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

Jesper Sören Dramsch*
Technical University of Denmark
Building 375, 2800 Kongens Lyngby
Denmark
jesper@dramsch.net

Mikael Lüthje
Technical University of Denmark
Building 375, 2800 Kongens Lyngby
Denmark
mikael@dtu.dk

Anders Nymark Christensen
Technical University of Denmark
Building 324, 2800 Kongens Lyngby
Denmark
anym@dtu.dk

Abstract

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. Furthermore, we show that the reduction of parameters in a complex network results in training on a smaller dataset without overfitting, in comparison to a real-valued network with the same performance.

1 Introduction

Seismic data is high-dimensional physical data. During acquisition, the data is collected 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 [Krizhevsky et al., 2012], geophysics has regained interest in automatic seismic interpretation (classification), and analysis of seismic signals. Through transfer learning, several initial successes were presented in Dramsch and Lüthje [2018b]. Nevertheless, seismic data has its caveats due to the

*Corresponding author.

Preprint. Under review.
complicated nature of bandwidth-limited wave-based imaging. Common problems are cycle-skipping of wavelets and nullspaces in inversion problems [Yilmaz 2001].

Automatic seismic interpretation is complicated, as the modelling of seismic data is computationally expensive and often proprietary. Seismic field data is mostly proprietary and their interpretation is highly subjective and 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 [Paganini et al. 2017].

The state-of-the-art method has been a iterative windowed Fourier transform for phase reconstruction [Griffin and Lim 1984]. Modern neural audio synthesis focuses on methods that do not require explicit reconstruction of the phase [Mehri et al. 2016] van den Oord et al. [2016, 2017] Prenger et al. 2018. Mehri et al. [2016] introduced a recurrent neural network formulation, where van den Oord et al. [2016] reformulated the synthesis network in a strided convolutional network. The original WaveNet formulation in van den Oord et al. [2016] is slow due to the autoregressive filter, warranting the parallel formulation in van den Oord et al. [2017].

We explicitly incorporate phase information in a deep convolutional neural network. These have been heavily explored in the digital signal processing community, before the recent renaissance of neural networks and deep learning. Relevant examples to seismic data processing include source separation [Scarpiniti et al. 2008], adaptive noise reduction [Suksmono and Hirose 2002] and optical flow [Miyauchi et al. 1993] with complex-valued neural networks. Sarroff [2018] gives a comprehensive overview of applications of complex-valued neural networks in signal and image processing.

In this work, we calculate the complex-valued seismic trace by applying the Hilbert transform to each trace. Phase information has been shown to be valuable in the processing [Liner 2002] and interpretation of seismic data [Roden and Sepulveda 1999] Mavko et al. 2003. Purves [2014] provides a tutorial that shows the implementation details of Hilbert transforms.

In this paper we give a brief overview of convolutional neural networks and then introduce the extension to complex neural networks and seismic data. 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.

2 Complex Convolutional Neural Networks

2.1 Basic principles

Convolutional neural networks [LeCun et al. 1999] use multiple layers of convolution and subsampling to extract relevant information from the data (see Figure 1).

![Figure 1: Schematic of equivalent complex- and real-valued convolutional neural network](image-url)
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 facies, 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.

2.2 Real- and Complex-valued 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

\[ f(t) \ast g(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \] (1)

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. Convolution is directly generalizable to N-dimensions by multiple integrations and maintains commutativity, distributivity, and associativity. In digital signals this extends to discrete values by replacing the integration with summation.

2.3 Complex Convolutional Neural Networks

Complex convolutional networks provide the benefit of explicitly modelling the phase space of physical systems [Trabelski et al., 2017]. The complex convolution introduced in Section 2.2 can be explicitly implemented as convolutions of the real and complex components of both kernels and the data. A complex-valued data matrix in cartesian notation is defined as \( \mathbf{M} = \mathbf{M}_\Re + i\mathbf{M}_\Im \) and equally, the complex-valued convolutional kernel is defined as \( \mathbf{K} = \mathbf{K}_\Re + i\mathbf{K}_\Im \). The individual coefficients \( (\mathbf{M}_\Re, \mathbf{M}_\Im, \mathbf{K}_\Re, \mathbf{K}_\Im) \) are real-valued matrices, considering vectors are special cases of matrices with one of two dimensions being one.

Solving the convolution of

\[ \mathbf{M}' = \mathbf{K} \ast \mathbf{M} = (\mathbf{M}_\Re + i\mathbf{M}_\Im) \ast (\mathbf{K}_\Re + i\mathbf{K}_\Im) \], (2)

we can apply the distributivity of convolutions (cf. section 2.2) to obtain

\[ \mathbf{M}' = \{\mathbf{M}_\Re \ast \mathbf{K}_\Re - \mathbf{M}_\Im \ast \mathbf{K}_\Im\} + i\{\mathbf{M}_\Re \ast \mathbf{K}_\Im + \mathbf{M}_\Im \ast \mathbf{K}_\Re\} \], (3)

where \( \mathbf{K} \) is the Kernel and \( \mathbf{M} \) is a data vector (see Figure 2).

We can reformulate this in algebraic notation

\[
\begin{bmatrix}
\Re\{\mathbf{M} \ast \mathbf{K}\} \\
\Im\{\mathbf{M} \ast \mathbf{K}\}
\end{bmatrix} =
\begin{bmatrix}
\mathbf{K}_\Re & -\mathbf{K}_\Im \\
\mathbf{K}_\Im & \mathbf{K}_\Re
\end{bmatrix} \ast
\begin{bmatrix}
\mathbf{M}_\Re \\
\mathbf{M}_\Im
\end{bmatrix}
\] (4)

Complex convolutional neural networks learn by back-propagation. Sarroff et al. [2015] state that the activation functions, as well as the loss function must be complex differentiable (holomorphic). Trabelski et al. [2017] suggest that employing complex losses and activation functions is valid for speed, however, refers that Hirose and Yoshida [2012] show that complex-valued networks can be optimized individually with real-valued loss functions and contain piecewise real-valued activations. We reimplement the code Trabelski et al. [2017] provides in keras [Chollet et al., 2015] with tensorflow [Abadi et al., 2015], which provides convenience functions implementing a multitude of real-valued loss functions and activations.
While max pooling and upsampling do not suffer from complex-valued neural networks, batch normalization [Ioffe and Szegedy, 2015] 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. Trabelsi et al. [2017] suggest implementing a 2D whitening operation as normalization in the following way.

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

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. This scales the covariance of the components instead of the variance of the data [Trabelsi et al., 2017].

2.4 Auto-encoders

Auto-encoders [Hinton and Salakhutdinov, 2006] are a special configuration of the 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 3. 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 then be used for anomaly detection, or classification.

3 Aliasing in Patch-based training

3.1 Mean-Shift in Neural Networks

A single neuron in a neural network can be described by \(\sigma(w \cdot x + b)\), where \(w\) is the network weights, \(x\) is the input data, \(b\) is the network bias, and \(\sigma\) is a non-linear activation function. During training, the network weights \(w\) and biases \(b\) are are adjusted to a value that represents the training minimum. Learning on a mean-shift of \(q\) of an arbitrary distribution over \(x\) leads to \(\sigma(w \cdot (x + q) + b)\), which increases the neuron response by \(q\), weighted by \(w\). During inference, both \(w\) and \(b\) are fixed, by extension the mean-shift \(q\) is fixed as well. The mean-shift over larger inference data disappears, introducing an additional bias of \(w \cdot q\) before non-linear activation. This training bias may lead to prediction errors of the neuron and consequently the full neural network.

3.2 Windowed Aliasing

Non-stationary data such as seismic data can contain sections within the data that contain spurious offsets from the mean. Figure 4 shows varying sizes of cutouts, with 101 and 256 samples respectively. In the middle, the full normalised amplitude spectra are presented. On the right, the corresponding phase spectra are presented. On the left, we focus on the frequency content of the amplitude spectra.
around 0 Hz. The cutouts were Hanning tapered, however, a mean shift appears with decreasing patch size.

These concepts of mean-shift corresponds to a DC offset in spectral data, which can be audio, seismic or electrical data. In images this corresponds to a non-zero alpha channel. While batch normalization can correct the mean shift in individual mini-batches [Ioffe and Szegedy 2015], this may shift the entire spectrum by the aliased offset. Additionally, batch normalization may not be feasible in some physical applications pertaining to regression tasks.

Figure 4: Spectral aliasing dependent on window-size (from [Dramsch and Lüthje 2018a])

4 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 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

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

(7)

of a real-valued time series \(u(t)\), 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 are set zero and the remaining frequencies are multiplied by 2. This can be written as

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

(8)

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 the 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 [Barnes 2007].
5 Experiments

5.1 Data

The data is the F3 seismic data, interpreted by Alaudah et al. [2019]. They provide a seismic benchmark for machine learning with accessible NumPy format. The interpretation (labels) of the seismic data is relatively coarse compared to conventional seismic interpretation, but the accessibility and pre-defined test case are compelling.

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.

5.2 Architecture

| Layer          | Real Output Shape | Complex Output Shape |
|----------------|-------------------|----------------------|
| Input          | 64 64 1           | 64 64 2              |
| Conv2D         | 64 64 8           | 64 64 16             |
| Conv2D + BN    | 64 64 8           | 64 64 16             |
| MaxPooling2D   | 32 32 8           | 32 32 16             |
| Conv2D + BN    | 32 32 16          | 32 32 32             |
| MaxPooling2D   | 16 16 32          | 16 16 64             |
| Conv2D + BN    | 16 16 32          | 16 16 64             |
| MaxPooling2D   | 8 8 32            | 8 8 64               |
| Conv2D + BN    | 8 8 64            | 8 8 128              |
| MaxPooling2D   | 4 4 64            | 4 4 128              |
| Conv2D         | 4 4 128           | 4 4 256              |
| Upsampling2D   | 8 8 128           | 8 8 256              |
| Conv2D + BN    | 8 8 64            | 8 8 128              |
| Upsampling2D   | 16 16 64          | 16 16 128            |
| Conv2D + BN    | 16 16 32          | 16 16 64             |
| Upsampling2D   | 32 32 32          | 32 32 64             |
| Conv2D + BN    | 32 32 16          | 32 32 32             |
| Upsampling2D   | 64 64 16          | 64 64 32             |
| Conv2D         | 64 64 8           | 64 64 16             |
| Conv2D + BN    | 64 64 8           | 64 64 16             |
| Conv2D         | 64 64 1           | 64 64 2              |
| Parameters     | 198,001           | 100,226              |

Table 1: Layers used in the auto-encoder

The Auto-encoder architecture uses 2D convolutions with 3x3 kernels. We employ batch normalization to regularize the training and speed up training [Ioffe and Szegedy, 2015]. The down and up sampling is achieved by MaxPooling and the UpSampling operation, respectively. 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).

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.

5.3 Training

We train the networks with an Adam optimizer and a learning rate of $10^{-3}$ without decay, 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, shown in Figure 5.
5.4 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 in figure 6 on 7 individual realizations of the respective four networks.

6 Results

| Network | Parameters | MSE  | MAE  |
|---------|------------|------|------|
| 1) $C_{small}$ | 100,226 | 0.0050 | 0.0477 |
| 2) $R_{small}$ | 198,001 | 0.0047 | 0.0468 |
| 3) $C_{large}$ | 397,442 | 0.0022 | 0.0320 |
| 4) $R_{large}$ | 790,945 | 0.0021 | 0.0313 |

Table 2: Parameters and errors for networks

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 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.

The seismic sections in figure 6 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 smooths 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.

7 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. We trained seven random initializations for each network, to allow for error bars on the estimates in Figure 5a. In figure 5b we scale the loss to the number of parameters in the respective network architectures.

Figure 6: Real-valued seismic sections, comparing Ground truth (top), Real-valued Network 2) prediction (middle), and Complex-valued Network 3) prediction (bottom).

8 Conclusion

The inclusion of phase-information leads to a better representation of seismic data in convolutional neural networks. Disregarding phase information may lead to low frequency aliasing, dependent on convolutional kernel size. Complex-valued networks outperform real-valued neural networks scaled on parameters.
9 Acknowledgments

We thank Andrew Ferlitsch for his valuable insights. 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.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernando Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL http://tensorflow.org/ Software available from tensorflow.org.

Yazeed Alaudah, Patrycja Michalowicz, Motaz Alfarraj, and Ghassan AlRegib. A machine learning benchmark for facies classification. arXiv preprint arXiv:1901.07659, 2019.

Arthur E Barnes. A tutorial on complex seismic trace analysis. Geophysics, 72(6):W33–W43, 2007.

François Chollet et al. Keras. https://keras.io 2015.

Jesper Sören Dramsch and Mikael Lüthje. Information theory considerations in patch-based training of deep neural networks on seismic time-series. In First EAGE/PESGB Workshop Machine Learning. EAGE Publications BV, November 2018a. doi: 10.3997/2214-4609.201803020. URL https://doi.org/10.3997/2214-4609.201803020

Jesper Sören Dramsch and Mikael Lüthje. Deep-learning seismic facies on state-of-the-art cnn architectures. In SEG Technical Program Expanded Abstracts 2018, pages 2036–2040. Society of Exploration Geophysicists, 2018b.

Daniel Griffin and Jae Lim. volume 32, pages 236–243. Institute of Electrical and Electronics Engineers, 1984. doi: 10.1109/icassp.1983.1172092. URL https://doi.org/10.1109/icassp.1983.1172092

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. science, 313(5786):504–507, 2006.

Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron C. Courville, and Yoshua Bengio. SamplerNN: An unconditional end-to-end neural audio generation model. CoRR, abs/1612.07837, 2016. URL http://arxiv.org/abs/1612.07837

Minami Miyauchi, Masatoshi Seki, Akira Watanabe, and Arata Miyauchi. Interpretation of optical flow through complex neural network. In New Trends in Neural Computation, pages 645–650. Springer Berlin Heidelberg, 1993. doi: 10.1007/3-540-56798-4_215. URL https://doi.org/10.1007/3-540-56798-4_215

Michela Paganini, Luke de Oliveira, and Benjamin Nachman. CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks. 2017.
Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. CoRR, abs/1811.00002, 2018. URL http://arxiv.org/abs/1811.00002.

Steve Purves. Phase and the hilbert transform. The Leading Edge, 33(10):1164–1166, October 2014. doi: 10.1190/tle33101164.1. URL https://doi.org/10.1190/tle33101164.1.

Rocky Roden and Héctor Sepúlveda. The significance of phase to the interpreter: Practical guidelines for phase analysis. The Leading Edge, 18(7):774–777, July 1999. doi: 10.1190/1.1438375. URL https://doi.org/10.1190/1.1438375.

Andy M. Sarroff. Complex Neural Networks for Audio. Technical Report TR2018-859, Dartmouth College, Computer Science, Hanover, NH, May 2018. URL http://www.cs.dartmouth.edu/~trdata/reports/TR2018-859.pdf.

Andy M. Sarroff, Victor Shepardson, and Michael A. Casey. Learning representations using complex-valued nets. CoRR, abs/1511.06351, 2015. URL http://arxiv.org/abs/1511.06351.

Michele Scarpiniti, Daniele Vigliano, Raffaele Parisi, and Aurelio Uncini. Generalized splitting functions for blind separation of complex signals. Neurocomputing, 71(10-12):2245–2270, June 2008. doi: 10.1016/j.neucom.2007.07.037. URL https://doi.org/10.1016/j.neucom.2007.07.037.

Andriyan Bayu Suksmono and Akira Hirose. Adaptive noise reduction of InSAR images based on a complex-valued MRF model and its application to phase unwrapping problem. IEEE Transactions on Geoscience and Remote Sensing, 40(3):699–709, March 2002. doi: 10.1109/tgrs.2002.1000329. URL https://doi.org/10.1109/tgrs.2002.1000329.

Chiheb Trabelsi, Olexa Bilaniuk, Ying Zhang, Dmitriy Serdyuk, Sandeep Subramanian, João Felipe Santos, Soroush Mehri, Negar Rostamzadeh, Yoshua Bengio, and Christopher J Pal. Deep complex networks. arXiv preprint arXiv:1705.09792, 2017.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. CoRR, abs/1609.03499, 2016. URL http://arxiv.org/abs/1609.03499.

Aäron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C. Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman, Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters, Dan Belov, and Demis Hassabis. Parallel wavenet: Fast high-fidelity speech synthesis. CoRR, abs/1711.10433, 2017. URL http://arxiv.org/abs/1711.10433.

Özdoğan Yilmaz. Seismic data analysis, volume 1. Society of exploration geophysicists Tulsa, OK, 2001.