Petrophysical properties prediction from pre-stack seismic data using CNNs

Vishal Das and Tapan Mukerji

Paper F4
Meeting 2019
Summary

- Petrophysical properties inversion using machine-learning
Motivation

- Use Convolutional Neural Networks (CNNs) to perform petrophysical inversion
- Compare and contrast neural network approach with conventional methods
- Understand the machine in machine learning process
Petrophysical inversion using CNNs

- Cascaded approach and end-to-end learning
- Synthetic modeling to generate labeled training data
- Building neural network architecture and hyper-parameter tuning
- Testing the performance of the neural network
Workflow

1. • Synthetic dataset generation
   • Training/ Development datasets

2. • Neural network architecture
   • Hyperparameter tuning

3. • Test and predict using CNN
   • Performance evaluation
Generating a labeled dataset

Petrophysical logs

<table>
<thead>
<tr>
<th>Depth (m)</th>
<th>( \phi ) (v/v)</th>
<th>Vclay (v/v)</th>
<th>Sw (v/v)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Seismic Gather

\( \theta \rightarrow \)
Generating a labeled dataset

- Sequential Gaussian Simulation
- Rock-physics modeling

Facies
- Petrophysical properties
  - Depth (m)
  - Facies 1
  - Facies 2

Sequential Indicator Simulation
Sequential Gaussian Simulation
Generating a labeled dataset

Elastic properties

- Vp (km/s)
- Vs (km/s)
- RHOB (gm/cm³)

Rock-physics modeling

Pre-stack seismic

- Time (s)
- Angle (in degree)

Zoeppritz’s and convolution
Labeled dataset summary

- Total labeled data generated = 2000 well-trace pairs (Petrophysical logs and their corresponding seismic gathers)
  - Training dataset = 1400 traces (70%)
  - Development dataset = 300 traces (15%)
  - Test dataset = 300 traces (15%)
Cascaded network architecture

1. Input: Seismic gather
   - 3x35x10
   - 2x2
   - 3x35x1
   - 2x2
   - 246x3
   - Output: Elastic logs

2. Input: Elastic logs
   - 1x20x1
   - 2x2
   - 1x20x60
   - Output: Petrophysical logs
End-to-end network architecture

Input: Seismic gather

Output: Petrophysical logs

2D CNN  Max-pooling  Fully-connected with Sigmoid activation
Test dataset example: Cascaded network

Average correlation coefficients

Training dataset:
Porosity = 64%
Vclay = 74%
Water saturation = 67%

Test dataset:
Porosity = 55%
Vclay = 66%
Water saturation = 58%
Test dataset example: End-to-end network

Average correlation coefficients

Training dataset:
Porosity = 63%
Vclay = 71%
Water saturation = 65%

Test dataset:
Porosity = 55%
Vclay = 66%
Water saturation = 58%
Comparison of correlation coefficients

Cascaded network

End-to-end network

<table>
<thead>
<tr>
<th>Porosity</th>
<th>Vclay</th>
<th>Sw</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>Test</td>
<td></td>
</tr>
</tbody>
</table>

Correlation coefficient
Comparison of coherence (Test example)

Cascaded network

End-to-end network

Coherence

Porosity

Vclay

Sw

\(\lambda/4\)
Distribution of porosity (Test example)

Cascaded network

End-to-end network
Distribution of Vclay (Test example)

Cascaded network

End-to-end network

Density

Vclay

Density

Vclay

True

Prediction

True

Prediction
Distribution of Sw (Test example)

Cascaded network

- True
- Prediction

End-to-end network

- True
- Prediction
Vertical variogram: Porosity (Test example)

Cascaded network

End-to-end network

Normalized semi-variance

Distance or Lag

True
Prediction

Normalized semi-variance

Distance or Lag

True
Prediction
Vertical variogram: Vclay (Test example)

Cascaded network

End-to-end network

Normalized semi-variance vs. Distance or Lag

True

Prediction
Vertical variogram: Sw (Test example)

Cascaded network

End-to-end network
Computational resources comparison

- Training using Amazon EC2 – p2.xlarge
- 4 CPUs + 1 K80 GPU

### Cascaded network

<table>
<thead>
<tr>
<th>Network</th>
<th>Computation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network 1 in Cascaded network</td>
<td>~15 minutes (1000 epochs)</td>
</tr>
<tr>
<td>Network 2 in Cascaded network</td>
<td>~4 minutes (250 epochs)</td>
</tr>
</tbody>
</table>

### End-to-end network

<table>
<thead>
<tr>
<th>Network</th>
<th>Computation time</th>
</tr>
</thead>
<tbody>
<tr>
<td>End-to-end network</td>
<td>~4 minutes (250 epochs)</td>
</tr>
</tbody>
</table>
## Prediction performance with noisy data

<table>
<thead>
<tr>
<th>Network</th>
<th>Without noise</th>
<th>With noise</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train corr. coeff. ((\phi, V_{\text{clay}}, Sw))</td>
<td>Test corr. coeff. ((\phi, V_{\text{clay}}, Sw))</td>
</tr>
<tr>
<td>Cascaded</td>
<td>64%, 74%, 67%</td>
<td>55%, 66%, 58%</td>
</tr>
<tr>
<td>End-to-end</td>
<td>63%, 71%, 65%</td>
<td>55%, 66%, 58%</td>
</tr>
</tbody>
</table>

### Figures

- **Cascaded network**
  - X-axis: Porosity, Vclay, Sw
  - Y-axis: Correlation coefficient
  - Training and Test phases

- **End-to-end network**
  - X-axis: Porosity, Vclay, Sw
  - Y-axis: Correlation coefficient
  - Training and Test phases
Conclusions

Both CNNs resulted in good correlations between predicted and true petrophysical properties.

CNN based predictions are instantaneous once network is trained.

Cascaded approach requires availability of elastic properties (or physics-based network).

End-to-end approach works better in the presence of noise in input data.
Summary

• Petrophysical properties inversion using machine-learning

Seismic Gather

Neural networks

Petrophysical logs

Depth (m)
Acknowledgements

• Sponsors of Stanford Rock Physics and Borehole Geophysics (SRB) project

https://github.com/vishaldas/petrophysical_inversion (Coming soon!)