Introduction

Physics in machine learning often relies on transformations of data to beneficial domains and simulating additional data. Karpatne et al. (2017) show a physics-guided approach to model lake temperatures with neural networks. Schütt et al. (2017) use deep neural networks to model molecule energies and de Oliveira et al. (2017) employ a special architecture to capture scatter patterns in high-energy physics. When building deep learning pipelines, we can make informed choices in data modeling, but also build neural networks to maximize information gain on the available data. Ulyanov et al. (2018) has shown that the network architecture itself can be used as prior in machine learning. These approaches translate well to geoscience, where strong priors are often necessary to inform decisions.

Deep learning has revolutionized machine learning by replacing the feature generation and augmentation step by learned internal representations of features that maximize information gain. On image data analysis of these neural network filters have shown close relations to edge filters and color separators (Grün et al., 2016). Dramsch and Lüthje (2018) have shown that these filters translate well to seismic data. However, classic feed-forward neural networks do not have the benefit of learning filters. However, these neural networks benefit from recent improvements for regularization (Ioffe and Szegedy, 2015), non-saturating and non-vanishing gradients (He et al., 2015), and training on GPUs.

Neural networks for inversion of seismic data have a long history (Roeth and Tarantola, 1994). In Dramsch et al. (2019) we show the application of a deep multi-layer perceptron for map-based 4D seismic pressure saturation inversion. In this work we show the information gain of feed-forward multi-layer perceptron neural networks by including an explicit calculation of the AVO gradient within the network architecture. It's exemplary for including domain knowledge as a prior in machine learning.

Method

We build a deep feed-forward network to invert seismic amplitude maps for pressure and saturation changes. We use the high-level Python framework `keras` with a `tensorflow` backend. The neural network was trained on synthetic data, to subsequently predict field data. The network takes the seismic input samplewise with near, mid, and far stacks, and pore volume. We inject 20% Gaussian noise to model the noisier field data directly after the input layer. This is fed to a custom layer that calculates the PP AVO gradient between far-mid, mid-near, and far-near. The main components are as follows:

Gaussian noise injection

The synthetic model is noise-free. While we get good results on the training data and the modelled test data, the network does not transfer well to noisy field data. Although the 4D NRMS is very low in the data set, the sample-wise fluctuations in the field seismic differ significantly from the synthetic data. We apply additive Gaussian noise with $\sigma = .02$ to the seismic inputs separately to simulate independent fluctuations of the seismic maps. This significantly decreases the training and validation performance on noise free synthetic data. On field data, however, this enables good transfer of the neural network.

```
noisy_input = GaussianNoise(0.02)(input_data)
```

Explicit AVO gradient calculation

The Schiehallion field is a good example of imbalanced learning. We have many samples of pressure changes ΔP, a good selection of water saturation changes ΔS_w, and very few gas saturation changes ΔS_g. Yet, the changes in gas saturation ΔS_g produce the strongest changes in seismic P wave amplitudes. Statistically, these can easily be regarded as outliers, and therefore, possibly disregarded by the neural network. From decades of seismic analysis, we know that the AVO gradient is very good for pressure saturation separation. We implement an explicit calculation of AVO gradients in the network.

$$G = \frac{A_{\Theta_1} - A_{\Theta_0}}{x_{\Theta_1} - x_{\Theta_0}}, \qquad (1)$$

where G is the PP AVO gradient, A is the seismic P wave amplitude, x is the offset, and Θ is the angle.

```
mid_near = Lambda(
    lambda inputs: (inputs[0] - inputs[1]) / (10)
)([noisy_mid, noisy_near])

far_mid = Lambda(
    lambda inputs: (inputs[0] - inputs[1]) / (10)
)([noisy_far, noisy_mid])

far_near = Lambda(
    lambda inputs: (inputs[0] - inputs[1]) / (20)
)([noisy_far, noisy_near])
```

Encoder-decoder architecture

Subsequently, the four input maps and the three gradient maps are concatenated and fed to an encoder architecture that condenses the information to an embedding layer z. This layer learns a collection of Gaussian distributions to represent the noisy input data The decoder samples this variational embedding layer to calculate the pressure change ΔP, change in water saturation ΔS_w, and gas saturation ΔS_g.

The full architecture is of the encoder-decoder class. The encoder reduces the number of parameters with each subsequent layer. This forces the network to learn a lossy compression of the input data as z-vector. The decoder increases the number of nodes per layer toward the output. The network therefore learns to correlate the low resolution representation with the desired output.

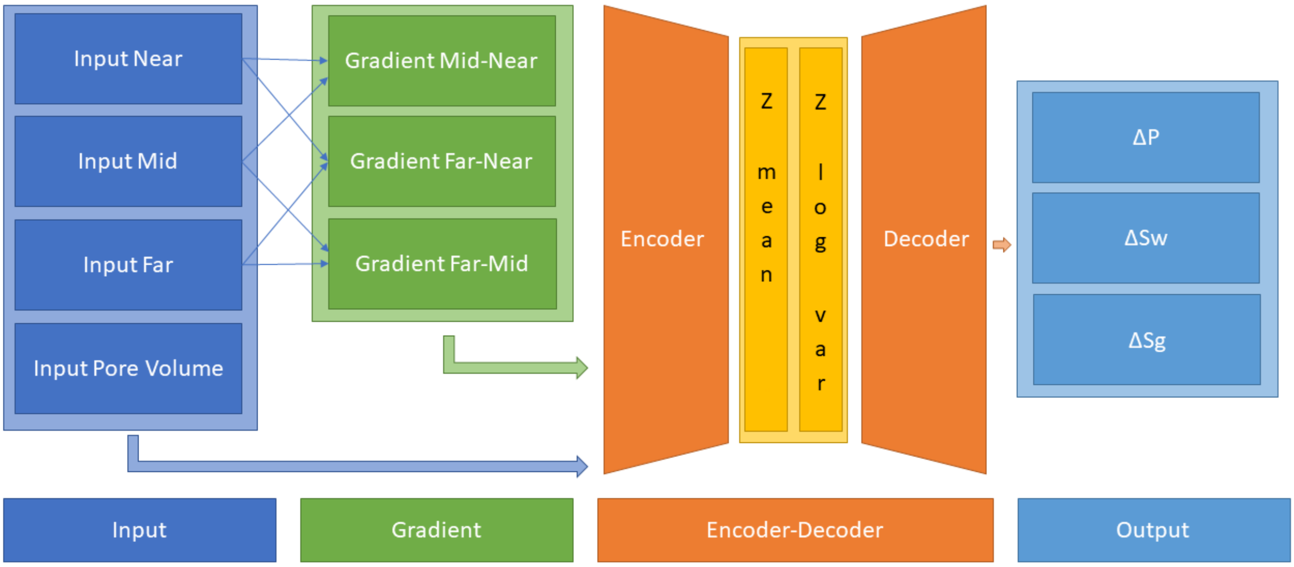

***Figure 1** Full Architecture from Dramsch et al. (2019).*

Variational Z Vector

The inversion of noisy input benefits from a variational representation of compressed z-vector. The networks learns Gaussian distributions in the embedding layer. Therefore, we have to apply the reparametrization trick outlined in Kingma and Welling (2013) to circumvent the sampling process cannot be learned by gradient descent. We use the implementation in Chollet (2015) for variational autoencoders.

Results

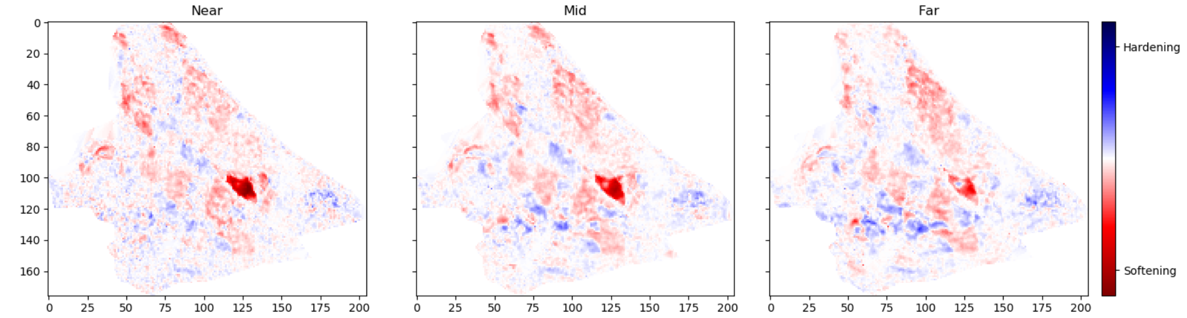

Figure 2 *Schiehallion 2004 Timestep Seismic data, pore volume and sim2seis results.*

In figure 2 we show the 2004 time step of the Schiehallion 4D. Figure 3 contains the inversion result using the variational encoder decoder architecture. Some coherency in the maps can be seen, but each map is very noisy and the gas saturation map contains many data points that indicate gas desaturation, which cannot be confirmed by production data.

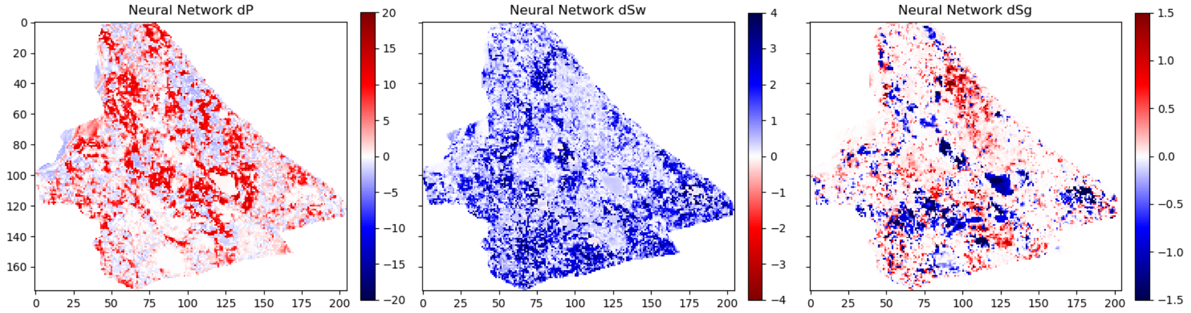

Figure 3 *Variational Encoder Decoder Architecture Inversion*

When we add the gradient, we can clean up some of the misfit in the gas saturation maps ΔS_g. Particularly, the event with the strongest softening in the amplitude maps, is partially reassigned to the pressure map ΔP. However, the inversion process is still very prone to noise. In figure 5, we show the inversion results of a AVO-gradient neural network with a noise injection at training of $\sigma = .02$. The inversion maps are very coherent. Noise injection without gradient calculation does not give adequate results.

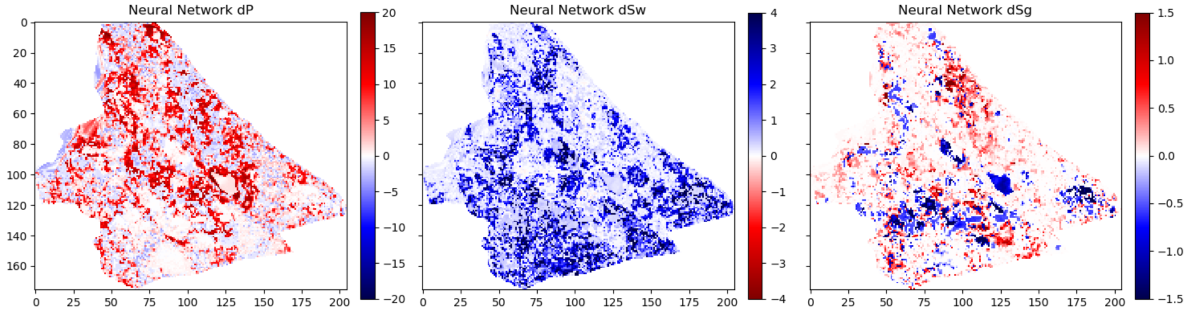

Figure 4 *AVO-Gradient Variational Encoder Decoder Architecture Inversion*

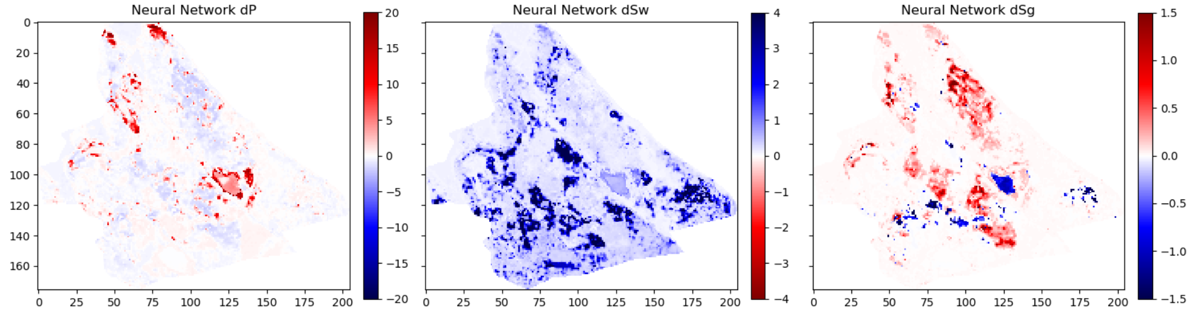

Figure 5 *Noiseinjected AVO-Gradient Variational Encoder Decoder Architecture Inversion*

Conclusions

We have shown a neural network architecture that incorporates physical domain knowledge to enable transfer from synthetic to field data. The final inversion result has very good coherency, despite the network not having any spatial context. While further investigation is necessary, this indicates that useful information has been learned. This is one example, where bias can be intentionally introduced into the network architecture to include physics into machine learning.

Acknowledgements

The research leading to these results has received funding from the Danish Hydrocarbon Research and Technology Centre under the Advanced Water Flooding program. We thank the sponsors of the Edinburgh Time-Lapse Project, Phase VII (AkerBP, BP, CGG, Chevron, ConocoPhillips, ENI, Equinor, ExxonMobil, Halliburton, Nexen, Norsar, OMV, Petrobras, Shell, Taqa, and Woodside) for supporting this research. The Brazilian governmental research-funding agency CNPq. We are also grateful to Linda Hodgson and Ross Walder for important discussions on the field and dataset.

References

Chollet, F. [2015] Keras. https://github.com/fchollet/keras.

Dramsch, J.S., Corte, G., Amini, H., Lüthje, M. and MacBeth, C. [2019] Deep Learning Application for 4D Pressure Saturation Inversion Compared to Bayesian Inversion on North Sea Data.

Dramsch, J.S. and Lüthje, M. [2018] Deep-learning seismic facies on state-of-the-art CNN architectures. In: *SEG Technical Program Expanded Abstracts 2018*, Society of Exploration Geophysicists.

Grün, F., Rupprecht, C., Navab, N. and Tombari, F. [2016] A taxonomy and library for visualizing learned features in convolutional neural networks. *arXiv preprint arXiv:1606.07757*.

He, K., Zhang, X., Ren, S. and Sun, J. [2015] Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the IEEE international conference on computer vision*. 1026–1034.

Ioffe, S. and Szegedy, C. [2015] Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Karpatne, A., Watkins, W., Read, J. and Kumar, V. [2017] Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*.

Kingma, D.P. and Welling, M. [2013] Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

de Oliveira, L., Paganini, M. and Nachman, B. [2017] Learning particle physics by example: location-aware generative adversarial networks for physics synthesis. *Computing and Software for Big Science*.

Roeth, G. and Tarantola, A. [1994] Neural networks and inversion of seismic data. *Journal of Geophysical Research: Solid Earth*, **99**(B4), 6753–6768.

Schütt, K.T., Arbabzadah, F., Chmiela, S., Müller, K.R. and Tkatchenko, A. [2017] Quantum-chemical insights from deep tensor neural networks. *Nature communications*, **8**, 13890.

Ulyanov, D., Vedaldi, A. and Lempitsky, V. [2018] Deep image prior. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 9446–9454.