



# Quantifying Data Needs for Deep Feed-forward Neural Network

## Application in Reservoir Property Predictions

Tanya Colwell

[cgg.com](http://cgg.com)





# “Having enough data, statistically one can predict anything”

“99 percent of statistics tells only 49 percent of a story” Ron DeLegge II

In statistical learning we establish a hypothesis first, while in machine learning the predictions are derived without a prior assumption and only from the training data given (supervised machine learning)

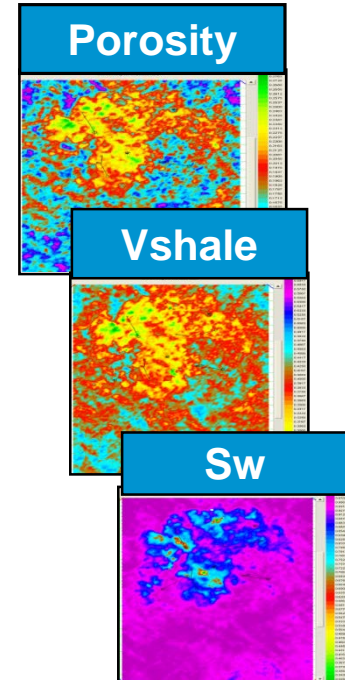
Some of the aspects that affect the accuracy of predictions are:

- Quality of data and sampling errors
- Degree of variance in sampling
- Size of data sampling

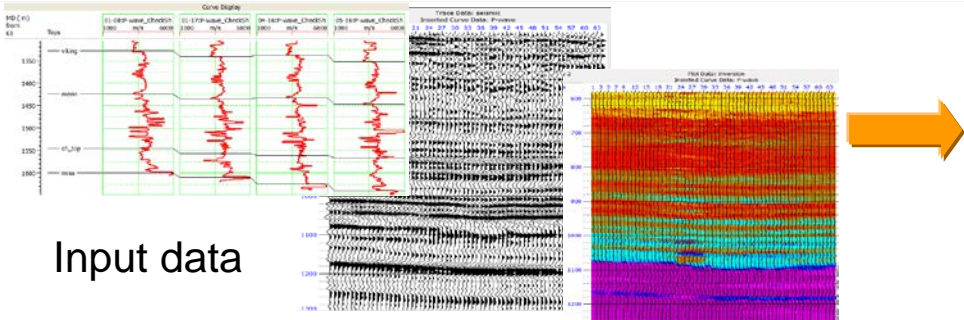


# Outline

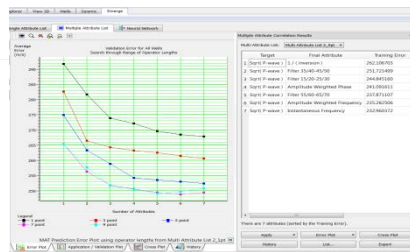
- Introduction: usage of neural networks in reservoir property predictions
- Deep Feed-forward Neural Network
- Validation and Parameterization
- Data requirements and adding synthetic data
- Summary



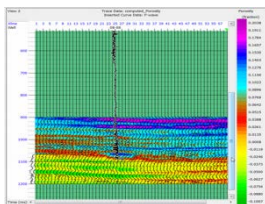
# Emerge Workflow



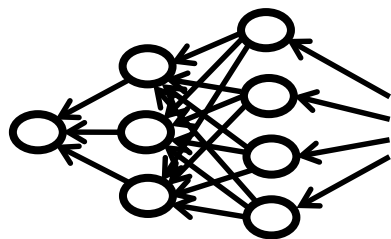
Input data



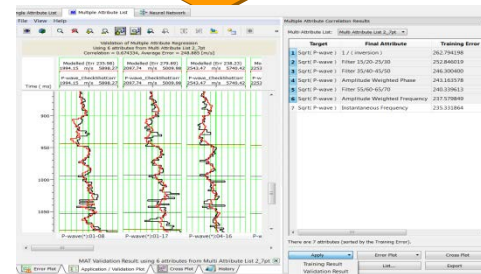
Multi-linear Regression Training



Predicted volume of log properties



Neural Network Training

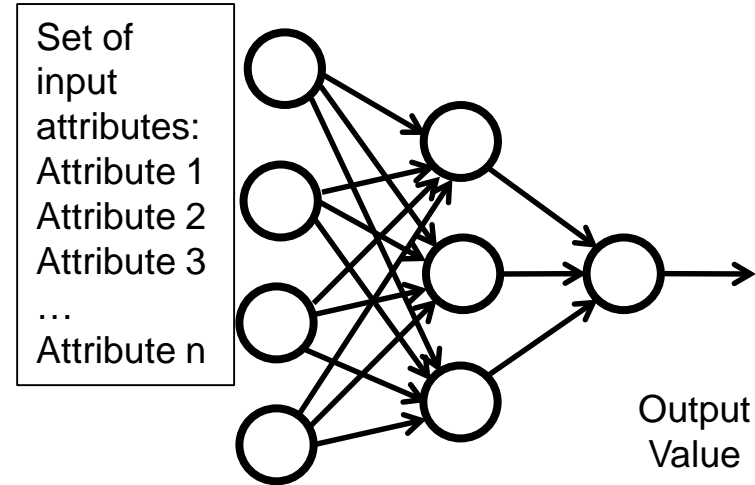


Validation Analysis

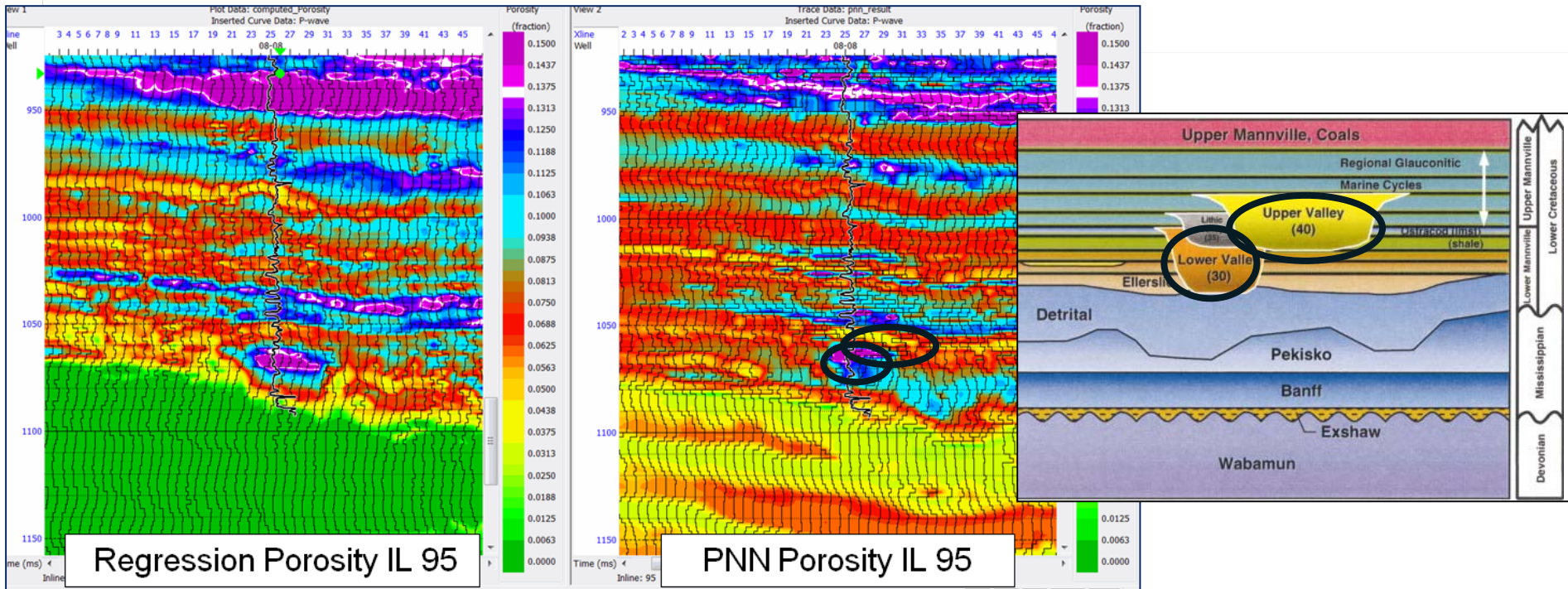


# Introduction: Neural Networks for Property Predictions

- Neural Networks have been used for a number of years to predict seismic reservoir properties from well data and seismic attributes.
- HampsonRussell Emerge has the ability to find and apply both linear and nonlinear models. Nonlinear solutions include Probabilistic Neural Networks (PNN) and Multi-Layer Feed-forward Networks (MLFN). The new addition is DFNN.
- Neural networks can produce better predictions than traditional multi-linear regression since they account directly for non-linear relationship between logs and attributes.



# Introduction: Porosity Prediction example



\* J.Dufour, "Integrated geological and geophysical interpretation case study, and Lamé rock parameter extractions using AVO analysis on the BlackFoot 3C-3D seismic data, Alberta, Canada"





## Deep feed-forward Neural Network (DFNN)

We implemented and studied DFNN, which is a supervised neural network. The supervised learning is the task of inferring a function from labeled training data. The learning algorithm then generalizes from the training data to unseen situations. The resulting model is statistical.

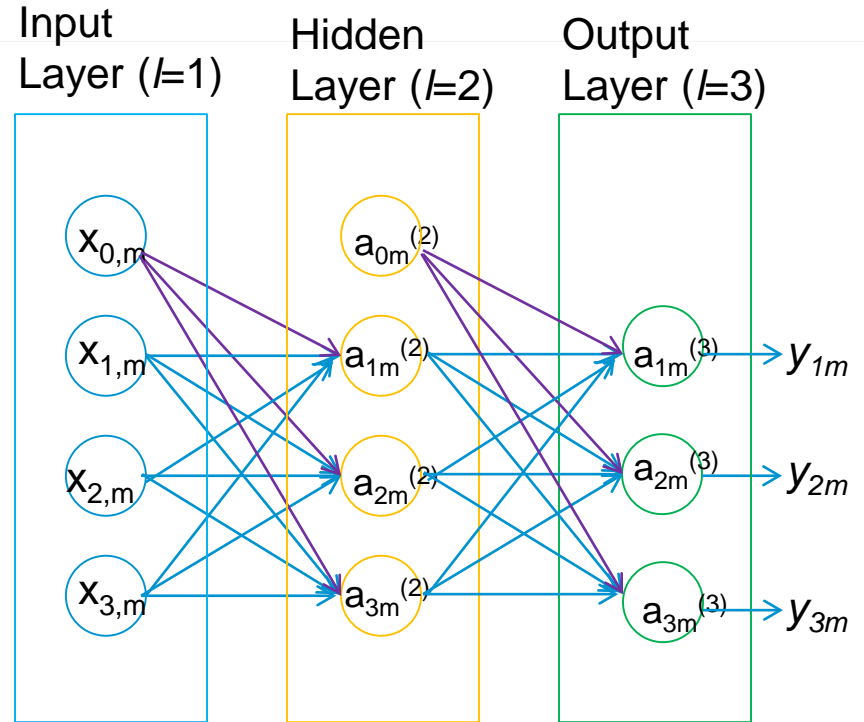
A multi-layer neural network is considered **deep** if it has 2 or more hidden layers. As the number of hidden layers increase, a deep forward network can model more complexity, 8-10 layers can simulate any non-linear function. The greater the number of hidden layers, the greater the amount of training data required.

Depending on the amount of well control available this typically limits the training data set to be in the order of hundreds of points. This practically limited the depth of the neural network and the adoption of DFNNs for reservoir geophysics.



# Deep Feedforward Neural Network (DFNN)

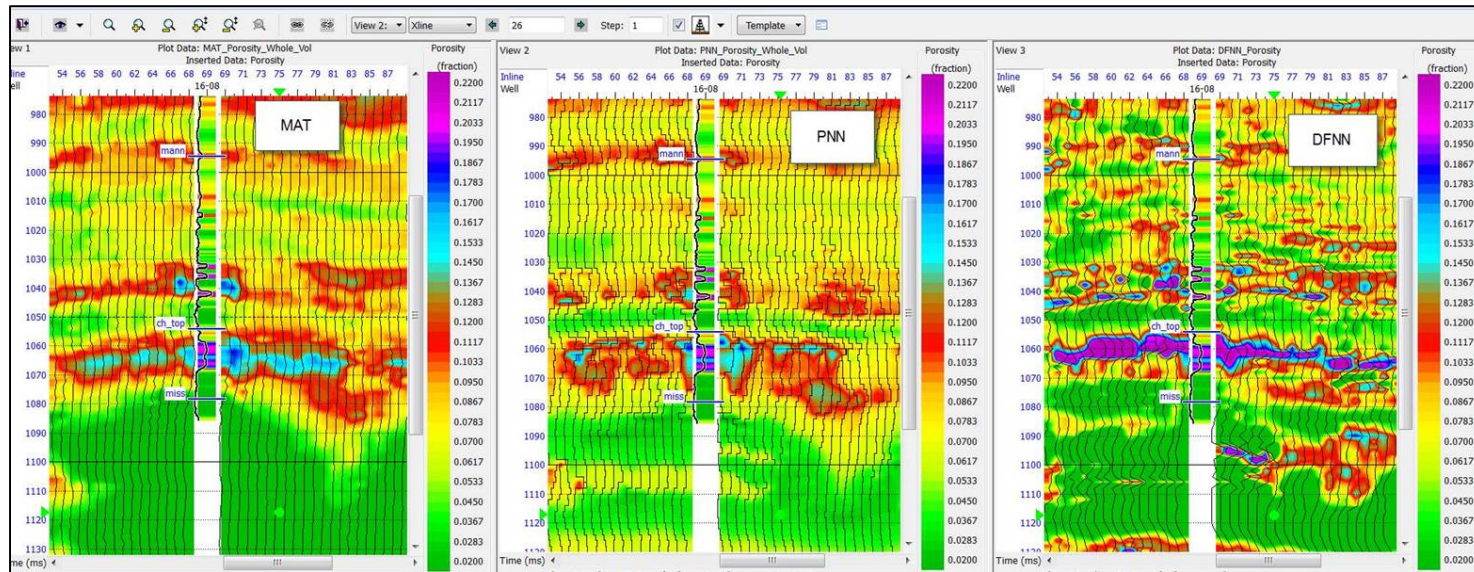
- The Deep Forward Neural Network (DFNN) is an extension of the Multi-Layer Feed Forward Network (MLFN).
- The output of the first layer is hidden from the user so it is called a hidden layer.
- We can combine many networks in series to create a multilayer network.
- Extra layers allow the network to model transforms such as higher order polynomials.





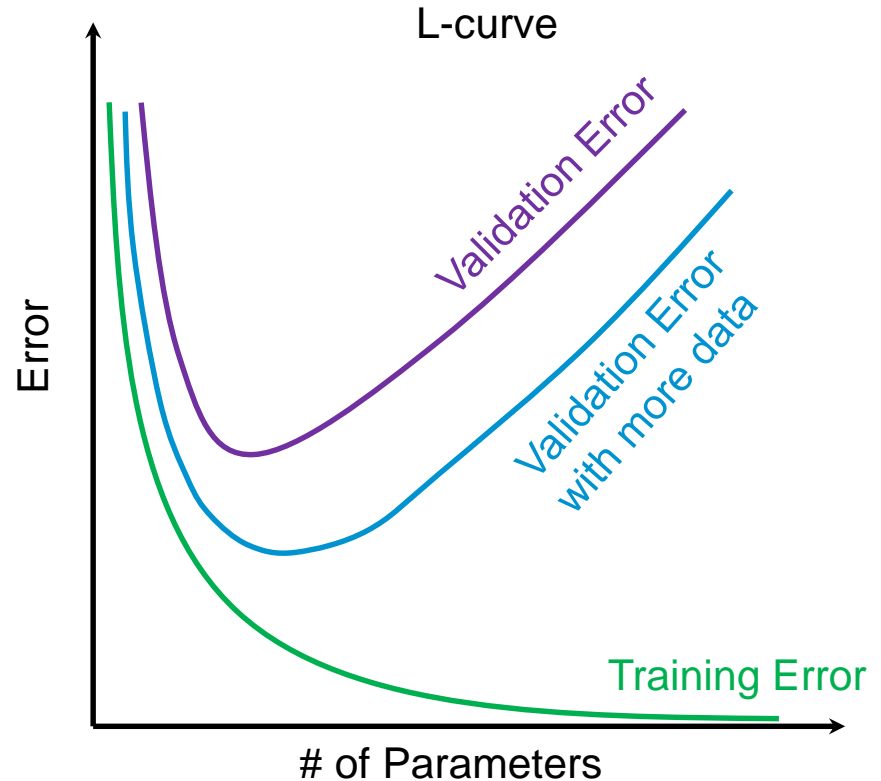
# Comparison of MAT, PNN, DFNN prediction

- DFNN provides more accurate predictions and has faster run-times in comparison to the Probabilistic Neural Network (PNN).



## Too many parameters?

- Training the DFNN is the process of determining the optimal set of weights.
- The weights are solved as a large nonlinear inverse problem using iterative techniques.
- To ensure the network is not over trained the network is tested on a separate validation dataset.
- Deep neural networks have many layers and parameters, increasing the risk of overfitting.
  - Overfitting is characterized by observing a small training error and a large validation error

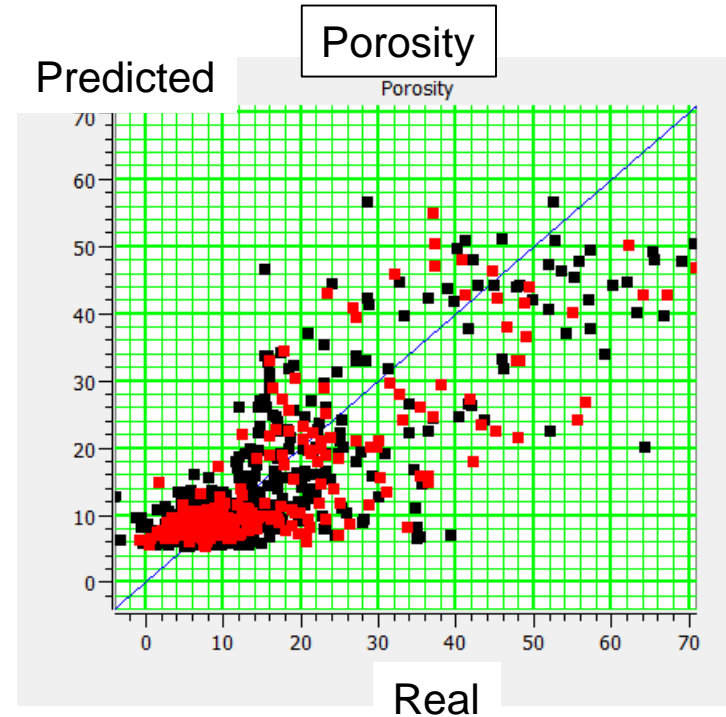


# How much training data is needed?

To quantify the data requirements, we try to quantify what determines a successful prediction of the data.

We use the validation procedure to measure the success of training, specifically the percentage-based validation.

In the %-based validation process, a subset of the original training data is removed. The selection process is controlled by a random number algorithm. The DFNN is re-trained on the reduced training data and applied to the hidden subset.



*Validation plot: red are the validation samples and black are the training samples*



# DFNN Parameter Control

DFNN Parameters

Number of Hidden Layers:	3
Nodes in Hidden Layers:	20
Minimization Option:	SD
L2:	0.100000
L1:	0.000000
Total Iterations:	200
Eta:	0.001000
Alpha:	0.001000
Decrease Constant:	0.000010
Number of Mini Batches:	50
<input checked="" type="checkbox"/> Shuffle Option	
Random State:	1

DFNN offers significant advantages in terms of control of training parameters and speed of application.

Each of the parameters shown on the left affects the accuracy of prediction.

Let's look specifically into these parameters:

- Number of hidden layers
- Number of nodes in a hidden layer
- Total number of iterations



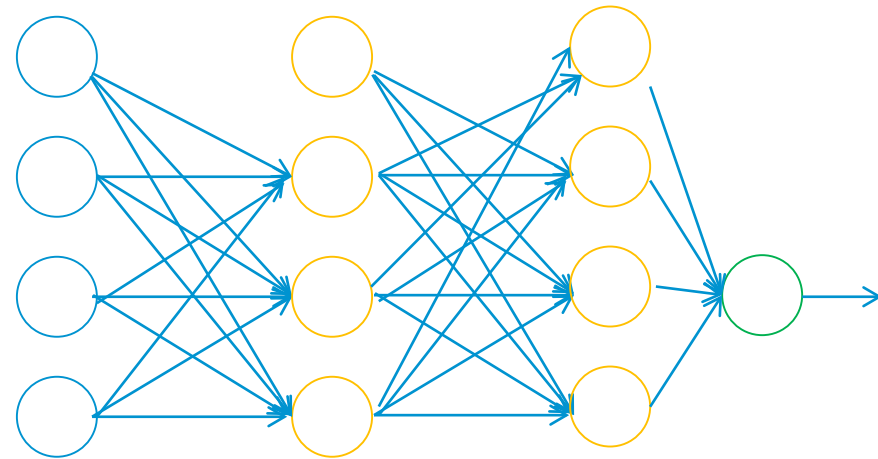
# Modifying the depth of DFNN: Number of Hidden Layers

As the number of hidden layers is increased, the network has increasing number of weights with which to predict the training data.

Hence, increasing the number of hidden layers generally reduces the training error, while potentially increasing the validation error.

If too many layers are specified there is not enough data to uniquely determine the weights, in this case the regularization terms will drive the weights to zero.

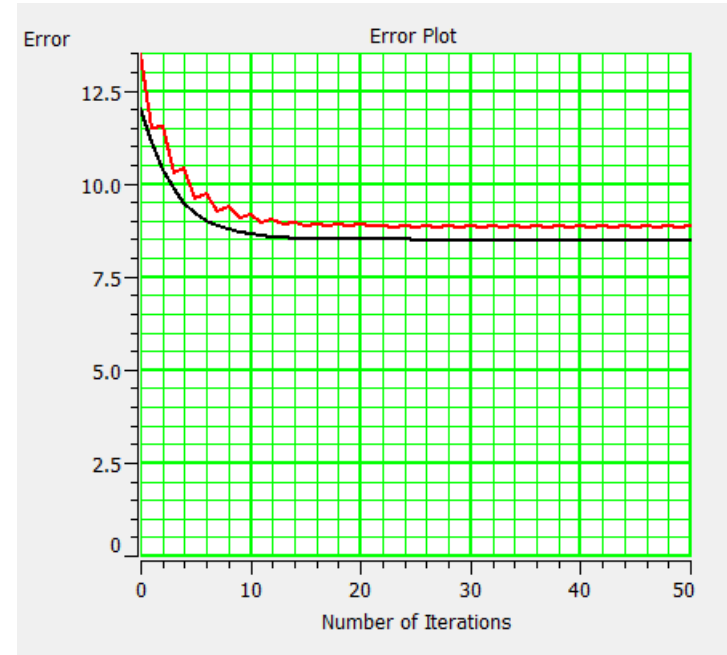
Input Layer ( $l=1$ )      Hidden Layer ( $l=2$ )      Hidden Layer ( $l=3$ )



## Testing the number of iterations

This parameter sets the total number of iterations or steps which will be used for either the Conjugate gradient (CG) or Steepest Descent (SD) algorithm.

This is the main control which the user has to balance the conflicting goals of simultaneously minimizing the training error and the validation error.



## What to do in case of lack of data?

In order to obtain more training data we explore the use of synthetic seismic data derived from perturbations from the known well control.

Two approaches have been investigated:

- Workflow 1: generate new wells using systematic changes
- Workflow 2: generate new wells based on adding statistical variations to the calibrated rock physics relationships.

For example, new wells are created for which the reservoir thickness, porosity and fluid content are varied.

Synthetic seismic gathers are then generated for each of these new wells. These data are then used to train the DFNN.



# Workflow 1: create wells and synthetics using systematic changes

The screenshot shows two panels from a software interface. The left panel, titled 'Reservoir', has tabs for Velocity, Density, In-Situ Fluid, Matrix, Output, and History. It shows 'Well\_1' selected. Under 'Reservoir Interval', the 'Start' section has 'Top' selected with a 'Pay' button, and the 'End' section has 'Top' selected with a 'Base Pay' button. A red box highlights the 'Start' section, and another red box highlights the 'End' section. A red arrow points from this panel to the right panel. The right panel has tabs for Reservoir, Velocity, Density, In-Situ Fluid, and Matrix. It shows 'Fluids' selected, with sub-tabs for Special Parameters, Calculated Properties, and Gr. Under 'Saturations In-situ', 'Water 30 %' and 'Gas 70 %' are checked, while 'Oil 0 %' and 'CO2' are unchecked. A red box highlights these saturation settings. A red arrow points from this panel to the bottom panel.

The screenshot shows the 'Basic' tab of a software interface. It has sections for 'Inline Changes' and 'Xline Changes'. The 'Inline Changes' section has fields for Parameter, Start Value, End Value, Increment, and Unit, with a checkbox for 'Use percentage for changing'. The 'Xline Changes' section has fields for Parameter, Start Value, End Value, Increment, and Unit. A red box highlights the 'Gas Saturation' dropdown menu, which is set to 'Gas Saturation'. Below it are input fields for '0', '100', and '20', and a dropdown menu set to '%'. Another red box highlights the 'Gassmann - Porosity' dropdown menu, which is set to 'Gassmann - Porosity'. Below it are input fields for '1', '29', and '9', and a dropdown menu set to '%'. A red arrow points from the saturation settings in the previous panel to this section.

This creates 24 wells with these combinations:

Porosity: 1, 10, 19, 28 %

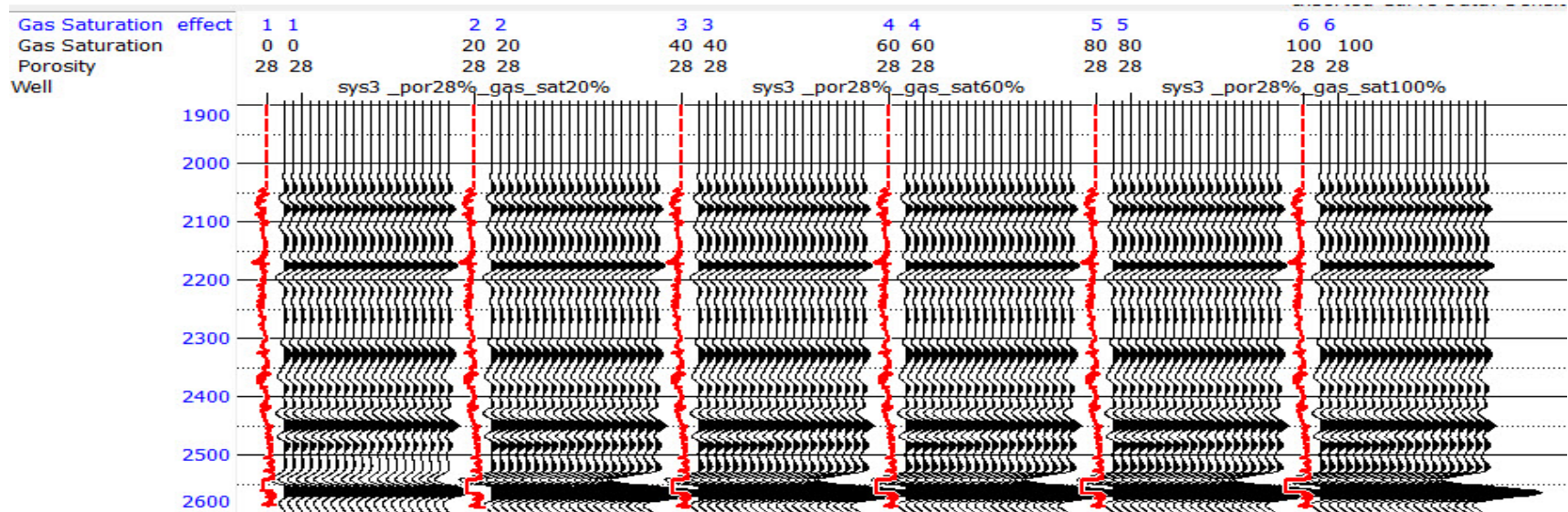
Gas saturation: 0, 20, 40, 60, 80, 100 %





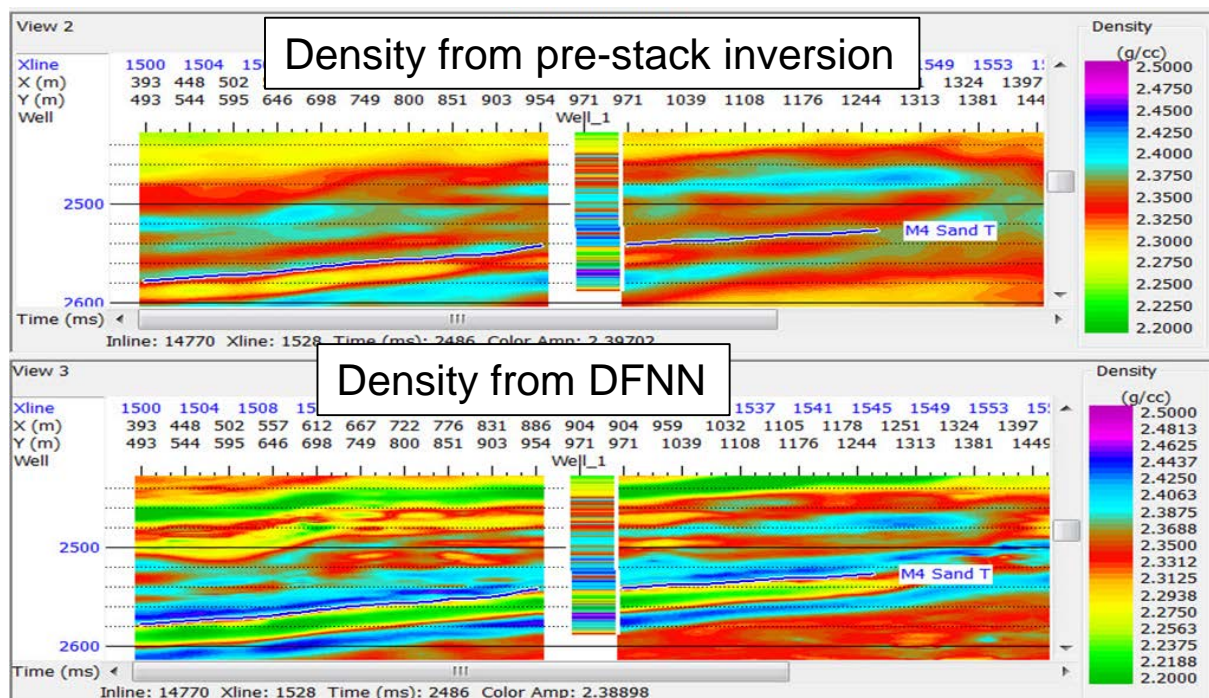
# Augmenting real data with synthetic

Synthetics with density inserted:



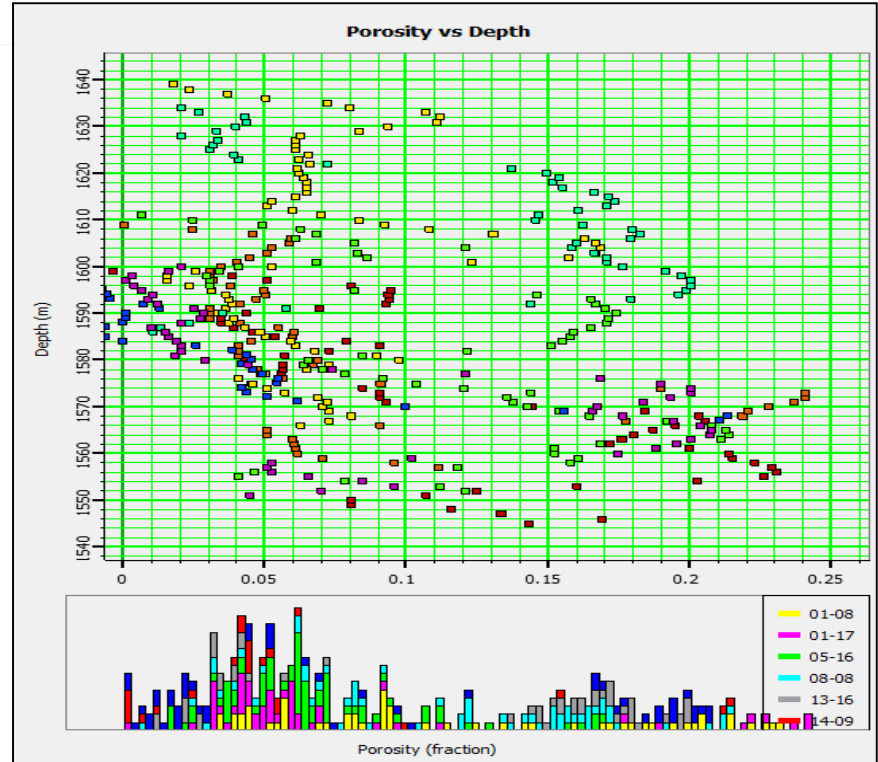
# Density prediction GOM data set

The density predicted by DFNN gives a higher resolution result than pre-stack inversion and appears to tie the well better.

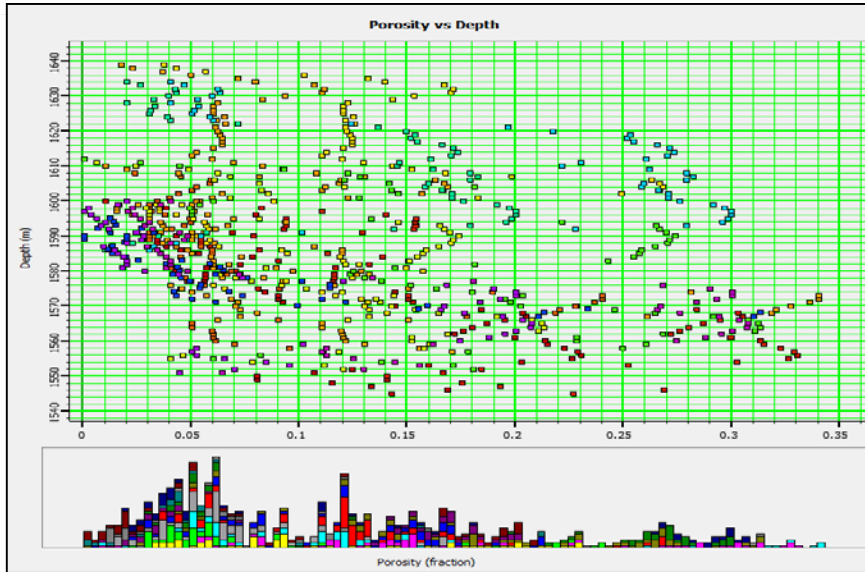


## Workflow 2: create wells using Rock Physics Modelling

- Fit one or more rock physics models (RPMs) to the well data
- Create additional logs to use in the application of the RPMs – here we compute Vshale volumetric logs
- Calibrate the RPMs to the real well data
- Create enhanced porosity logs
- Use the enhanced porosity logs as input to the calibrated RPMs to compute predicted elastic logs
- The input training data for the channel interval has porosities up to around 22%, with a mean value of ~5%



# Enhanced EMERGE training set

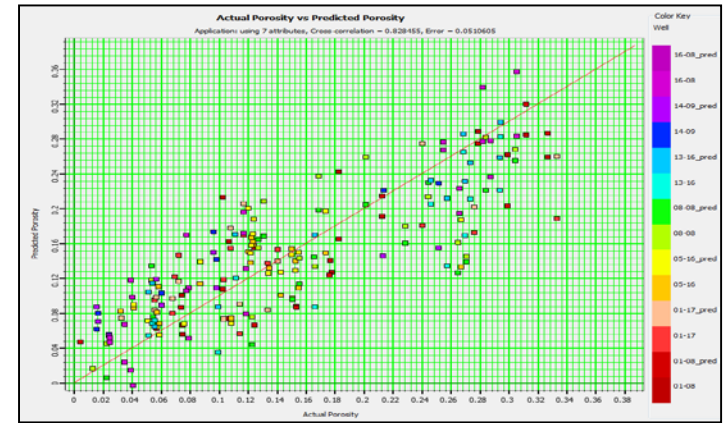
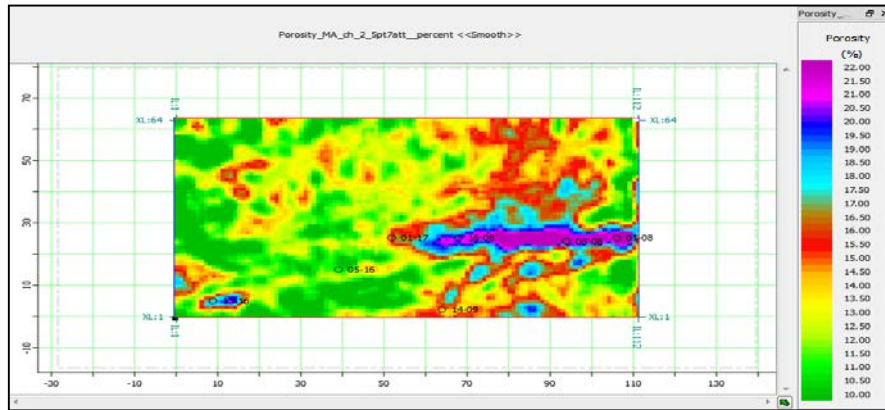


- Looking at the histogram of porosity samples for the channel interval, the new training data now has porosities up to 34%, with a mean value of ~11%



# Prediction using extra data

- After completing a standard EMERGE project to predict a volume of porosity using all the new data, the best prediction gives a clear definition of the high porosity channel feature:



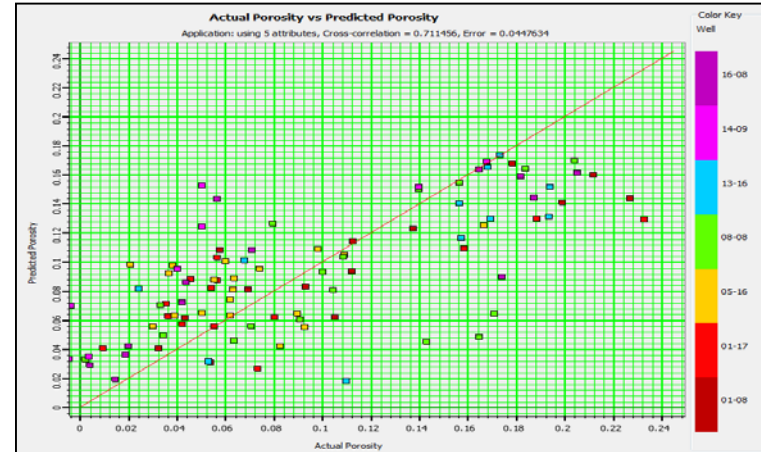
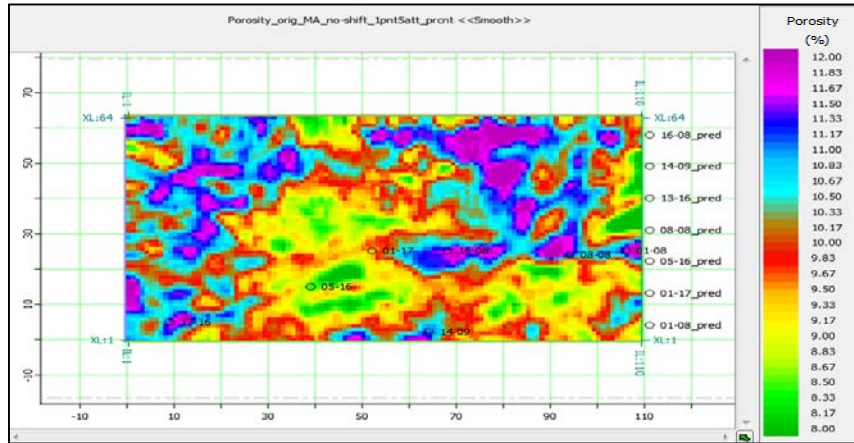
	Correlation	*Error (%)	**Total error (%)
Application:	0.828	5.11	29.4
Validation:	0.743	6.15	35.4

\*This is RMS error (the difference between actual & predicted logs)



# Prediction using only original data

- Here is the equivalent slice for the best porosity prediction in EMERGE using only the original seven wells
- Notice the narrower dynamic range in predicted porosity, lower correlation values and higher errors



	Correlation	Error (%)	Total error (%)
Application:	0.711	4.47	40.5
Validation:	0.648	4.88	44





## Summary

- Introduced the machine learning functionality via Deep Feed-forward Neural Network
- Demonstrated the validation procedure of neural network training
- Discussed DFNN to parameter control and to the amounts of training data.
- Introduced two approaches to expanding the training data model





## Acknowledgements

- Dan Hampson and Jon Downton of CGG for their work on DFNN and contributions to this talk
- Øyvind Kjøsnes, AkerBP for the joined work with CGG HampsonRussell on DFNN that is submitted to EAGE's first workshop on machine learning







# Thank you

[cgg.com](http://cgg.com)

