UNDERSTANDING NEURAL NETWORKS AND THEIR APPLICATIONS IN PETROLEUM ENGINEERING

Susan M. Schrader, University of Texas of the Permian Basin
Robert S. Balch and Roger Ruan, New Mexico Petroleum Recovery Research Center

ABSTRACT

Artificial neural networks (ANN) are computer programs designed to mimic the functioning of the human brain. ANNs can be designed to "learn" by reviewing a data set consisting of a set of known inputs and a corresponding set of desired outputs. Once an ANN has been trained, it can predict outputs given just the set of known inputs. While the applications are broad reaching, one valuable application of such tools is in petroleum exploration. In places where both exploration data (such as log, core or geophysical data) and production data are available, a network can be designed and trained, and then used to predict production given a similar suite of exploration data. This work will discuss design issues in exploration neural networks, explore available software, and review two case studies where neural networks were used to predict total production for undrilled sites in two formations in the Permian Basin.

INTRODUCTION

Artificial neural networks (ANN), part of a series of techniques sometimes referred to as soft computing or artificial intelligence, are computational models designed to mimic biological neurons. They consist of a network of simple elements that create a function which can be trained to a specific task by adjusting the weights between elements. While there are a variety of types and techniques associated with neural networks, a basic approach involves providing a complete set of data, with both input variables and output variables, and using these data to develop, train and test a neural network. Once the network has been completed, a second set of data with just the input variables is presented to the ANN, which then predicts values for the output variable.

The applications for ANNs are varied and cross a wide variety of disciplines and industries. The technology itself has become very accessible, with a number of software packages that can be set up quickly to apply neural networks to a number of different problems. With access to such a package, a basic understanding of the principles of neural network design, and appropriate data, a person can easily develop a network to apply to their applications. One area that seems to be well suited to neural networks is petroleum exploration. In many petroleum exploration problems, there are a lot of data available for potential inputs, including geophysical, well log, and core data. The desired output is often some type of predicted production.

ARTIFICIAL NEURAL NETWORKS

The human brain consists of a network of biological neurons and synapses. Each neuron has a nucleus in the cell body with numerous branches called dendrites and one large single fiber termed the axon. At the end of the axon and dendrites are small substrands that contain the transmitting or receiving ends of a synapse. A series of signals are received by the neuron through the receiving ends of the synapses and transmitted through the dendrites to the nucleus. If the signals exceed threshold strength, the neuron "fires" and a signal is sent through the axon to the transmitting end of the neuron's synapse.

An artificial neural network consists of units or neurodes, which mimic the biological neurons, and weights or connections, which mimic the biological synapses. Each neurode receives a weighted average of the outputs from the proceeding neurodes, and these become inputs to the neurode's activation function (analogous to the threshold function of a biological neuron).[1] If the activation function 'fires', then the output becomes part of the input of the next neurodes. An artificial neural network architecture is typically developed by designing a structure with the individual neurodes as rows and the connections as lines connecting the neurodes. There is an initial layer which contains a point for each input variable, at least one layer containing a number of neurodes, and a final layer containing the output variable or variables. (Figure 1) For instance, a network designed to use four input variables to predict an output variable, using five neurodes in three layers (two hidden layers) would have a 4-3-2-1 architecture with 20 connections.

ANNs can learn in a variety of ways, including both unsupervised and supervised methods. Unsupervised learning allows the network to look for patterns in the data without labeling outcomes as correct or incorrect. It is often used in data mining in the form of clustering algorithms. In supervised learning, the network is first provided with a set of training data in which all of the inputs as well as the required output are known. When presented with this data, the network “develops” by calculating values of the weights until a desired correlation is reached. Then the neural network is tested by providing only the input portion of a data set for which the desired outputs are known. If the network outputs can match the known outputs to a desired degree of correlation, the network is said to be trained.

DESIGN ISSUES

The issues involved with designing a neural network include selecting the type of learning, determining the number of neurodes to use, determining the number of layers, selecting the activation function and choosing appropriate input variables. One of the main concerns is in the area of overtraining, in which a complex network is developed that perfectly matches the desired outputs in the training set, but misses the overall trend (Figure 2). Research has shown that in order to prevent overtraining, the network size and complexity must be related to the amount of data used for training, in other words, the fewer records available for training, the less complex the network. Heuristic rules range from having one tenth as many connections as records to having half as many connections as records.[2,3] As an example, if there are 200 records (a record is the values for all input variables plus the associated output variable(s) at a point), there can be between 20 and 100 connections in the neural network. The 4-3-2-1 network defined above has 20 connections. Since using more inputs will require more connections, part of the network design involves selecting the best available inputs. With the input data selected, a training data set containing the inputs and output(s) is built. By using the number of records in the training data to set limits on the number o f connections, a number of acceptable network architectures may be tried. Each potential neural network is trained with the training data and then tested by using a portion of the training data held aside for this purpose. The neural networks are evaluated based on their R^2 value, a type of correlation coefficient. The closer this value is to 1, the better the network was at predicting the test outputs. When a network with a high correlation coefficient is found, the testing and training process may be repeated. If the results are consistent, then the network is developed and ready to be used to predict outputs from a set of data containing only the inputs.

SOFTWARE

There are many software options available for the construction and use of neural networks. Commercial computer algebra systems, such as Matlab, often have add-ons or toolboxes for creating neural networks. There are also stand-alone programs, including some public domain programs, for neural network development.[4] The simplest programs provide a single activation function, are fully connected, and may only allow one hidden layer. More advanced programs allow the user to select the activation functions for each neurode, develop multi-layer architectures, and choose where the connections will go.

PredictOnline (v6) is an example of a public domain neural network program that allows the user to design the architecture of a neural network, train the network, and then use it to predict the desired output.[5] The user designs the architecture by selecting the number of hidden layers and the number of neurodes in each layer, then selects the percentage of training data to use to test the network. An example screenshot of the PredictOnline user interface is given in figure 3. PredictOnline was the software used in the following case studies and is still available for interested users.

CASE STUDY 1 – DELAWARE BASIN

The lower Brushy Canyon formation is a sandstone formation in the lower end of the Delaware Mountain group, with areas of oil production found in southeastern New Mexico.[6] With data available from many of the producing wells drilled into this formation; it was possible to design a neural network to generate a map of potential productive locations.

The region was gridded to a 40-acre spacing with a total of 60,478 gridpoints, and all available input data was interpolated with krigging algorithms to the same grid. Data were also available for 2434 wells drilled into this region. This well production data provided the desired outputs for training of the neural network. The available input data was primarily geophysical, and could be described in four major categories: gravitational, Aeromagnetic, structural and thickness. In addition to a value for each of the four variables at every gridpoint, additional attributes were also computed, including first and second derivatives along the latitude and longitude, dip azimuth and magnitude and curvature azimuth and magnitude. With these additional attributes, 36 variables were available to use as input variables for the neural network.[7]

Using production data expressed as barrels of oil per month (BOPM) for the producing wells, a neural network was developed and trained. The first step in the process was to select out of the 36 input variables a reduced number to use in the network. In this case, a fuzzy ranking algorithm was used to select four relevant variables to use as inputs.[7] These variables were the dip azimuth of gravity, second latitude derivative of thickness, longitude derivate of gravity and longitude derivative of structure. Out of the producing well data, 520 wells, including some with zero production, were selected for training and blind testing. The final network architecture was 4-10-10-10-1, with three hidden layers of 10 nodes each. The correlation coefficient for training was 0.9 and for the blind test data was 0.81. The trained neural network was then applied to all 60,478 gridpoints. A map showing areas with high predicted production is given in figure 4.

CASE STUDY 2- DEVONIAN CARBONATES

The Devonian Carbonates zone in southeast New Mexico is a structural formation dating from the Silurian and Devonian period. Three important characteristics that correlate to production from this formation are the thickness, organic richness and thermal maturity of the Woodford Shale, the principal source rock for the formation.

With this in mind, a database was developed for the region that included the following variables:

- Woodford Shale Thickness
- Subsea Elevation (to top of the Woodford Shale)
- Total Organic Carbon (TOC)
- Production Index (PI)
- Generative Potential (GP) (defined for this project as Thickness·TOC·PI)
- Log (GP)
- Permeability (in md)
- Curvature (calculated curvature of subsea elevation)
- Structure
- Closure
- Structural Relief
- Paleostructure (calculated from Woodford and Abo formation tops)

The variables Woodford Shale thickness, subsea elevation, TOC, PI, and permeability were recorded at producing wells throughout the region[3], and then interpolated for the entire region using a krigging technique. The remaining variables were calculated from these variables. The result was a database that included the values of these eleven variables at each of 64,347 gridpoints (each gridpoint corresponds to a 160 acre square in the region).

To train the neural network, a data set containing both the selected input variables and the corresponding production had to be developed. Production data was compiled for 172 wells completed to this formation. This data included gas wells, oil wells, wells with mixed production and unsuccessful wells. For each well, production was measured in barrels of oil equivalent per month, averaged over the first producing year (using 6mcf = 1BOE). Of the wells used, 105 were unsuccessful, 15 had production less than 1000 BOEPM, 31 had production between 1000 and 5000 BOEPM, and 21 had production over 5000 BOEPM with a maximum of 38,970.[8]

The next step in the development of this network was to determine which of the eleven variables to use as inputs. With only 172 records (corresponding with the 172 wells) using 11 variables would likely lead to overtraining. On the other hand, if only one or two variables were sufficient, a neural network would not be the best choice. In order to select the most relevant data to use as input variables, two methods were used; a fuzzy ranking algorithm and a linear correlation. It was determined that four input variables would be used. The fuzzy ranking algorithm selected two, Woodford Thickness and Total Organic carbon, and linear algorithms were used to select the other two, Permeability and Generative Potential.[8]

A systematic approach was used to develop the neural network. First, a training data file containing the four selected inputs and related outputs at the 172 wells was uploaded to PredictOnline. With 172 wells, an acceptable network should have between 17 and 86 connections to avoid overtraining. The approach chosen to find the best network in this range was to start out with simple networks and add complexity until a desired and repeatable correlation coefficient was reached. Beginning with a 4 – 1 – 1 architecture, the process of training the network progressed until a network of 4-6-1 was selected. This network was trained multiple times to prevent it from stopping at a local minimum. The 4-6-1 neural network that was used for prediction had an R^2 value of 0.84 and a correlation coefficient of 0.92.[8] Once the network was trained, a predict file containing the values of the input variables for the entire region was uploaded.

When the 4-6-1 neural network returned the values of predicted production, the results were evaluated first by mapping them over the region. The producing wells and dry holes were then overlain over the contour map of predicted production. The data was then filtered by zeroing out production in a small section of the region where the Woodford shale was not present. Descriptive statistics of the predicted production were then computed, and a histogram was produced (Figure 5). As another means of testing the data, the predicted values at the 172 well locations were correlated with the actual values at these locations, resulting in a correlation of 0.79.[8] Finally, a contour map was produced, showing the predicted values, the producing wells and the dry holes. (Figure 6)

CONCLUSIONS

With the availability of software packages, it has become possible to apply ANNs to a variety of problems. For a supervised learning algorithm, data sets that include the selected inputs and the desired output(s) are used to train and test the network. The completed network can then be used to predict outputs given values for the input variables. Applications in petroleum engineering include creating ANNs to serve as exploration tools. They often provide a quick way to look at a large quantity of data, allowing the engineers to focus on narrower areas.

REFERENCES

1. Hertz, J., Krogh, A., and Palmer, R.G.: *Introduction to the Theory of Neural Computation*, Addison-Wesley, Redwood City, CA 1991
2. Statsoft: *Electronic Textbook, Neural Networks*, http://www.statsoft.com/textbook/stneunet.html, 2003
3. Du, Y.: *Optimization of Artificial Neural Network Design through Synthetic Datasets Analysis*, M.S. Thesis, New Mexico Institute of Mining and Technology, May 2002
4. Demuth, H., Beale, M. Hogan, M.: *Neural Network Toolbox User's Guide v5*, The MathWorks, Natick, MA, 2006
5. Schrader, S.M.: *REACT Software User's Guide*, PRRC, New Mexico Tech, December 2004
6. Broadhead, R., and Justman, H.: "Regional Controls on Oil Accumulations, Lower Brushy Canyon Formation, Southeast New Mexico" in *The Permian Basin, Proving Grounds for Tomorrow's Technology*, West Texas Geological Society, 2000
7. Balch, R.S., Hart, D., Weiss, W.W. and Broadhead, R.F: "Regional Analysis to Better Predict Drilling Success, Brushy Canyon Formation, Delaware Basin, NM," paper SPE 75145 presented at the SPE/DOE Improved Oil Recovery Symposium, Tulsa, OK, April 2002
8. Schrader, S.M., Balch, R.S., and Ruan, T: "Using Neural Networks to Estimate Monthly Production, A Case Study for the Devonian Carbonates, Southeast New Mexico," paper SPE 94089 presented at the SPE Production and Operations Symposium, Oklahoma City, OK, April 2005

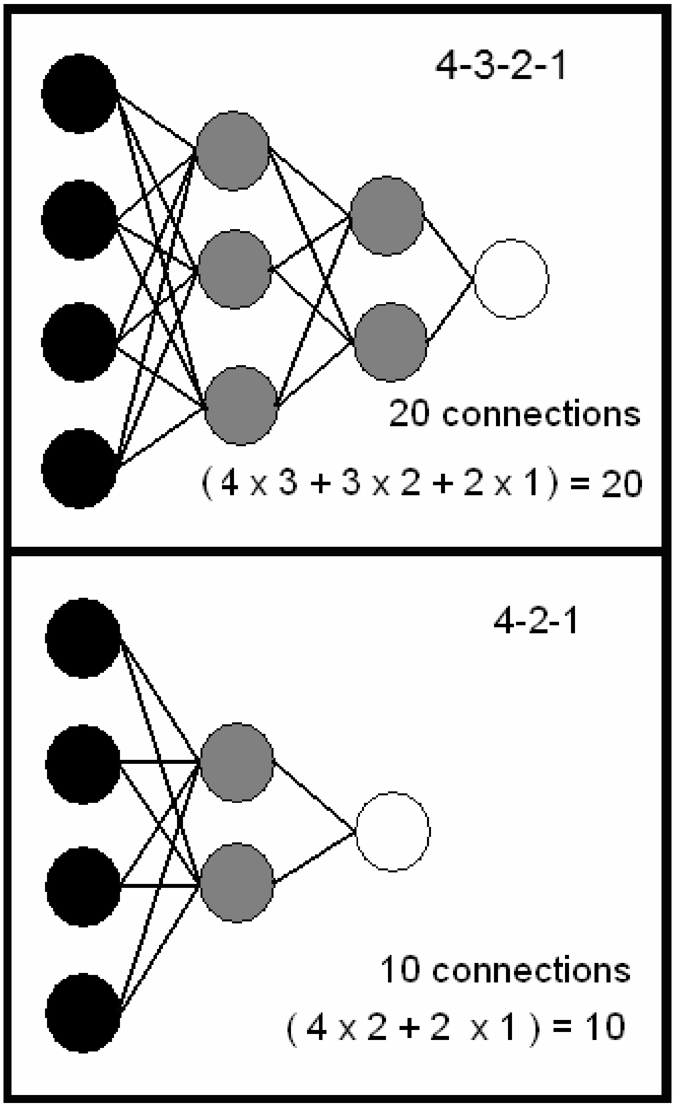


Figure 1 - Examples of two neural network architectures and the number of connections in each.

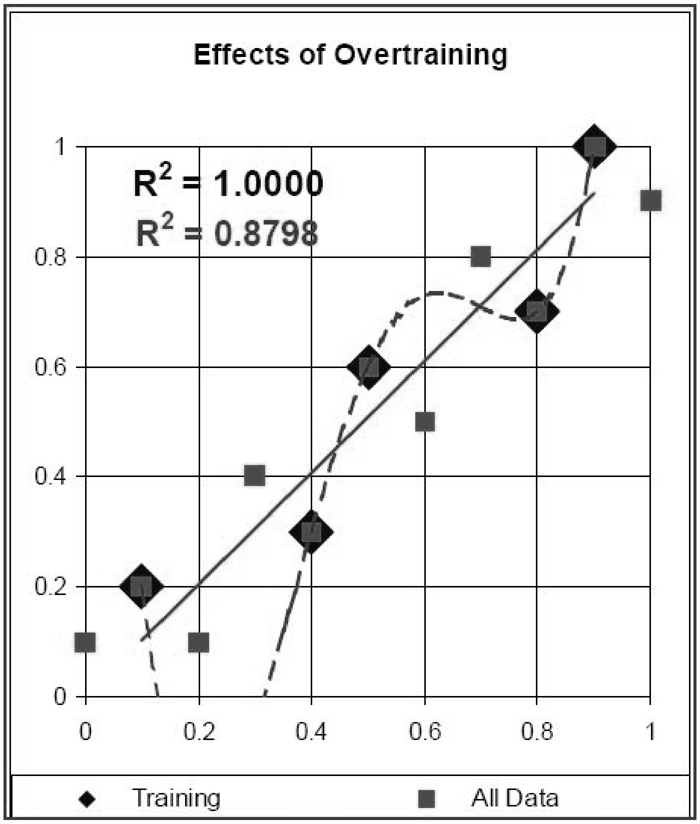
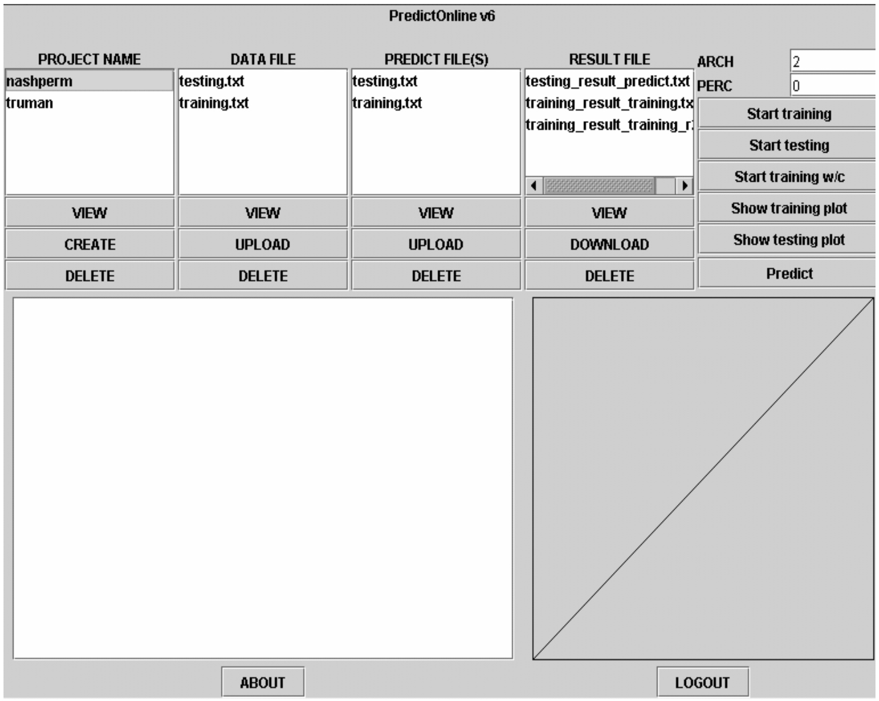


Figure 2 - Illustration of the problem of overtraining. When presented with the small training data set, a complex network is developed that misses the major trend of the full data set.



PROJECT NAME	DATA FILE	PREDICT FILE(S)	RESULT FILE
nashperm	testing.txt	testing.txt	testing_result_predict.txt
truman	training.txt	training.txt	training_result_training.tx
			training_result_training_r:

ARCH 2
PERC 0

Figure 3 - Screenshot of the PredictOnline neural network software tool. The architecture and percentage of data held aside for testing is set in the upper right hand corner.

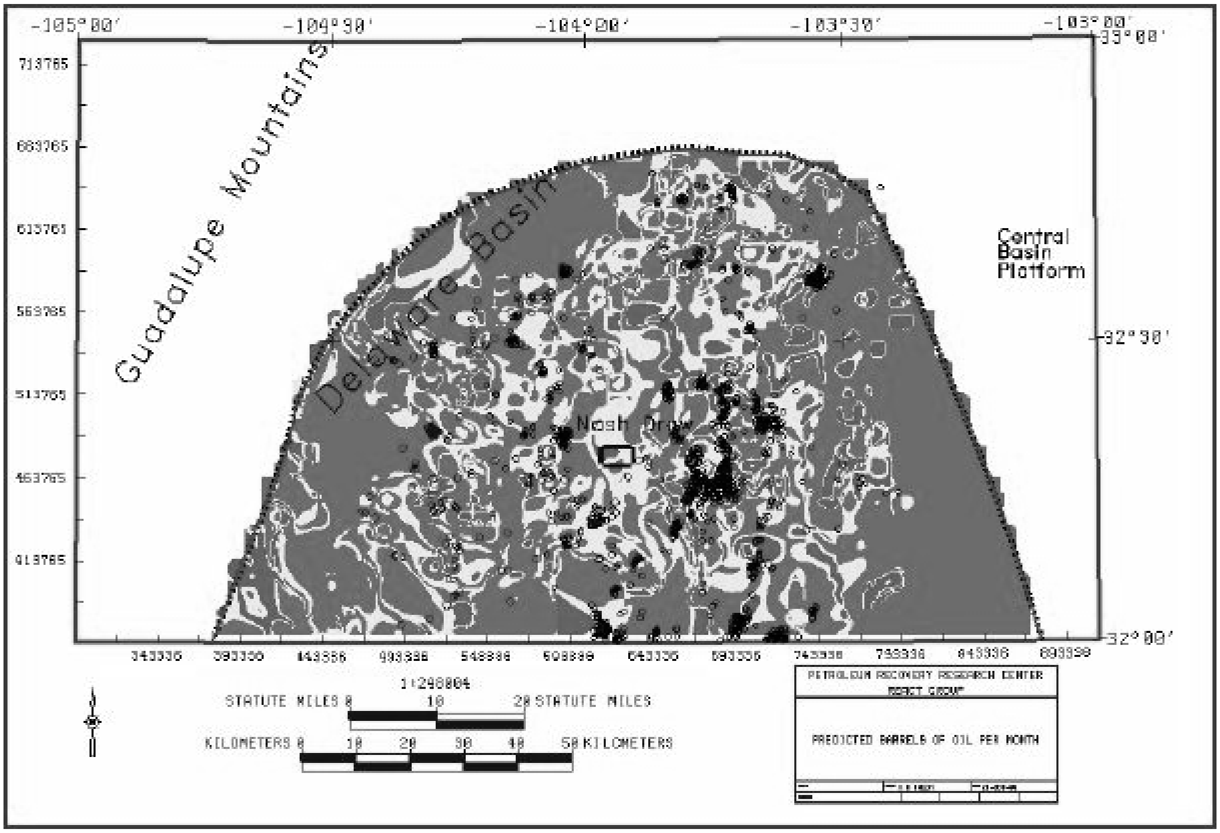


Figure 4 - Map of the outputs from the neural network developed to predict production for the Lower Brushy Canyon formation.

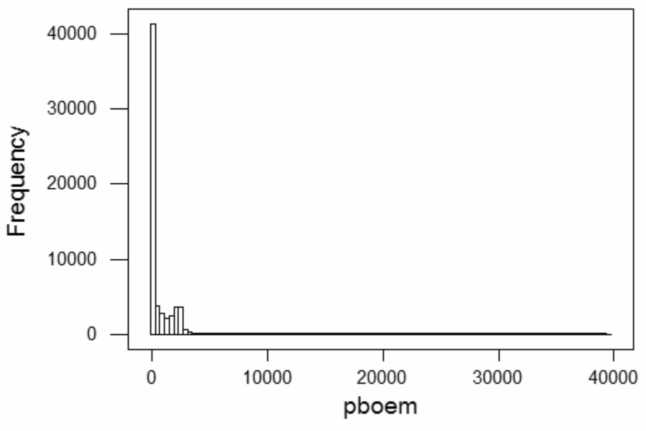


Figure 5 - Histogram for the variable pboem (predicted barrels of oil equivalents per month) as given by the neural network developed for the Devonian Carbonates.

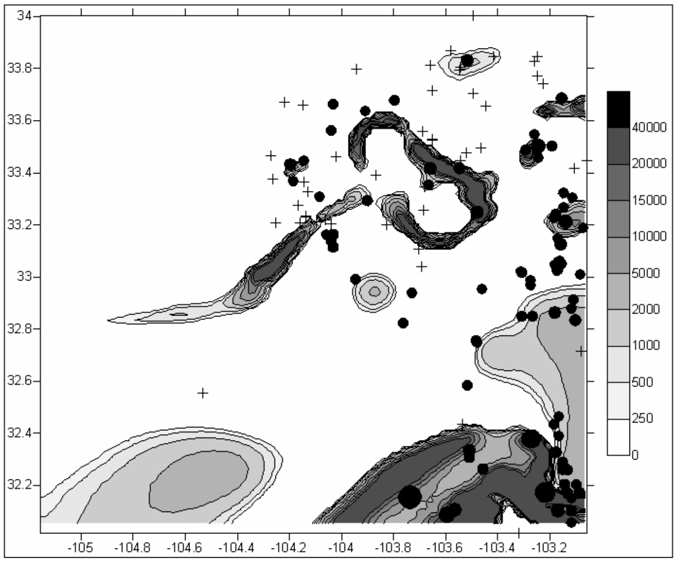


Figure 6 - Contour map of the predicted production overlain with the producing wells (dots) and unsuccessful wells (+) in the Devonian Carbonate formation.