*DISP '19, Oxford, United Kingdom*
*ISBN: 978-1-912532-09-4*

# Complex-valued neural networks for machine learning on non-stationary physical data

Jesper Sören Dramsch
Technical University of Denmark
Elektrovej 375
DK-2800 Kongens Lyngby
+45 502 503 82
jesper@dramsch.net

Mikael Lüthje
Technical University of Denmark
Elektrovej 375
DK-2800 Kongens Lyngby
+45 93 51 13 92
mikael@dtu.dk

Anders Nymark Christensen
Technical University of Denmark
Richard Petersens Plads 324
DK-2800 Kongens Lyngby
+45 45 25 52 58
anym@dtu.dk

## ABSTRACT

1. Deep learning has become an area of interest in most scientific areas, including physical sciences. Modern networks apply real-valued transformations on the data. Particularly, convolutions in convolutional neural networks discard phase information entirely. Many deterministic signals, such as seismic data or electrical signals, contain significant information in the phase of the signal. We explore complex-valued deep convolutional networks to leverage non-linear feature maps. Seismic data commonly has a lowcut filter applied, to attenuate noise from ocean waves and similar long wavelength contributions. Discarding the phase information leads to low-frequency aliasing analogous to the Nyquist-Shannon theorem for high frequencies. In non-stationary data, the phase content can stabilize training and improve the generalizability of neural networks. While it has been shown that phase content can be restored in deep neural networks, we show how including phase information in feature maps improves both training and inference from deterministic physical data.

## Keywords

Deep Learning, Neural Network, Physics, Geophysics, Seismic

## 2. INTRODUCTION

Seismic data is high-dimensional physical data. During acquisition the data is acquired over an area on the Earth's surface. This images a 3D cube of the subsurface. Due to low reflection coefficients and low signal-to-noise ratio, the measurements are repeated, while moving over the target area. This provides a collection of illumination angles over a subsurface area. The dimensionality of this data has historically been reduced to a stacked 3D cube or 2D sections for interpreters to be able to grasp the information of the seismic data.

With the recent revolution of image classification, segmentation and object detection through deep learning [1], geophysics has regained interest in automatic seismic interpretation (classification), and analysis of seismic signals. Through transfer learning several initial successes were presented [2]. Nevertheless, seismic data has its caveats due to the complicated nature of bandwidth-limited wave-based imaging. Common problems are cycle-skipping of wavelets and nullspaces in inversion problems [3].

Automatic seismic interpretation is complicated, as the modeling of seismic data is computationally expensive and often proprietary. Seismic field data is mostly proprietary and their interpretation is highly subjective and a ground truth is not available. The lack of training data has been delaying the adoption of existing methods and hindering the development of specific geophysical deep learning methods.

Incorporating domain knowledge into general deep learning models has been successful in other fields. We have not found other works, that explicitly incorporates phase information in a deep convolutional neural network.

In this work we calculate the complex-valued seismic trace by applying a Hilbert transform to each trace. We go on to give a brief overview of convolutional neural networks and then introduce the extension to complex neural networks.

We show that including explicit phase information provides superior results to real-valued convolutional neural networks for seismic data. Difficult areas that contain seismic discontinuities due to geologic faulting are resolved better without leakage of seismic horizons. We train and evaluate several complex-valued and real-valued auto-encoders to show and compare these properties. These results can be directly extended to automatic seismic interpretation problems.

## 3. COMPLEX SEISMIC DATA

Complex seismic traces are calculated by applying the Hilbert transform to the real-valued signal. The Hilbert transform applies a convolution with $1/(\pi t)$ to the signal, which is equivalent to a -90 degree phase rotation. It is essential that the signal does not contain a DC component, as this would not have a phase rotation.

The Hilbert transform is defined as

$$H(u)(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{u(\tau)}{t - \tau} \, d\tau,$$

of a real-valued timeseries $u(t)$, where the improper integral has to be interpreted as the Cauchy principal value. In the Fourier domain, the Hilbert transform has a convenient formulation, where frequencies $\omega < 0$ are set zero and the remaining

DOI: http://dx.doi.org/10.17501.....

*ISBN: 978-1-912532-09-4*

frequencies are multiplied by 2. This can be written as

$$x_a = F^{-1}(F(x)2U) = x + iy$$

where $x_a$ is the analytical signal, $x$ is the real signal, $F$ is the Fourier transform, and $U$ is the step function. The imaginary component $y$ is simultaneously the quadrature of the real-valued trace.

This provides locality to explicit phase information, where the Fourier transform itself does not lend itself to resolution of the phase in the time domain.

In conventional seismic trace analysis the complex data is used to calculate the instantaneous amplitude and instantaneous frequency. These are beneficial seismic attributes for interpretation [4].

## 4. CONVOLUTION

Convolution is an operation on two signals *f* and *g* – or a signal and a filter - that produce a third signal, containing information from both of the inputs. An example is the moving average filter, which smoothes the input, acting as a low-pass filter. Convolution is defined as

$$f(t) * g(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

While often applied to real value signals, convolution can be used on complex signals. For the integral to exist both *f* and *g* must decay when approaching infinity.

An illustrative example is given below. We start by the Hilbert transform of a cosine. The real part is the same as the original signal - the cosine.

We now convolve the complex Hilbert signal with the imaginary number *i* (see Figure 1)

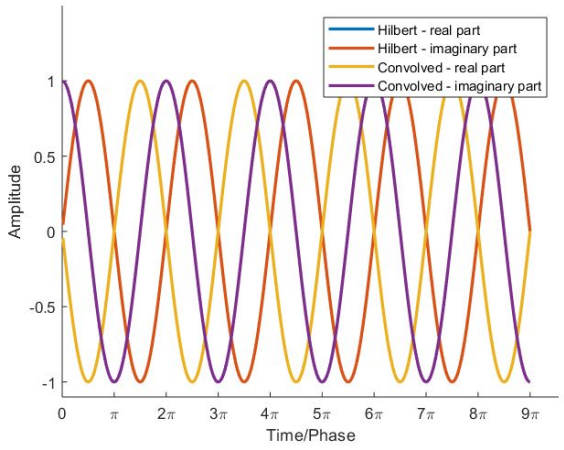


**Figure 1: The Hilbert transformed cosine, convolved with *i***

The real part is unchanged (covered behind the purple line), while the phase has been shifted 2π. We now convolve with a box-shaped signal with a width of π and an amplitude of 1/π (see Figure 2). This corresponds to a moving average.

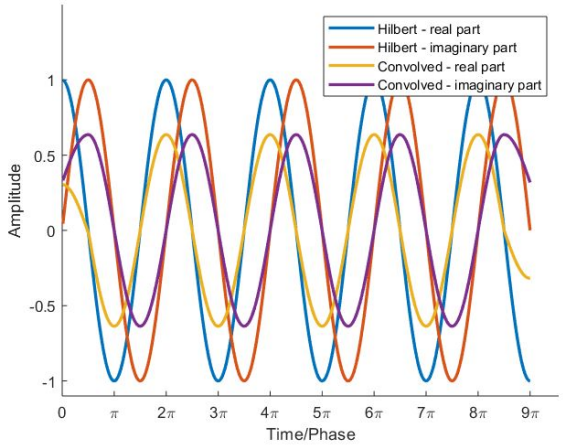


**Figure 2: The Hilbert transformed cosine, moving average**

Note that both the real and imaginary part is affected.

Convolution is directly generalizable to N-dimensions by multiple integrations, and to discrete values by replacing the integrations with summations.

## 5. Convolutional Neural Networks

### 5.1 Basic principles

Convolutional neural networks [5] use multiple layers of convolution and subsampling to extract relevant information from the data. An example is given below (see Figure 3)

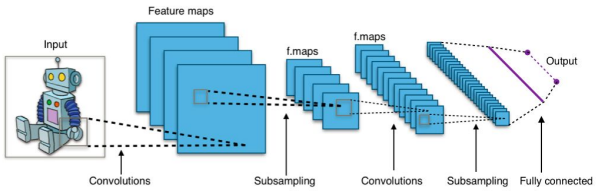


**Figure 3: Illustration of a convolutional neural network. The input image is convolved with different kernel, extracting information, subsampled and further convolved. *By Aphex34 - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=45679374***

The input image is repeatedly convolved with filters and subsampled. This creates many, but smaller and smaller images. For a classification task, the final step is then a weighting of these very small images leading to a decision about what was in the original image. The filters are learned as part of the training process by exposing the network to training images.

The salient point is, that the convolution kernels are learned based on the training. If the goal is - for example - to classify geological fascies, the convolutional kernels will learn to extract information from the input, that helps with that task. It is thus a very strong methodology, that can be adapted to many tasks.

DOI: http://dx.doi.org/10.17501.....

*ISBN: 978-1-912532-09-4*

## 5.2 Complex Convolutional Neural Networks

Complex convolutional networks provide the benefit of explicitly modeling the phase space of physical systems [6]. The complex convolution introduced in Section 4, can be explicitly implemented as convolutions of the imaginary and complex kernels with the real and complex components of the data, these are then combined as follows

$$K = \{M_{\Re} * K_{\Re} - M_{\Im} * K_{\Im}\} +$$

$$i\{M_{\Re} * K_{\Im} + M_{\Im} * K_{\Re}\},$$

where $K$ is the Kernel and $M$ is the data Matrix (see Figure 4).

While max pooling and upsampling do not suffer from complex-valued neural networks, batch normalization [7] does. Real-valued batch normalization normalizes the data to zero mean and a standard deviation of 1. This does not guarantee normalization in complex values. Trabeski et al, [6] implement a 2D whitening operation as follows

$$\widetilde{x} = V^{-\frac{1}{2}}(x - \mathbb{E}[x]),$$

where $x$ is the data and $V$ is the 2x2 covariance matrix, with the covariance matrix being

$$V = \begin{bmatrix} V_{\Re\Re} & V_{\Re\Im} \\ V_{\Im\Re} & V_{\Im\Im} \end{bmatrix}$$

Effectively, this multiplies the inverse of the square root of the covariance matrix with the zero-centred data. Scaling the covariance of the components instead of the variance of the data.

## 5.3 Auto-encoders

Auto-encoders [8] are a special configuration of encoder-decoder network that map data to a low-level representation and back to the original data. These networks map $f(x) = x$, where $x$ is the data and $f$ is an arbitrary network. The architecture of auto-encoders is an example of lossy compression and recovery from the lossy representation. Commonly, recovered data is blurred by this process.

The principle is illustrated in figure 5. The input is transformed to a low-dimensional representation - called a code or latent space - and then reconstructed again from this low dimensional representation. The intuition is, that the network has to extract the most salient parts from the data, to be able to perform a reconstruction. As opposed to other methods for dimensionality reduction - e.g. principal component analysis - an auto-encoder can find a non-linear representation of the data. The low-dimensional representation can den be used for anomaly detection, or classification.

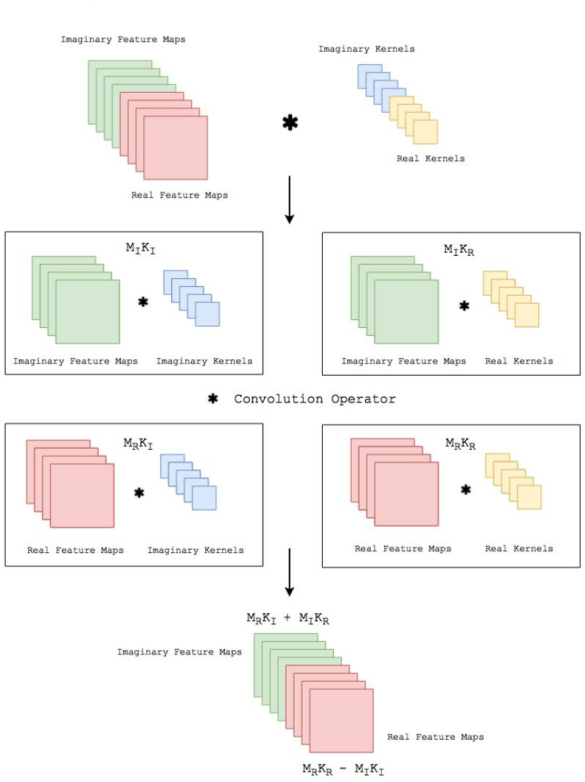


**Figure 4: Implementation details of Complex Convolution CC-BY (Trabelski et al. 2017).**

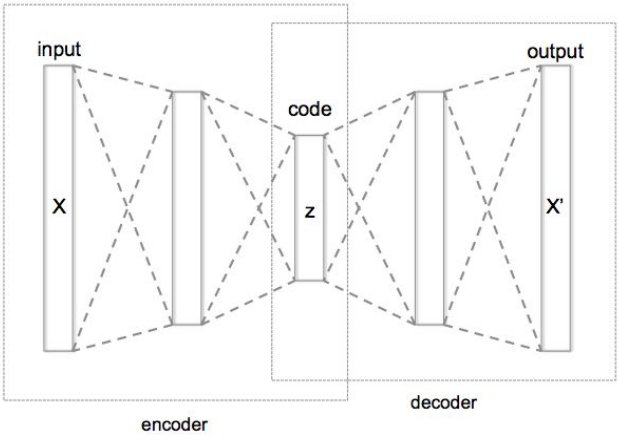


**Figure 5: Typical autoencoder architecture. The data is compressed to a low dimensional code, and then reconstructed. *By Chervinskii - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=45555552***

# 6. Experiments

## 6.1 Data

The data is the F3 seismic data, interpreted by Alaudah et al. [9] (see Figure 6). They provide a seismic benchmark for machine learning with accessible NumPy format. The interpretation

DOI: http://dx.doi.org/10.17501.....

(labels) of the seismic data is relatively coarse compared to conventional seismic interpretation, but the accessibility and pre-defined test case are compelling.

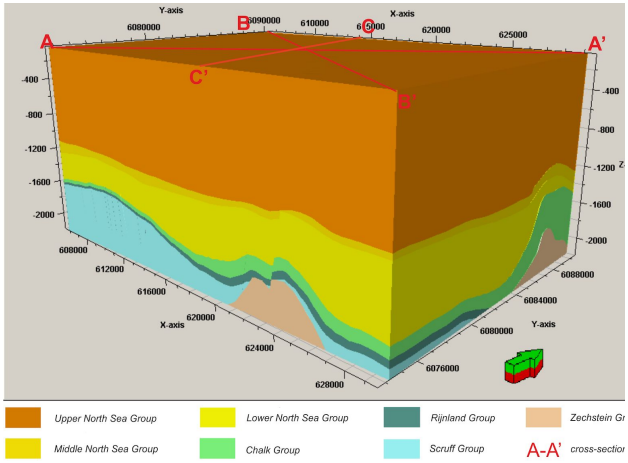


**Figure 6 : Interpretation (labels) for seismic F3 block. CC-BY [8]**

We generate 64x64 patches in the inline and crossline direction to train our network. The fully convolutional architecture can predict on arbitrary sizes after training.

The seismic data is normalized to values in the range of [-1, 1]. To obtain complex-valued seismic data we Hilbert transform every trace of the data.

## 6.2 Architecture

The Auto-encoder architecture uses 2D convolutions with 3x3 kernels. We employ batch normalization to regularize the training and speed up training [9]. The down and up sampling is achieved by MaxPooling and the UpSampling operation. We reduce a 64x64 input 4 times by a factor of two to encode a 4x4 encoding layer. The architecture for the complex convolutional network is identical, except for replacing the real-valued 2D convolutions with complex-valued convolutions. The layers used are shown below (see Table 1).

**Table 1: Layers used in the auto-encoder**

| Layer (type) | Output Shape |
|---|---|
| Input | 64, 64, 1 |
| Conv2D | 64, 64, 8 |
| Conv2D + BN | 64, 64, 8 |
| MaxPooling2D | 32. 32. 8 |
| Conv2D + BN | 32, 32, 16 |
| MaxPooling2D | 16, 16, 16 |
| Conv2D + BN | 16, 16, 32 |
| MaxPooling2D | 8, 8, 32 |
| Conv2D + BN | 8, 8, 64 |
| MaxPooling2D | 4, 4, 64 |
| Conv2D | 4, 4, 128 |
| Upsampling2D | 8, 8, 128 |
| Conv2D + BN | 8, 8, 64 |
| Upsampling2D | 16, 16, 64 |
| Conv2D + BN | 16, 16, 32 |
| Upsampling2D | 32, 32, 32 |
| Conv2D + BN | 32, 32, 16 |
| Upsampling2D | 64, 64, 16 |
| Conv2D | 64, 64, 8 |
| Conv2D + BN | 64, 64, 8 |
| Conv2D | 64, 64, 1 |

Complex-valued neural networks contain two feature maps for every feature map contained in a real-valued network. The connecting edges are not treated equally due to the real and complex components not being independent. Matching real-valued and complex-valued neural networks is quite complicated, as the same filter values yield a vastly different amount of parameters. The real-valued network described in Table 1 has 198,001 parameters. A complex-valued network with equal output shapes has 100,226 parameters due to parameter sharing of complex values. A complex-valued network with the same amount of nodes as the real-valued network in Table 1 would have 397,442 parameters. A real-valued network with an equivalent formulation to the larger complex-valued neural network has 790,945 parameters. We evaluate these four configurations.

## 6.3 Training

We train the networks with an Adam optimizer and a learning rate of $10^{-3}$ for 100 epochs. The loss function is mean squared error, as the seismic data contains values in the range of [-1,1]. All networks reach stable convergence without overfitting.

## 6.4 Evaluation

We compare the complex auto-encoder with the real-valued auto-encoder, through the reconstruction error on unseen test data and qualitative analysis of reconstructed images.

# 7. Results

We trained four neural network auto-encoders. The mean squared error and the mean absolute error for each parameter configuration is given in Table 2. There is a clear correspondence

DOI: http://dx.doi.org/10.17501.....

of the reconstruction error of the auto-encoder to the size of network. The complex-valued networks outperform the real-valued networks in regards to the mean squared error and mean absolute error, based on the number of parameters.

**Table 2: Parameters and errors for the two networks**

| Network | Parameters | Mean Squared Error | Mean Absolute Error |
|---|---|---|---|
| 1) Complex-Valued | 100,226 | 0,0050 | 0,0477 |
| 2) Real-Valued | 198,001 | 0.0047 | 0.0468 |
| 3) Complex-Valued | 397,442 | 0.0022 | 0.0320 |
| 4) Real-Valued | 790,945 | 0.0021 | 0.0313 |

The seismic sections in figure 7 show the unseen test seismic sections and the outputs of the real-valued and complex-valued neural network. Both auto-encoder outputs are blurred. The largest differences of the outputs in real-valued and complex-valued networks can be observed in discontinuous areas. The real-valued network smoothes over discontinuities and steep reflectors. This can also be seen in the central bottom fault block. Fault lines are imaged better in the complex-valued network output.

# 8. Discussion

We see from the results, that a real-valued network needs around twice as many parameters as a complex-valued network to attain the same reconstruction error. The reduction in number of parameters, means that a complex network can be trained on a smaller dataset without overfitting, than a real-valued network with the same performance.

In seismic data processing, including phase information stabilizes discontinuities and disambiguates cycle-skipping in horizons. Complex trace analysis enables the uses of instantaneous phase and amplitude attributes, which can give valuable information to human interpreters. We show that including phase information in deep neural networks improves the imaging of said discontinuities as well as steep reflectors, particularly in chaotic seismic textures that are strongly smoothed by real-valued neural networks.

Both the mean squared error and the mean absolute error is lower for the complex-valued network, when taking the number of parameters into account. This shows the advantage of including the phase information. Ideally, several random initialisations should have been made for each network, to allow for error bars on the estimates. However, the same pattern is seen for all four networks.

# 9. Conclusion

The inclusion of phase-informations leads to a better representation of seismic data in convolutional neural networks. This means a reduction both in the amount of training data needed and the size of the networks.

# 10. 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 DTU Compute for access to the GPU Cluster.

# 11. REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 60, no. 6, pp. 84–90, May 2012.

[2] J. S. Dramsch and M. Lüthje, "Deep-learning seismic facies on state-of-the-art CNN architectures," *SEG Tech. Progr. Expand. Abstr. 2018*, pp. 2036–2040, 2018.

[3] Ö. Yilmaz, *Seismic Data Analysis*. Society of Exploration Geophysicists, 2001.

[4] A. E. Barnes, "A tutorial on complex seismic trace analysis," *Geophysics*, vol. 72, no. 6, pp. W33–W43, 2007.

[5] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object Recognition with Gradient-Based Learning," no. 0, 1999, pp. 319–345.

[6] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep Complex Networks," no. 2016, pp. 1–19, May 2017.

[7] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," Feb. 2015

[8] G. E. Hinton, "Reducing the Dimensionality of Data with Neural Networks," *Science (80-. ).*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[9] Y. Alaudah, P. Michalowicz, M. Alfarraj, and G. AlRegib, "A Machine Learning Benchmark for Facies Classification," pp. 1–12, Jan. 2019.

.

*DISP '19, Oxford, United Kingdom*
*ISBN: 978-1-912532-09-4*

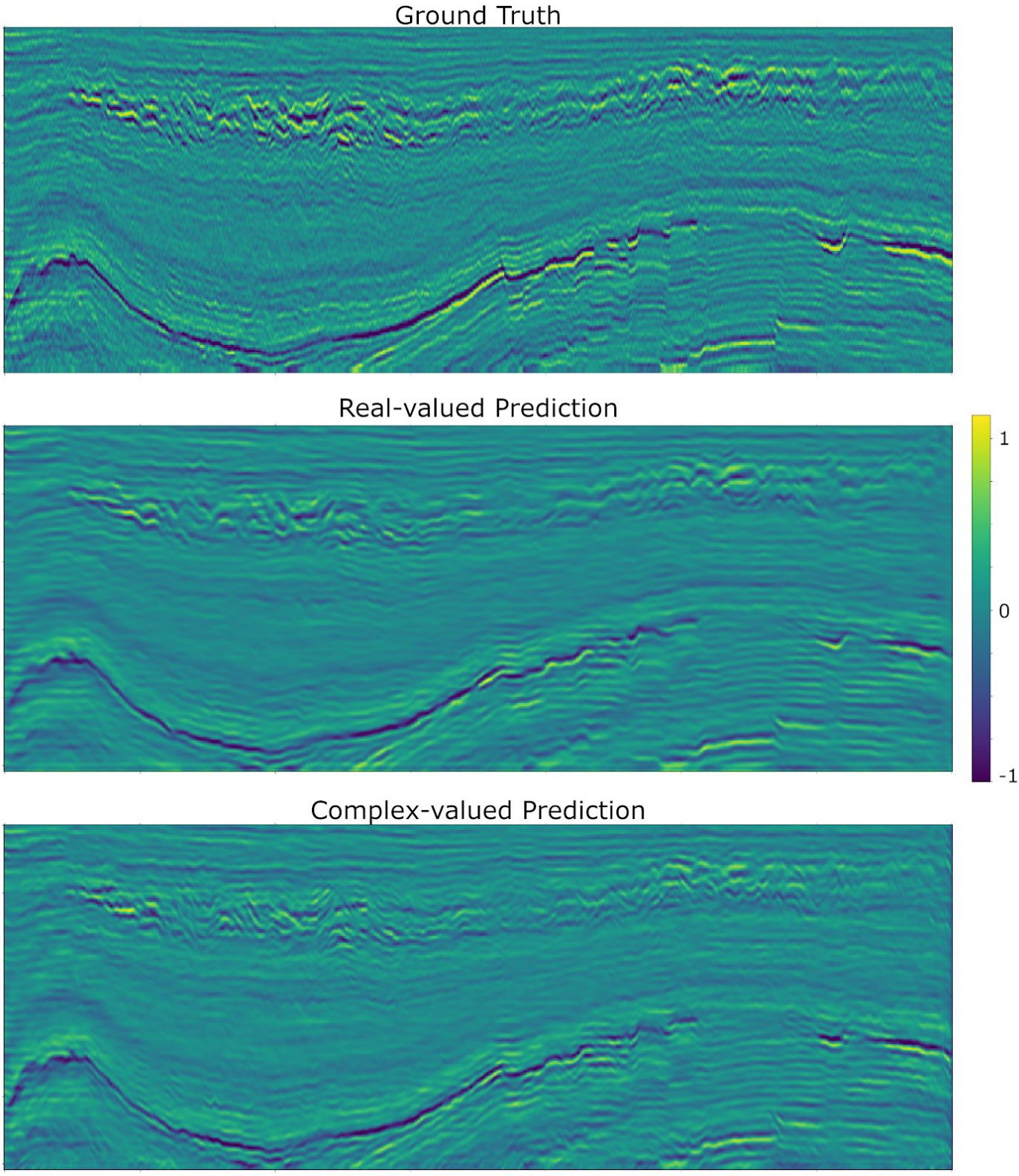


**Figure 7: Real-valued seismic sections, comparing Ground truth (top), Real-valued Network 2) prediction (middle), and Complex-valued Network 3) prediction (bottom).**

DOI: http://dx.doi.org/10.17501.....