Deep neural networks to predict reservoir properties from seismic

Jon Downton

CGG GeoSoftware

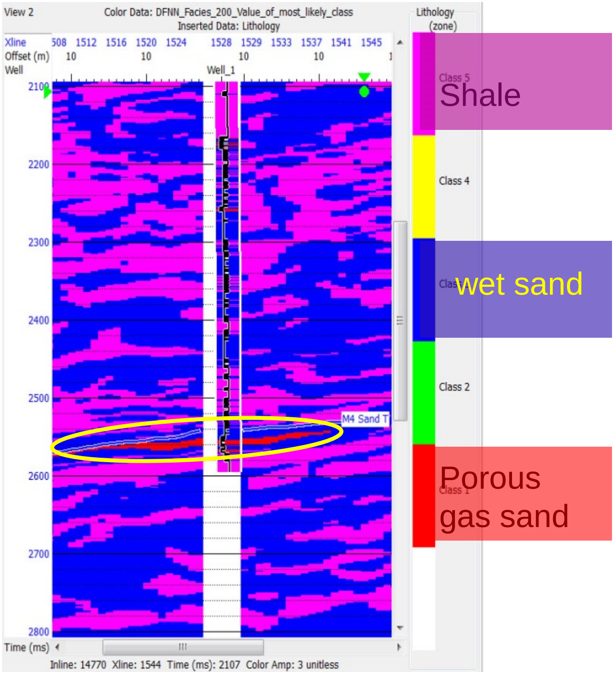
Theme

- Neural networks have been used for some time in geophysics to quantitatively predict rock properties from seismic data.
- In the last decade there has been tremendous progress in the field of machine learning thanks to a powerful new technique called deep learning.
  - Applications of this include hand writing recognition, image recognition, translation, and self driving cars.
- These examples make use of “Big” labelled datasets in order to train the neural networks.
- In the geosciences we are much more restricted in amount of labelled data that we have access to.
- This presentation explores different strategies to overcome this limitation and predict reservoir properties using Deep Neural Networks.

Outline

- Introduction
  – Deep Neural Networks
- The problem of “Small” data
- What can be done with “Small” data?
  – North Sea example
- Theory-guided data science
- Theory-guided design
  – Using a CNN to estimate P-wave Impedance
- Hybrid theory and data analysis
  – Using synthetic data to train a neural network
- Summary

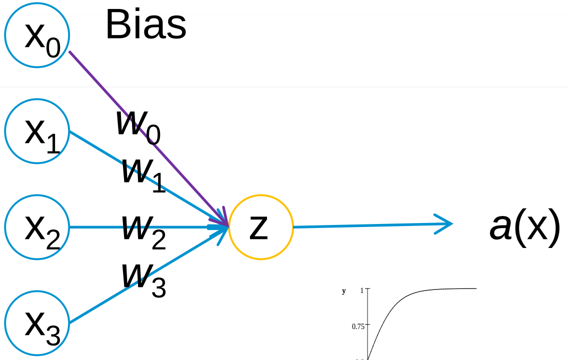
Lithofacies Prediction

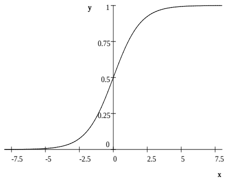


Neuron Representation

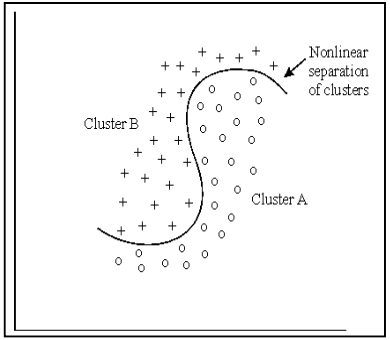
- Neural Networks are based on an idealized version of how we believe the brain works (McCullock and Pitts,1943).
- The basic unit within a Neural Network is the neuron.
- Neural networks start with a linear model
  - The input are the attributes **x**=[1,x_0, x_1, x_2, …,x_N]
  - The input attributes are summed in a similar fashion as in the multilinear model.

$$z = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots$$





Attribute 1



Attribute 2

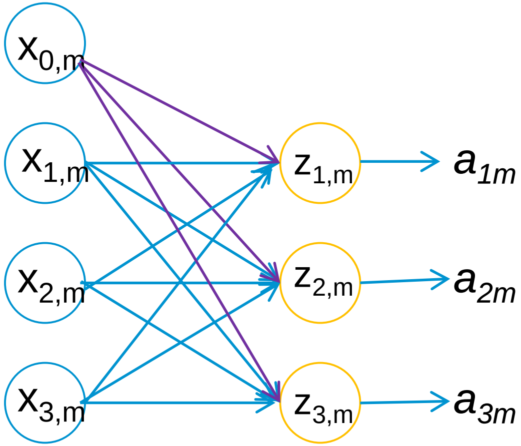
- The output of this is fed into a nonlinear logistic function (or similar).

$$a(x) = \frac{1}{1 + \exp(-z)}$$

As the function *a(x)* is between 0 and 1, the neuron is making a decision whether we are in category 0 or category 1.

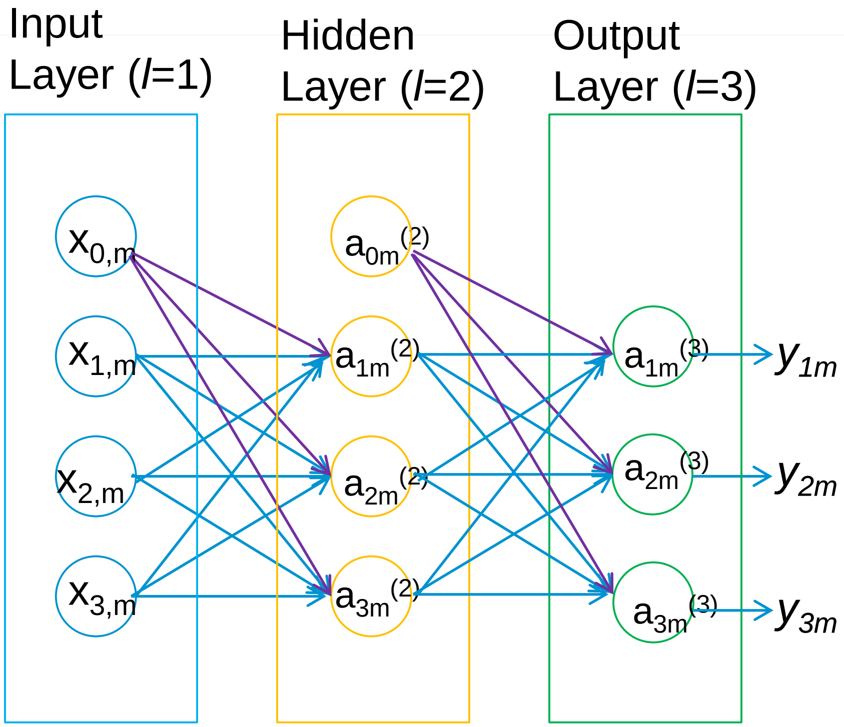
Multiclass Classification (3 classes)

- Neurons can be combined in parallel to perform multiclass classification for three categories.
- The output with the highest value is the most probable and classified as this category



Multilayer Feedforward Neural Network

- By combining two multiclass networks in series we can model nonlinear functions.
- 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.
- The input feeds forward from the input layer to the output layer thus this network is called a Multilayer Feedforward Neural Network (MLFN).

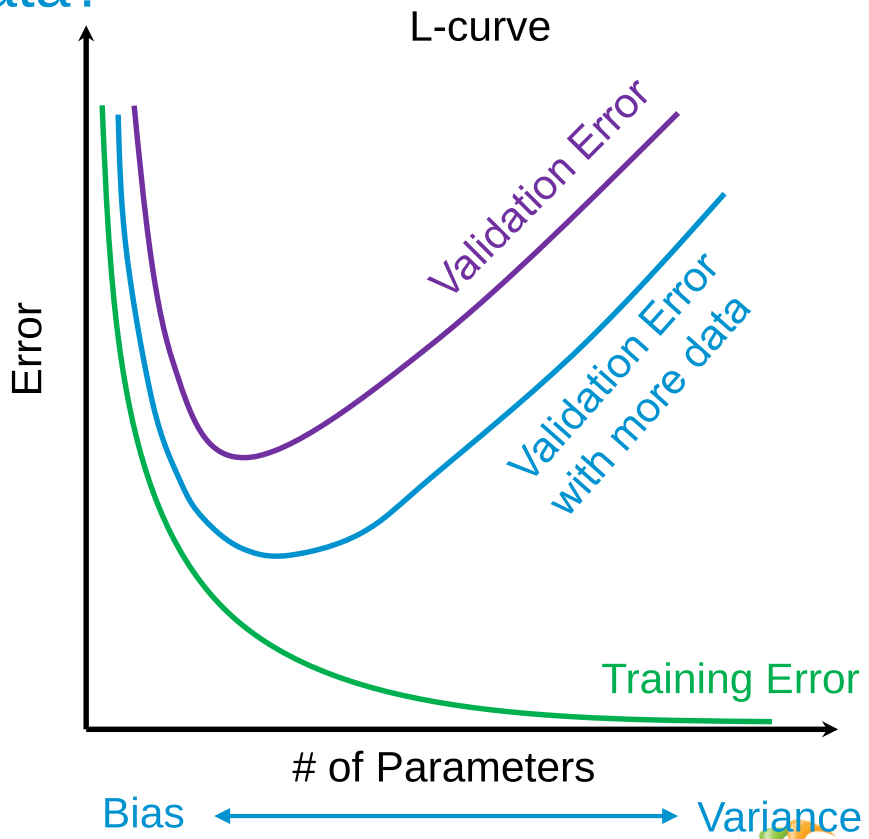


Deep Feedforward Neural Networks (DFNNs)

- In order to model the nonlinear interactions between different features the network must contain at least one hidden layer.
- Additional hidden layers provide extra complexity.
  - Extra layers allow the network to parsimoniously model nonlinear transforms and imposes a hierarchical structure.
  - This allows the network to find and extract the features as part of the training.
- If a network has two or more hidden layers it is considered deep.
- The weights are solved as large nonlinear inverse problem using iterative techniques.
  - For a Deep Feedforward Neural Network the weights are solved using backpropagation.
- Like other supervised methods
  - the weights are calculated on a training dataset.
  - To ensure the network is not over trained the network is tested on a separate validation dataset.

Do we have enough training data?

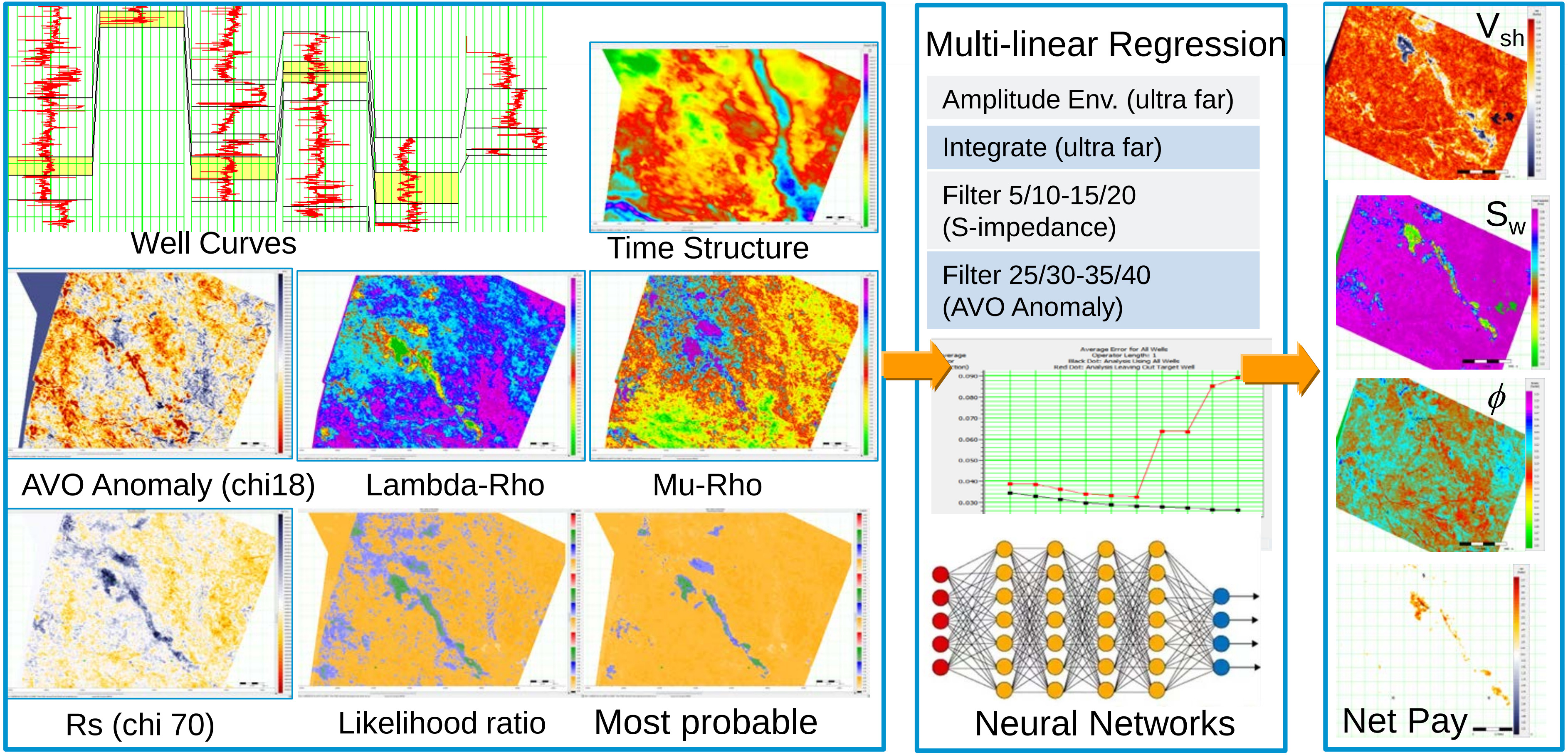
- Deep neural networks have many layers and parameters increasing the risk of overfitting
  - Overfitting is characterized by observing
    - Small training error
    - Large validation error
- Possible solutions
  - Reduce the number of parameters / layers
  - Regularization, early stopping
  - Greedy layer-wise pre-training
  - Increase the amount data
    - Needs to be labelled data!
    - Synthetic data
  - Theory-guided data science



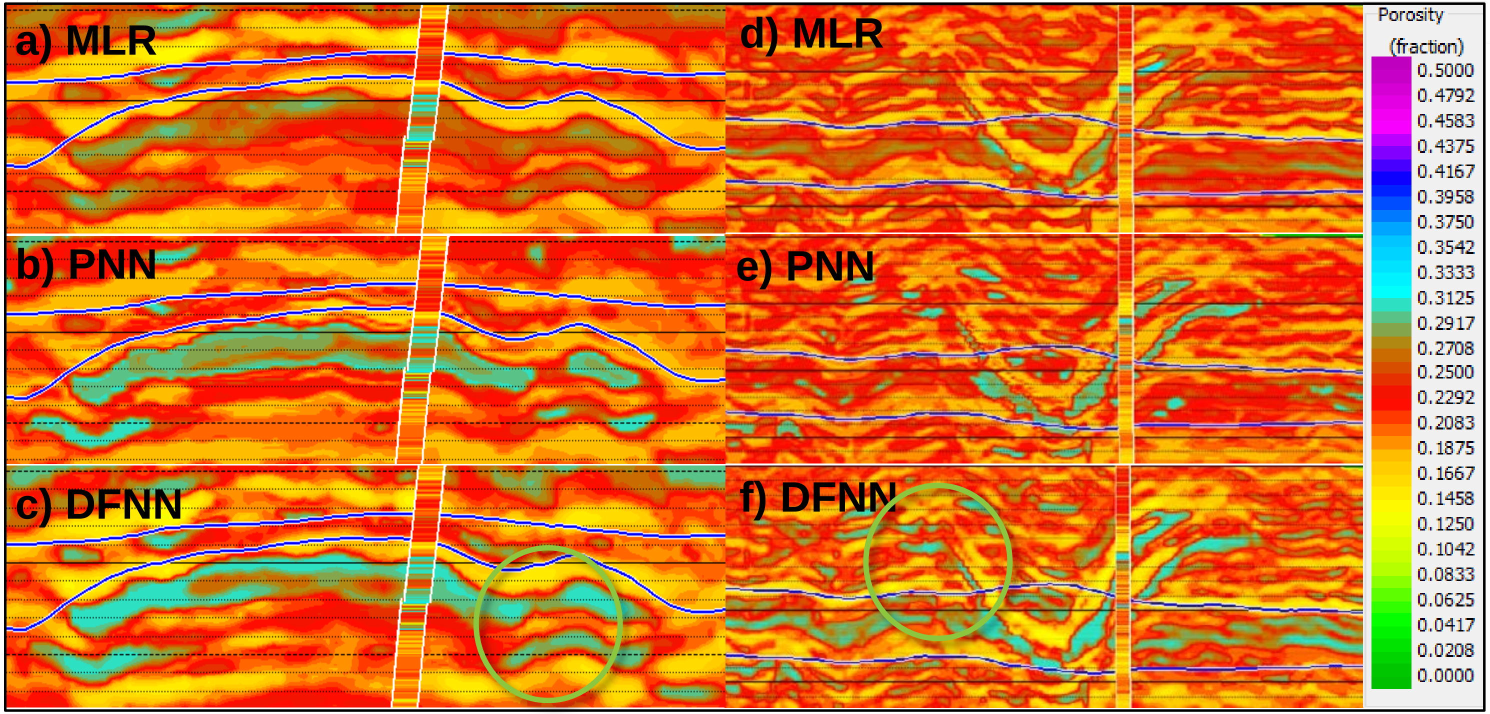
What can be done with small data?

- To overcome the issue of limited training data the first example limits the depth of the network to three hidden layers and uses early stopping.
- The example is from the North Sea and covers two fields both producing commercial volumes of oil from reservoir intervals from within the Paleocene.
  – Field A is a deep marine channelized submarine fan system.
  – Field B is in a remobilized injectite sand, cross cutting a range of stratigraphy at very steep angles.
- The goal of the study was to predict the porosity, volume of shale, water saturation and volume of net pay.
- Six wells were used to train and validate the machine learning.
- Three machine learning techniques were tried and compared including
  – Multi-Linear Regression (MLR),
  – Probabilistic Neural Network (PNN),
  – Deep Feedforward Neural Network (DFNN).

Net-Pay Prediction Workflow



Porosity Prediction



Left: Field A *Right: Field B*

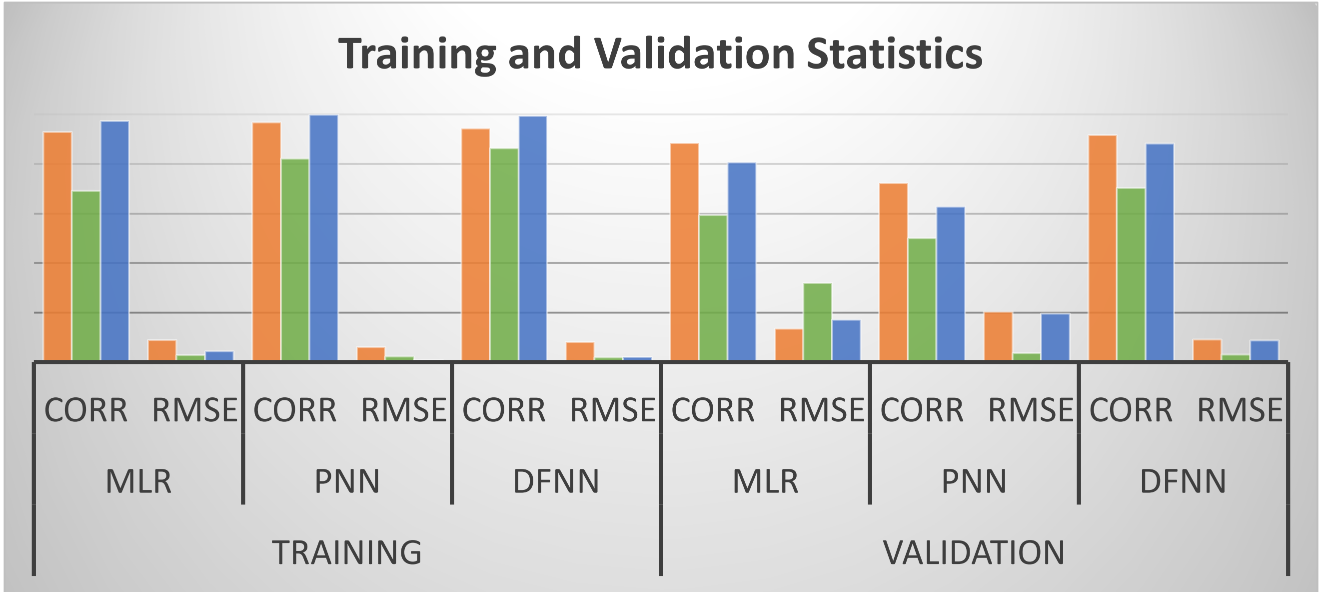
DFNN provides:

- Better lateral continuity in the thin reservoir of field A
- Good estimation of the injectite sand properties in field B

Colwell & Kjøsnes, 2018

Training and Validation Statistics at the well locations

	Training						Validation					
	MLR		PNN		DFNN		MLR		PNN		DFNN	
	Corr	Avg. Error	Corr	Avg. Error	Corr	Avg. Error	Corr	Avg. Error	Corr	Avg. Error	Corr	Avg. Error
VSH	0.929	0.089	0.968	0.0599	0.944	0.081	0.884	0.135	0.723	0.204	0.916	0.091
PHIT	0.692	0.028	0.822	0.023	0.864	0.019	0.593	0.32	0.5	0.036	0.703	0.03
SW	0.974	0.043	0.999	0.009	0.994	0.021	0.806	0.171	0.628	0.196	0.883	0.087



Colwell & Kjøsnes, 2018

- MLR predicts correct variations at the well locations but not correct magnitude
- PNN drops significantly from training to validation
- DFNN shows the highest correlation value and lowest validation RMS error (RMSE)
- DFNN gives consistent statistics from training to validation

Theory-based vs. Data Science Models (Karpatne et al. 2017)

Contain knowledge gaps in describing certain processes

(ϕ and S_w from velocity and density)

Zero-offset reflectivity

$$R_i = \frac{Z_{i+1} - Z_i}{Z_{i+1} + Z_i}$$

Biot-Gassmnn

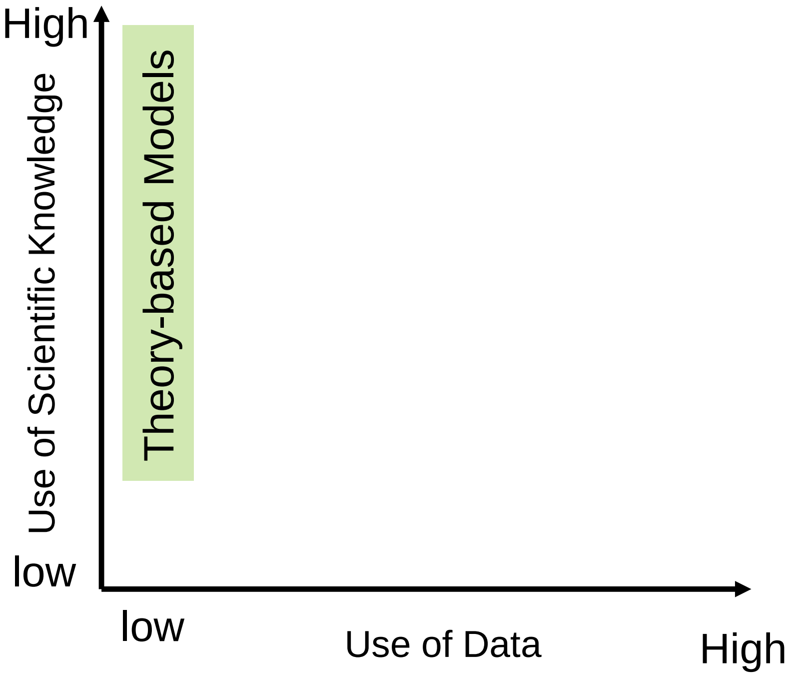
$$K_{sat} = K_{dry} + \frac{\left(1 - \frac{K_{dry}}{K_m}\right)^2}{\frac{\phi}{K_{fl}} + \frac{1-\phi}{K_m} - \frac{K_{dry}}{K_m^2}}$$

Convolutional Model

$$S = W * R + Noise$$

Density-Porosity

$$\rho_{sat} = \rho_m (1-\phi) + \rho_w S_w \phi + \rho_{hc} (1 - S_w) \phi$$

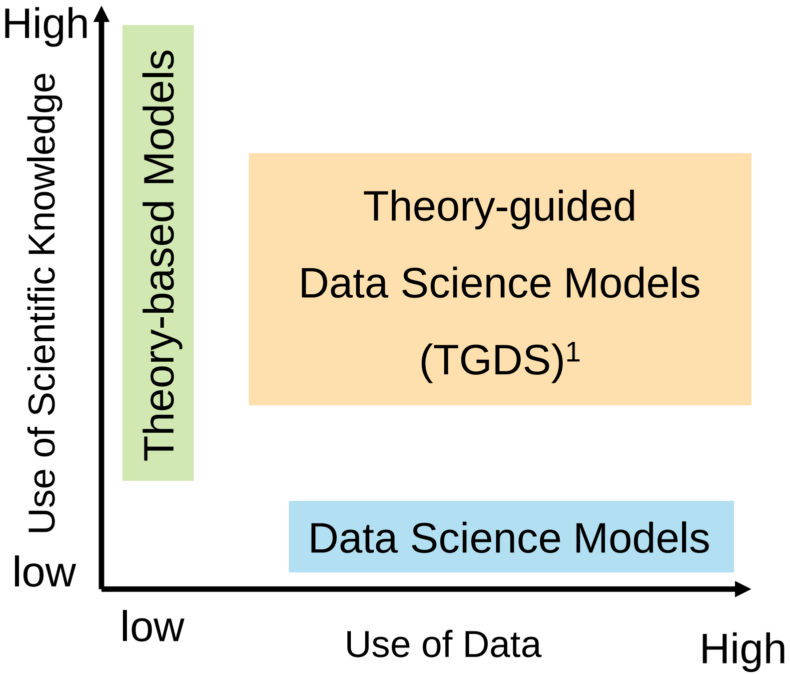


Karpatne et al. "**Theory-guided data science: A New paradigm for scientific discovery**," TKDE 2017

Theory-based vs. Data Science Models (Karpatne et al. 2017)

Contain knowledge gaps in describing certain processes (ϕ and S_w from velocity and density)

Take full advantage of data science methods without ignoring the treasure of accumulated knowledge in scientific "theories"



Require large number of representative samples

[1]Karpatne et al. "**Theory-guided data science: A New paradigm for scientific discovery**," TKDE 2017

Theory-guided Data Science

1) Theory-guided Learning

- Choice of Loss Function
- Constrained Optimization methods
- Probabilistic Models

[Limnology, Chemistry, Biomedicine, Climate, Genomics]

2) Theory-guided Design

- Choice of Response/Loss Function
- Design of Model Architecture

[Turbulence Modeling, Neuroscience]

[1]Karpatne et al. "**Theory-guided data science: A New paradigm for scientific discovery**," TKDE 2017

3) Theory-guided Refinement

- Post-processing
- Pruning

[Remote Sensing, Material Science]

4) Hybrid Models of Theory and Data Science

- Residual Modeling
- Predicting Intermediate Quantities

[Hydrology, Turbulence Modeling]

5) Augmenting Theory-based Models using Data

- Calibrating Model parameters
- Data Assimilation

[Hydrology, Climate Science, Fluid Dynamics]

Theory-guided network design: Impedance Inversion

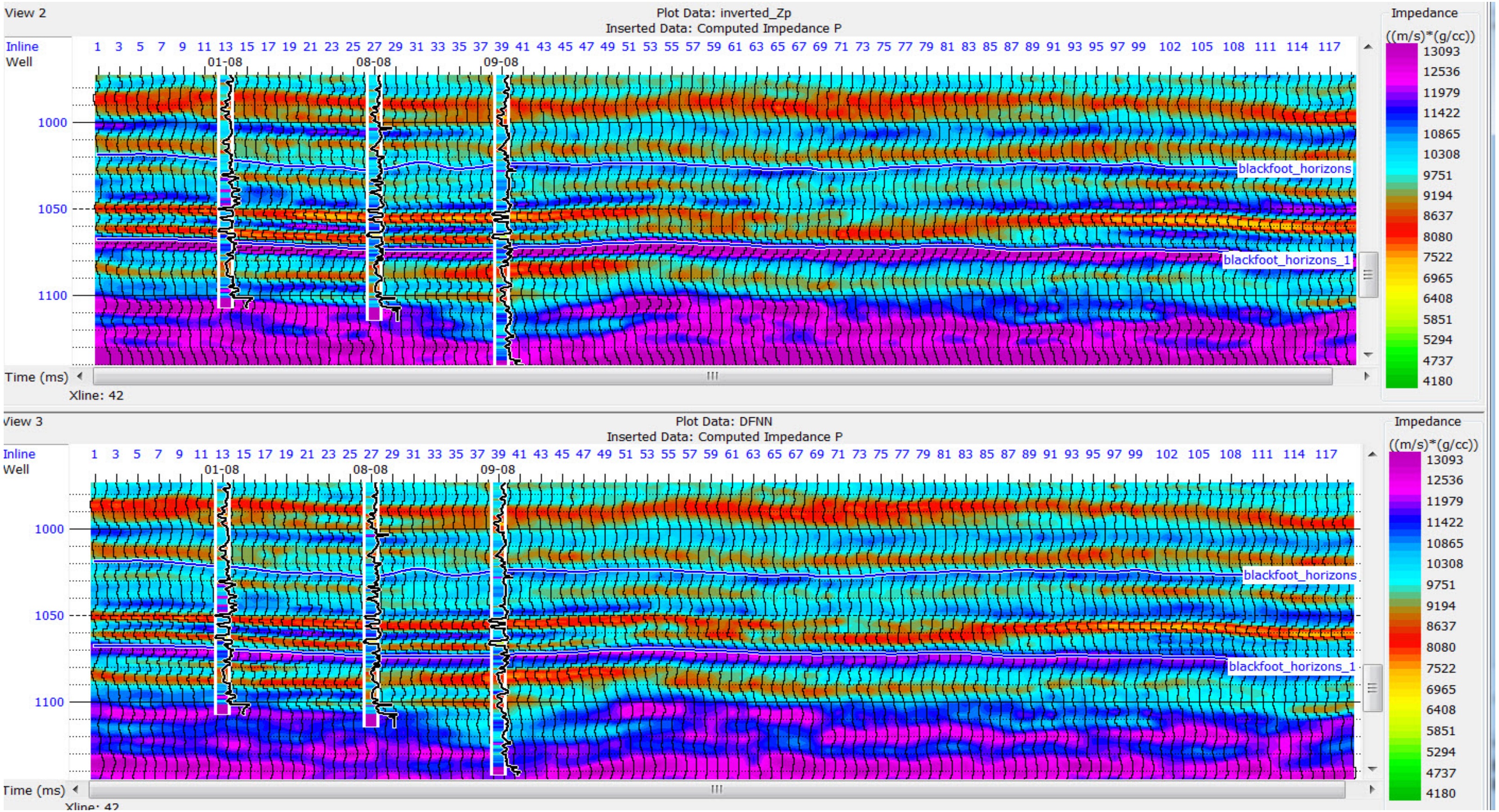
Goal

- Use theory-guided design to build a convolutional neural network (CNN) to simulate poststack impedance inversion
  - poststack impedance inversion is based on the convolutional model
  - The idea is to build the physics into the DFNN architecture by using a convolutional operator.

We tested this concept on the Blackfoot data set.

1. We correlated the wells to the seismic and extracted a wavelet.
2. Using the well control and seismic horizons we built a 3D P-wave impedance model.
   - From this we created a low frequency version of this to serve as a background model.
   - Performed Impedance inversion to serve as a reference.
3. 3D AVO synthetics were generated based on the 3D P-wave impedance model.
4. Train the DFNN at the well locations using a 9 point convolutional operator
   - The input attributes are based on the near offset synthetic data and the low frequency impedance model
   - The 3D Impedance model serves as the target
5. The resulting CNN operator was applied to the seismic to estimate the P-wave impedance and compared to Impedance Inversion.

Blackfoot comparison between conventional inversion (top) and DFNN (bottom).

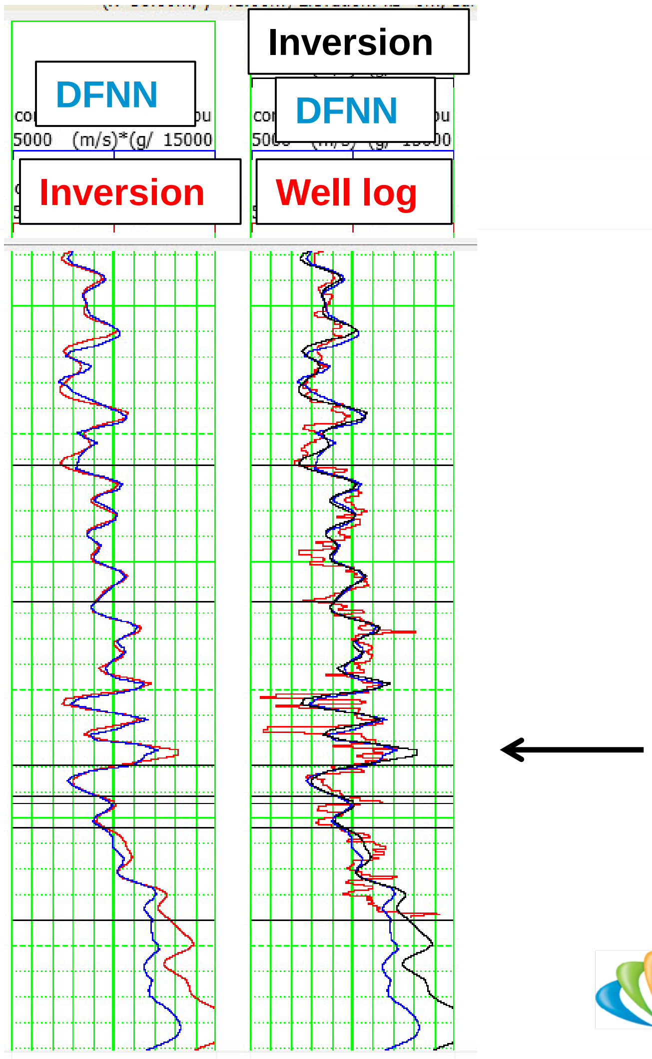


Comparison of the methods at the 09-08 well location

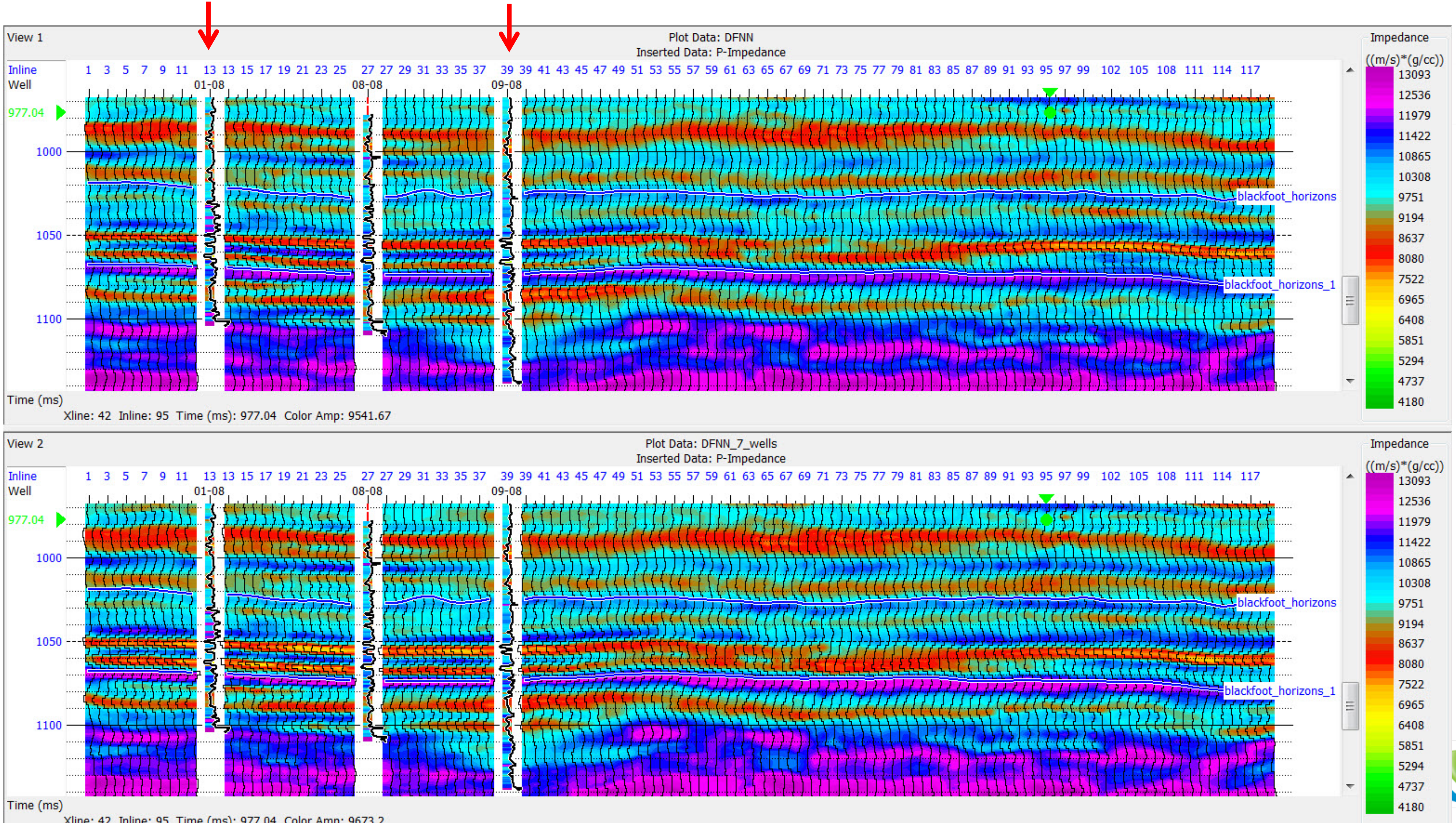
These results look amazingly similar.

This slide shows an overlay of the inversion trace and the DFNN trace at well location 09-08.

Note that the character of the results is very similar, with slight differences in amplitudes. Where there is a noticeable difference, like at the location indicated, the DFNN matches the log curve better.



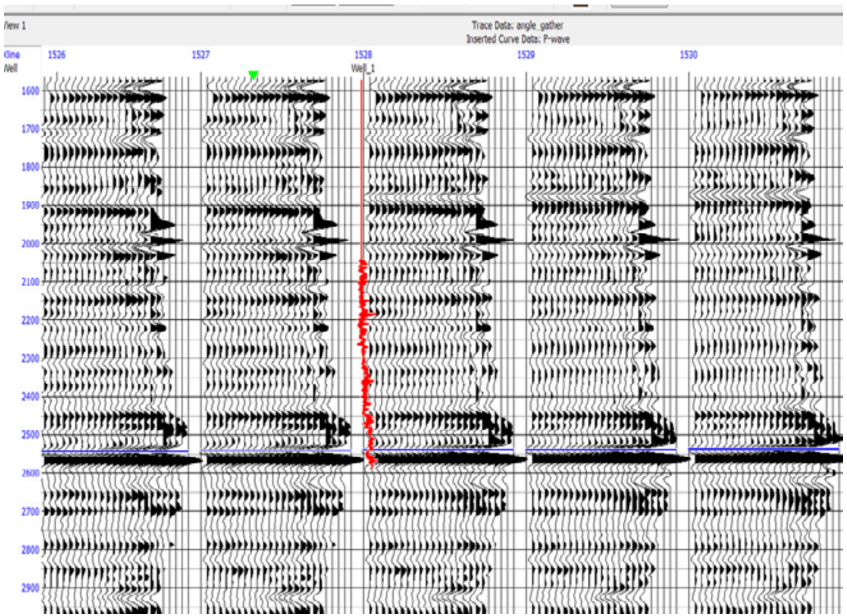
DFNN with all wells (above) and 7 wells (below). The indicated wells are blind.



Gulf Coast example using a hybrid theory and data science model

- In this last example from the Gulf Coast we used a hybrid theory and data model to predict reservoir properties
  - We used a theory-guided model to predict the seismic response due to changes in gas saturation, porosity and fluid
  - Then we use a data science approach (DFNN) to predict the Lithofacies and saturation from the seismic data
- We worked with the fully processed data, including log correlation, wavelet extraction, and transform to angle gathers.
- Only one well is used in this analysis
  - The DFNN is trained and validated on synthetic data
  - Then, the DFNN operator is applied on the real data

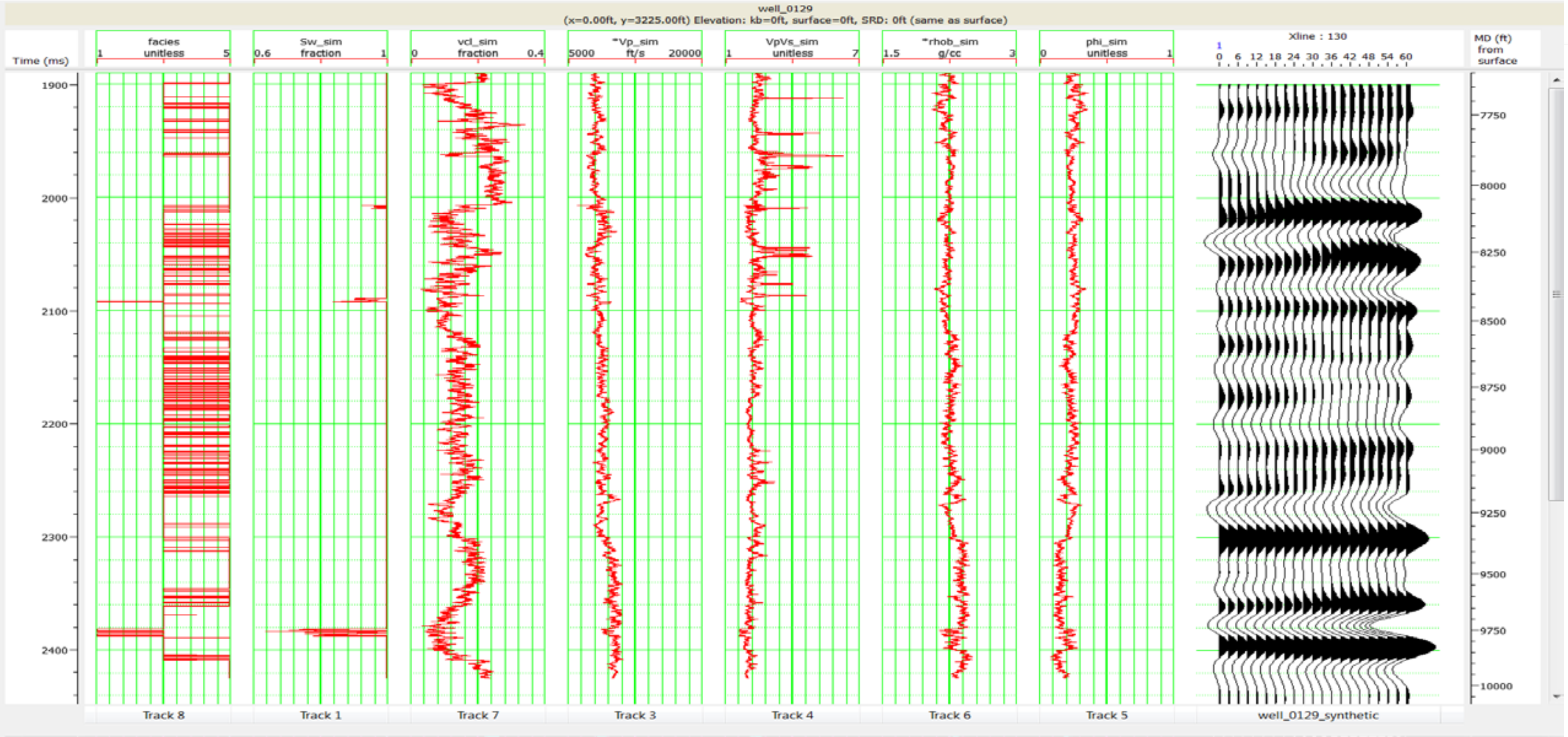
Angle Gathers



Generate a synthetic catalog

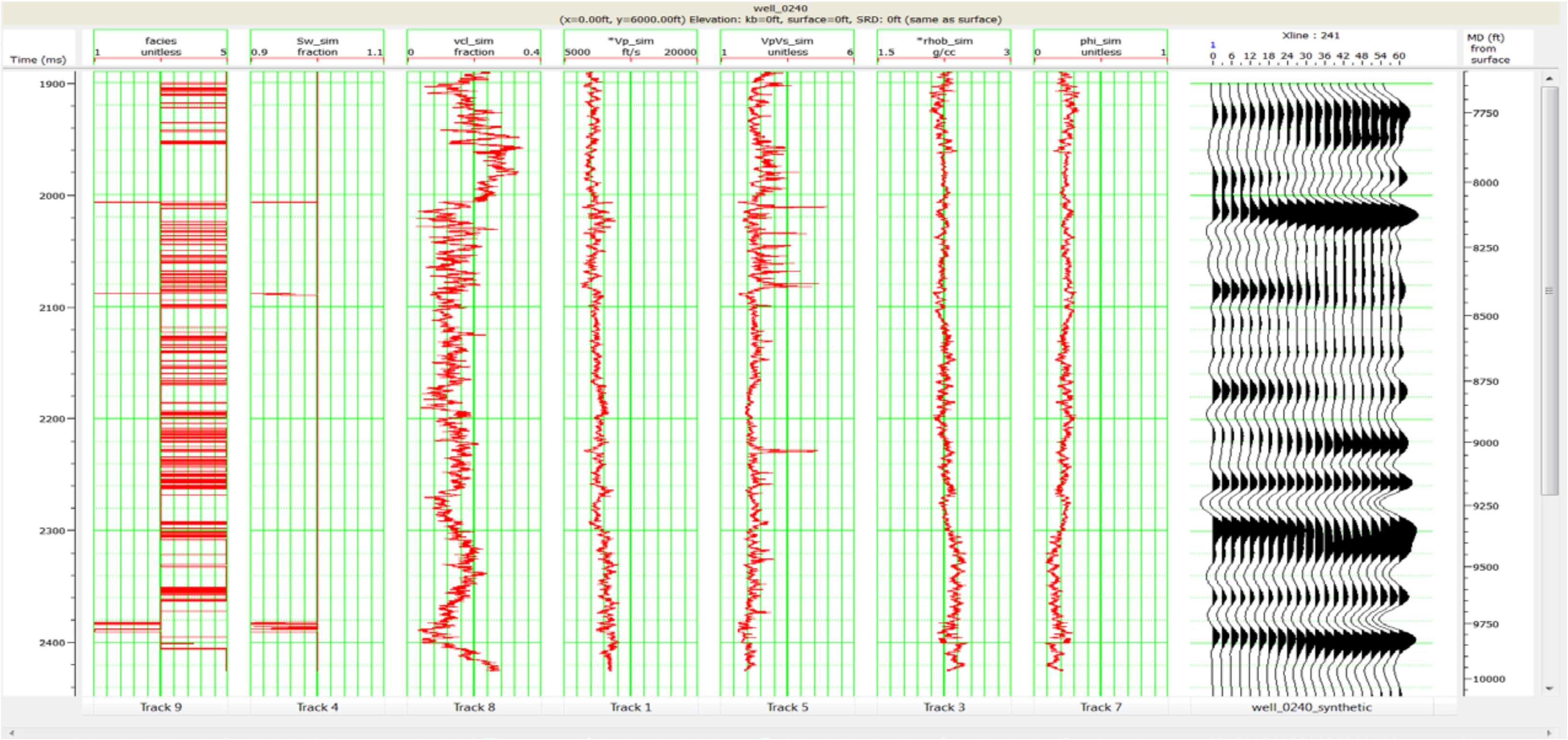
1. For each well perform a petrophysical analysis
2. Establish the rock physics model $Vp, Vs, \rho = \mathrm{RPM}(\phi, V_{cl}, S_w, P_{eff}, T)$
   – To establish the links between the elastic domain with the petrophysical properties
3. Establish the statistics of the key parameters governing the model
   – The background trend and variance of the key parameters
4. Generate elastic models that span the range of the known geology by performing simulations based on the statistics established in step 3
   – Each simulation represents a pseudo-well
5. For each simulation generate synthetic seismic angle gathers
   – The collection of these gathers is called a “Synthetic Seismic Catalog” (Dvorkin et al., 2014)
6. The synthetic seismic gathers are used to train the neural network
   – The actual 3D seismic data has been blind to the creation/training process

Generate synthetic seismic angle gathers for each simulation: simulation 130



November 30, 2018

Generate synthetic seismic angle gathers for each simulation: simulation 241



November 30, 2018

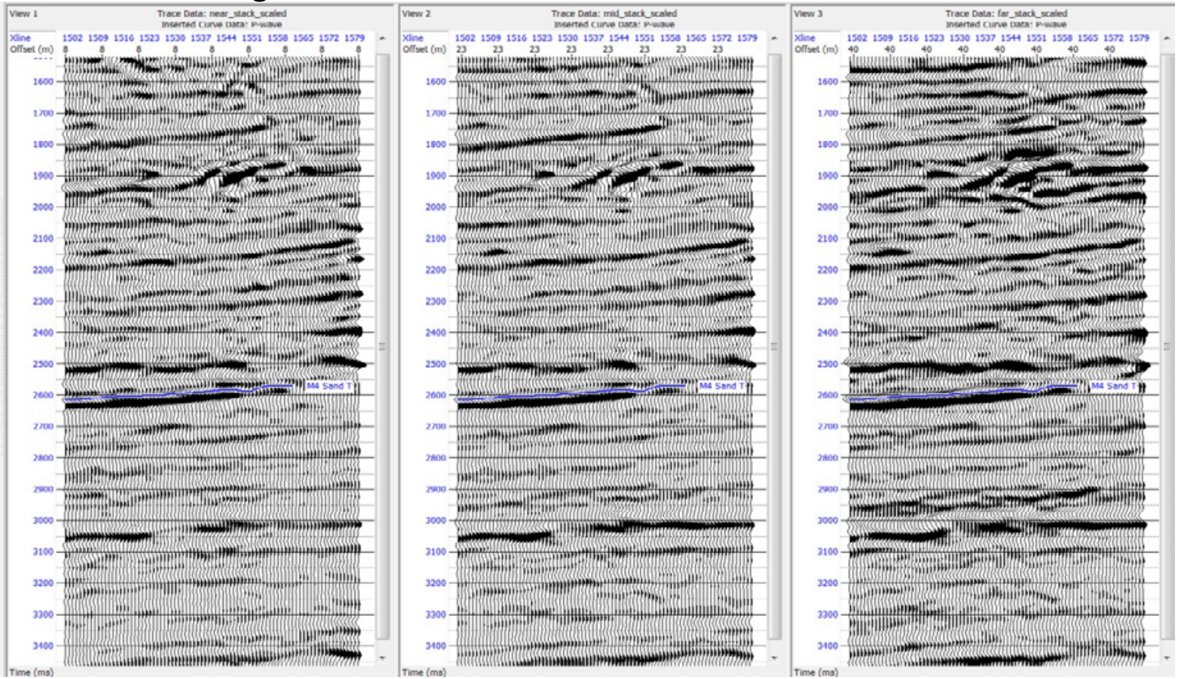
Training the DFNN

- The synthetic seismic gathers are then used to train the neural network
- The synthetic gathers were processed to generate near, mid and far angle stacks consistent with the seismic data.
- Any curve can be used as a target including the
  - Elastic Parameters: P-wave and S-wave impedance, density
  - Rock Properties: Gamma Ray, Porosity, Saturation
  - Facies
- The operator is then applied to the seismic which has been blind to the whole process so far

Scaled Near Angle
0 to 20 degrees

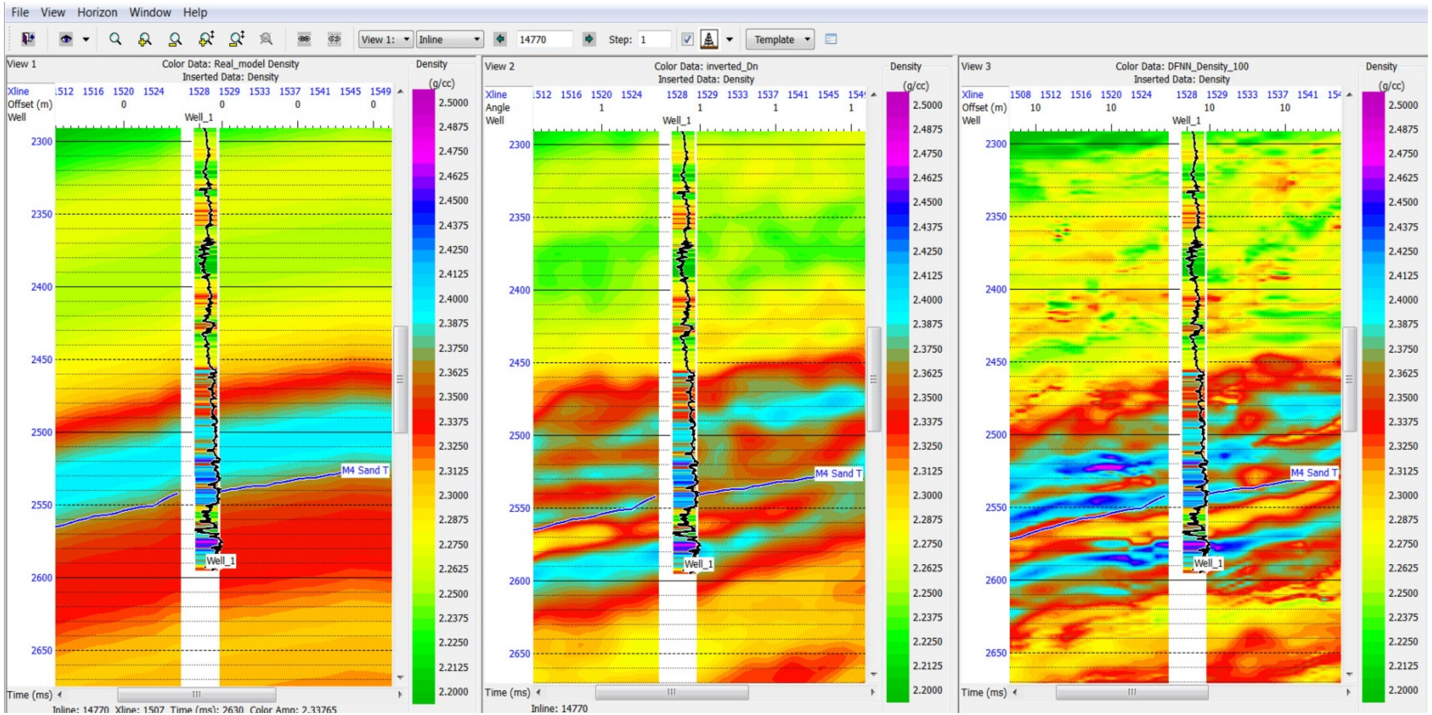
Scaled Mid Angle
20 to 40 degrees

Scaled Far Angle
40 to 50 degrees



Applying the DFNN operator to the real data

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

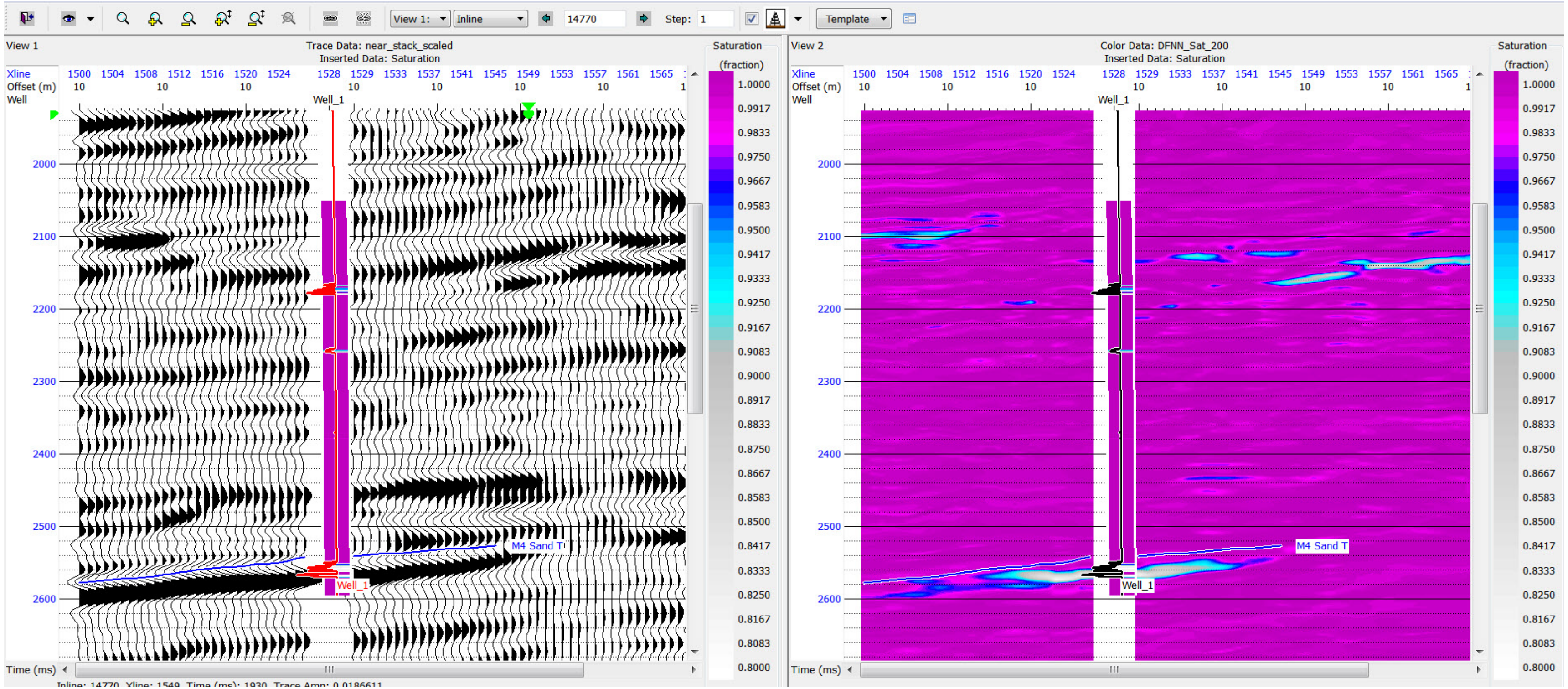


Low-frequency density model

Density from inversion

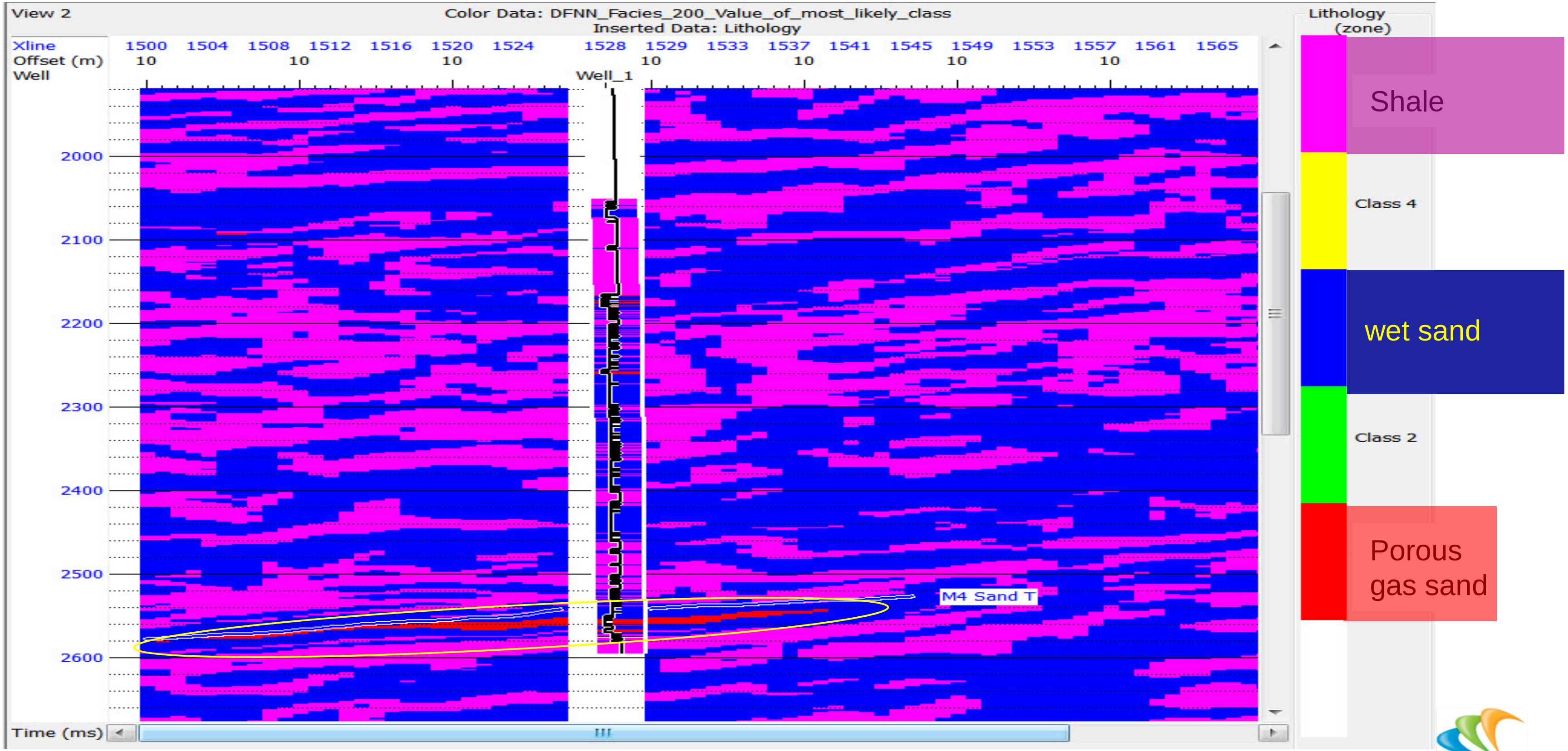
Density from DFNN

Gas Saturation



November 30, 2018

Application of DFNN lithology prediction to real volume.



Summary

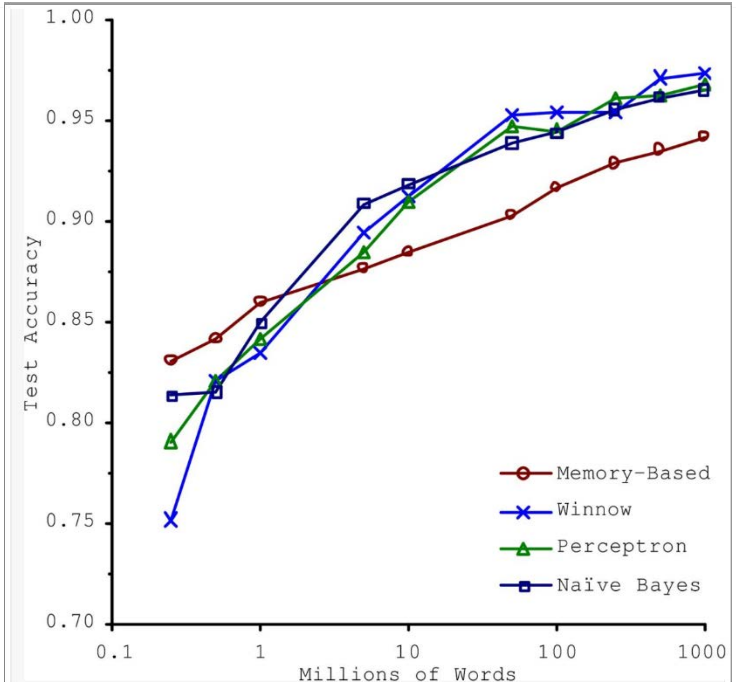
- Deep feedforward neural networks (DFNNs) show great promise as a methodology to quantitatively predict the reservoir.
- The challenge in adopting DFNNs in the geosciences is the relative scarcity of labeled training data.
- The three examples shown in this presentation showed different strategies in dealing with this issue.
  - The North Sea example showed that by limiting the complexity of the network to three hidden layers and using early stopping the DFNN achieved better results than other machine learning techniques.
- The next two examples used theory-guided data science solutions
  - The Blackfoot example uses theory to guide the network architecture. The CNN impedance estimates are close to identical to the poststack impedance inversion results.
  - The Gulf Coast example used rock physics and seismic theory to generate synthetic data to train the neural network. The resulting DFNN was then applied to real seismic data generating geologically plausible estimates of the water saturation and lithofacies.

Acknowledgements

- I would like that the following companies and individuals for their help in generating this presentation
  - CGG GeoSoftware
    - Dan Hampson
    - Brian Russell
    - Tanya Colwell
  - AkerBP
    - Øyvind Kjøsnes

Big Data

- Increasing the amount of training data improves the accuracy of the network
- The paper “The Unreasonable Effectiveness of Data” by Norvig et al. (2009) argues that increasing the amount of data is often more important than the selection of choice of algorithm.



Norvig et al. (2009)