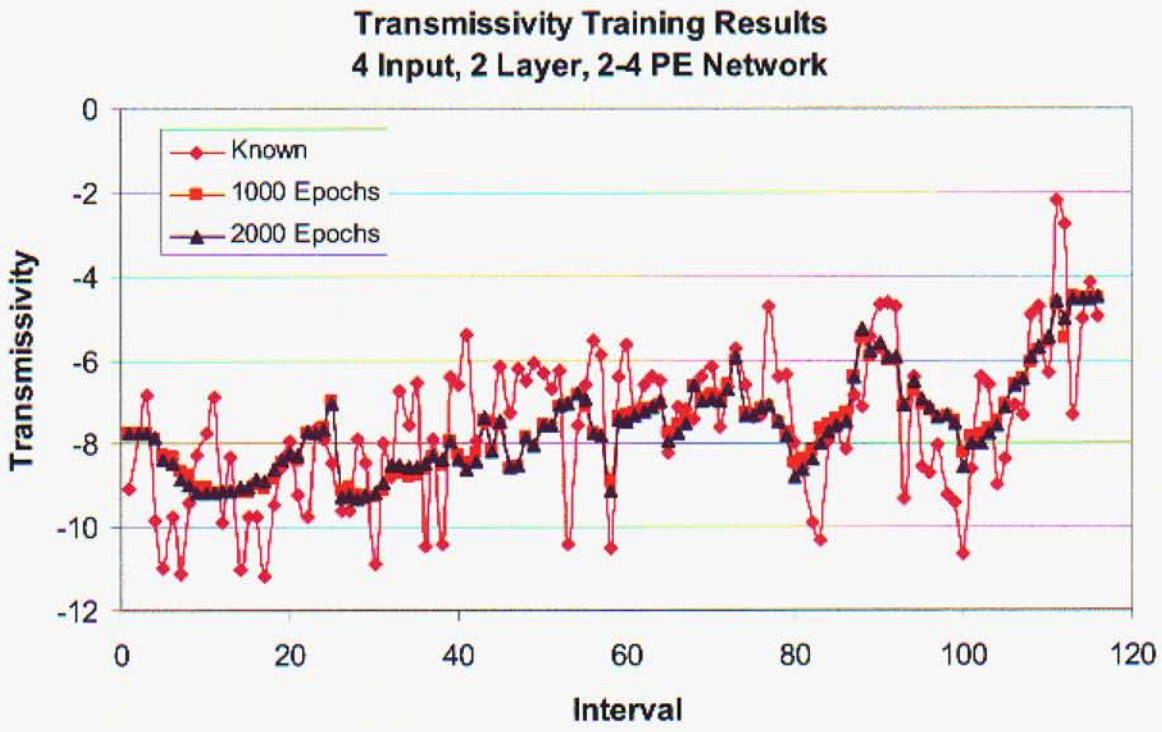

Figure 22. Predicted transmissivity values of the training results for the 4-Input, 2-Layer, 2-4 PE network after 1000 and 2000 epochs of training.

Conclusions

In Phase I we attempted to evolve a neural network, including training parameters and optimal input measurements that could accurately predict the correct class of a set of unseen measurements. Despite a variety of attempts, these calculations were unable to accurately predict the correct hydraulic conductivity class.

The negative results of Phase 1 led us to break the problem into two portions: identifying the most informative set of borehole measurements, and then attempting to optimize a neural network that can use those identified features to predict hydraulic transmissivity. We refer to this work as Phase II. These simulations are a change from Phase I in that the length of the hydraulic test interval is incorporated into the procedure by estimating transmissivity instead of hydraulic conductivity as done in Phase I. Additionally, in the Phase II calculations the actual value of the continuous property is predicted rather than just the class of the measurement. Results of the Phase II calculations show that depth, sonic log, long normal, and self-potential measurements seem to have the highest predictive value.

After identifying the most informative geophysical measurments, we attempted to develop a predictive neural network (Phase III experiments). These results may reflect a

weak physical relationship between the measured values and hydraulic transmissivity (at least with respect to the noise level of the measurements), or they may reflect an insufficient number of data points to train a neural network. An alternative explanation is that the boreholes cannot be grouped together as a single training set due to differing characteristics. We suspect the problem lies with insufficient training data since we saw a relatively clear indication that the best behavior was located at the maximum number of weights relative to the training set size. Perhaps additional measurements would give a large enough data set to train a network to a smaller MSE without overtraining.

Results of this work indicate that neural network prediction of hydraulic conductivity, or transmissivity using geophysical log measurements and the results of packer testing is a difficult problem and more difficult than that of facies prediction or prediction of permeability that is measured on a similar scale to that of the geophysical measurements.

References

Baldwin, J.L., D.N. Otte and C.L. Wheatley, 1989, Computer emulation of human mental processes: Application of neural network simulators to problems in well log interpretration, *SPE Paper 19619*, presented at the 64th Annual Technical Conference and Exhibition of the Society of Petroleum Engineers, San Antonio, Texas, October 8th-11th.

Baldwin, J.L., R.M. Bateman and C.L. Wheatley, 1990, Application of a neural network to the problem of mineral identification from well logs, *The Log Analyst*, Sept-Oct., pp. 279-293.

Doveton, J. and R. Olea, 2001, Petroleum Log Analysis, Short Course Notes, Published by the Kansas Geological Survey, International Association of Mathematical Geology Annual Meeting, Cancun, Mexico

Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Inc., Reading Massachusetts, 1989, 412 pp..

Mohaghegh, S., S. Ameri and R. Arefi, 1996, Virtual measurement of heterogeneous formation permeability using geophysical well log responses, *The Log Analyst*, March-April, pp. 32-39.

Principe, J.C., N.R. Euliano, and W.C. Lefebvre, *Neural and Adaptive Systems: Fundamentals Through Simulations*, John Wiley and Sons, Inc.: New York, 2000, 656 pp.

Reeves, Paul C., POBBLE: A Simple Genetic Algorithm in C++ / JAVA, Sandia National Laboratories, Unpublished, 2001a.

Reeves, Paul C., JUMBLIE: A Feed-Forward Neural Network in C++ / JAVA, Sandia National Laboratories, Unpublished, 2001b.

Rogers, S.J., J.H. Fang, C.L. Klarr and D.A. Stanley, 1992, Determination of lithology from well logs using a neural network, *American Association of Petroleum Geologists Bulletin*, Vol 76(5), pp. 731-739.

Wong, P.M., and S.A.R. Shibli, Use of interpolation neural networks for permeability estimatoin from well logs, *The Log Analyst*, 18-26, 1998.

Distribution List for SAND2003-2950:

Internal

Dept. 6115	MS 0735
MS 0735	Susan Altman
MS 0735	Scott James
MS 0735	Sean McKenna (3)
MS 0735	Ray Finley
MS 0735	Hirotaka Saito

Dept. 6131	MS 0719
MS 0719	Susan Howarth

Dept. 6533	MS 1138
MS 1138	Paul Reeves (3)

Dept. 6805	MS 0731
MS 0731	NWM Library (10)

Dept. 8945-1	MS 9018
MS 9018	Central Technical Files

Dept. 9616	MS 0899
MS 0899	Technical Library (2)

External:

Hitoshi Makino
Tokai Works
Japan Nuclear Cycle Development Institute
4-33, Muramatsu, Tokai-Mura
Naka-Gun, Ibaraki 319-1194 JAPAN

Katsushi Nakano
Tono Geoscience Center
Japan Nuclear Cycle Development Institute
959-31, Jorinji, Izumi, Tokishi
Gifu T509-5102 JAPAN

Hiromitsu Saegusa (5)
Tono Geoscience Center
Japan Nuclear Cycle Development Institute
959-31, Jorinji, Izumi, Tokishi
Gifu T509-5102 JAPAN

Atsushi Sawada
Tokai Works
Japan Nuclear Cycle Development Institute
4-33, Muramatsu, Tokai-Mura
Naka-Gun, Ibaraki 319-1194 JAPAN

Masao Shiotsuki (3)
Tokai Works
Japan Nuclear Cycle Development Institute
4-49, Muramatsu, Tokai-Mura
Naka-Gun, Ibaraki 319-1184 JAPAN

Shinji Takeuchi
Tono Geoscience Center
Japan Nuclear Cycle Development Institute
959-31, Jorinji, Izumi, Tokishi
Gifu T509-5102 JAPAN

Masahiro Uchida
Tokai Works
Japan Nuclear Cycle Development Institute
4-33, Muramatsu, Tokai-Mura
Naka-Gun, Ibaraki 319-1194 JAPAN

John L. Wengle
U.S. Department of Energy, Headquarters
Forrestal Bldg., EM-50
1000 Independence Avenue, S.W.
Washington, DC 20585

Dr. Ahmed Hassan
Desert Research Institute
755 E. Flamingo Road
Las Vegas, NV 89119

Craig Shirley
Desert Research Institute
755 East Flamingo
Las Vegas, Nevada USA 89119