

Deep Convolutional Neural Networks in prestack seismic—two exploratory examples

Ali Siahkoobi¹, Mathias Louboutin¹, Rajiv Kumar², and Felix J. Herrmann^{1,2}

¹*School of Computational Science and Engineering, Georgia Institute of Technology,*

²*School of Earth and Atmospheric Sciences, Georgia Institute of Technology*

ABSTRACT

Deep convolutional neural networks are having quite an impact and have resulted in step changes in the capabilities of image and voice recognition systems and there are indications they may have a similar impact on post-stack seismic images. While these developments are certainly important, we are after all dealing with an imaging problem involving an unknown earth and not completely understood physics. When the imaging step itself is not handled properly, this may possibly offset gains offered by deep learning. Motivated by our work on deep convolutional neural networks in seismic data reconstruction, we discuss how generative networks can be employed in prestack problems ranging from the relatively mundane removal of the effects of the free surface to dealing with the complex effects of numerical dispersion in time-domain finite differences. While the results we present are preliminary, there are strong indications that generative deep convolutional neural networks can have a major impact on complex tasks in prestack seismic data processing and modeling for inversion.

INTRODUCTION

There have been lots of recent successes applying machine learning techniques to speech recognition and image classification (Krizhevsky et al., 2012; Hinton et al., 2012; Szegedy et al., 2015) including in seismic (Rawles and Thurber, 2015; Zhao et al., 2015; Mousavi et al., 2016; Chen, 2017; Shafiq et al., 2018). While there is ample evidence that machine learning and deep convolutional neural networks (CNNs) in particular have or will very soon lead to major breakthroughs in “post-stack” seismic, the application of this exciting new technology to “prestack” seismic, whether this applies to seismic data processing or wave-equation based imaging and inversion is not clear. This paper aims to gain some insight into their potential.

While several efforts are currently underway to incorporate CNNs in imaging problems—by using them to parameterize prior probability density functions that serve as prior information on the model in inverse problems (Dokmanić et al., 2016); or by learning certain prox operators (read denoisers) in optimization (Sun et al., 2016; Fai et al., 2017; Mardani et al., 2017; Adler and Öktem, 2018) to name a few—the authors are not aware of studies that elaborate on the use of CNN’s in problems that involve complex and high-dimensional wave physics. Having said this there is a growing body of work incorporating ideas from machine learning into physical systems—e.g. there have been successes in the prediction of production curves from large production histories (Li et al., 2013; Aizenberg et al., 2016; Anderson et al., 2016) or in learning (nonlinear) physics from observed data (Brunton et al., 2016). But again, these examples do rely on extremely large training sets in combination with underlying manifolds that are (extremely) low dimensional.

This brings us to the topic of this contribution where we ask

ourselves the question “*what have CNNs to offer in seismic data processing and modeling for inversion?*” To answer this question we explore the use of generative networks—i.e., networks that are not trained as a classifier but instead are capable of generating examples drawn from a probability distribution that may include a certain mapping. When properly trained, and therein lies of course the challenge, these generative networks can lead to remarkable results, see for instance recent work by Siahkoobi et al. (2018) where we recovered seismic data successfully from missing traces to a high degree of accuracy.

To further explore the potential of generative networks, we consider two different tasks for which we do have an answer namely, removal of the effects of the free surface (ghost + surface related multiples) and the removal of numerical dispersion in finite-difference modeling. Both tasks serve as proxies on how CNNs can be used to deal with inadequate physics—i.e. physical systems that for some reason are not described adequately either by not including the free surface in our formulation or by violating the stability criterion of finite differences.

As the examples demonstrate, CNNs show a remarkable ability to conduct complicated tasks even when trained on different models—i.e., a different water depth for the multiples or a different part of the velocity model for the dispersion. We feel that this opens exciting new possibilities to rethink how we formulate our problems.

We organize our work as follows. We first give a brief introduction to CNNs followed by a detailed discussion on the application of CNNs to the removal of the effects of the free surface and numerical dispersion effects in modeling.

THEORY

In general, for given parameters θ , CNNs consist of several *layers* where within each layer the input is convolved with multiple kernels, followed by either up or down-sampling, and the elementwise application of a non-linear function (e.g. thresholding ...). Let $Z_0 \in \mathbb{R}^{m_0 \times n_0 \times k_0}$ be the input to the first layer. The i^{th} layer with input Z_{i-1} and output Z_i of a CNN consisting of N layers is given by:

$$Z_i^{(j)} = h \left(\left(A_i^{(j)} \circ S_i \right) * Z_{i-1} + B_i^{(j)} \right) \quad \text{for } j = 1, \dots, k_i, \quad (1)$$
$$i = 1, \dots, N.$$

where for all j in i^{th} layer, $A_i^{(j)} \in \mathbb{R}^{r_i \times s_i \times k_{i-1}}$ is the j^{th} convolution kernel, k_i is the number of kernels and kernels only slide over the first two dimensions during convolution. S_i is an up or down-sampling operator, $B_i^{(j)} \in \mathbb{R}^{m_i \times n_i}$ is j^{th} bias variable with same size as the result of convolution, and h is the elementwise non-linear function. The output of i^{th} layer is $Z_i = [Z_i^{(1)}, Z_i^{(2)}, \dots, Z_i^{(k_i)}] \in \mathbb{R}^{m_i \times n_i \times k_i}$ where the size of the third dimension of the output is equal to the number of convolutional kernels used in i^{th} layer. The operator S_i , in case of

down-sampling, performs the convolution with more than one shift in all directions. In the case of up-sampling, it periodically puts zeros between samples of Z_{i-1} and then performs the convolution with usual one shift. The set of parameters θ for the described CNN consists of the convolution kernels and biases in all the layers of CNN. During training, for given input and output pairs, θ gets optimized.

We implemented our algorithm in TensorFlow (Abadi et al., 2016) and used Devito (Lange et al., 2017; Louboutin et al., 2017) to solve the wave equation using finite difference method.

EXPERIMENTS

It is obvious that there are many potential areas where CNNs could be employed in seismic data processing and modeling for inversion. As a first example to demonstrate the potential of CNNs, we select removal of the free surface effects because this problem is well understood and widely studied. Our goal here is to see whether a CNN can learn the mapping from data generated with a free surface to data without a free surface. Since the effects of the free surface only depend on the free surface itself, we expect the network to in principle generalize well—i.e., to work on data generated from a different velocity model.

As a second example, we study a possible map from numerically dispersed solutions of the wave equation to non-dispersed solutions. While a solution for this problem exists, namely modeling with sufficient numbers of gridpoints per wavelength, we are interested to see whether a trained CNN is capable to perform this far from trivial task.

Disclaimer. We are fully aware that powerful solutions exists for both tasks and it is not our claim to say we can do better or that this will ever work competitively in practice. We are merely interested in exploring potential capabilities of CNNs in prestack seismic data processing and modeling for inversion.

Example 1. Removal of the free surface

We train our CNN on shot records generated with a quasi-oracle velocity model (a training model close to the actual model) that has 25 meters difference with the oracle (= true velocity model) velocity in the water depth. In order to learn the relationship between the primaries and multiples and ghosted and deghosted data, we need to have shot records both with and without a free surface in our training dataset. Training is done over all 401 shots and we pass twenty times over the entire dataset. While we see all shots, we are training on a velocity model with a significantly different water depth.

Once the training is finished, our CNN maps a shot from the 2D dataset generated with a free surface to a shot without a free surface in about 1.8 seconds. This is very fast if we exclude the time it took to train the CNN, which is of the same order as estimation of primaries by sparse inversion.

Figure 5 summarizes our results by showing the data without and with the free surface and with direct waves included. Figure 5b contains the output of the trained network mapping data generated with the free surface (Figure 5c) to data w/o a free surface (Figure 5a)—i.e., a ghost and w/o surface related multiples. Comparing the mapped data to the modeled data w/o a

free surface shows that the CNN did a remarkable good job, an observation supported by the difference between the modeled and mapped data w/o a free surface (Figure 5d); the trace comparisons in Figure 1, and the f-k spectra in Figure 2. Aside from removing most of the multiple energy, the CNN also restores the low-end of the spectrum. However, the mapped result (juxtapose the two top figures of Figure 5) are certainly not perfect. We miss energy from the strong Ocean bottom reflection; from the direct wave; and from areas where there are conflicting dips (late arrival times and far offsets). Having said that these results are certainly encouraging because they were obtained more or less out of the box. We also need to remember that we are also deghosting the data.

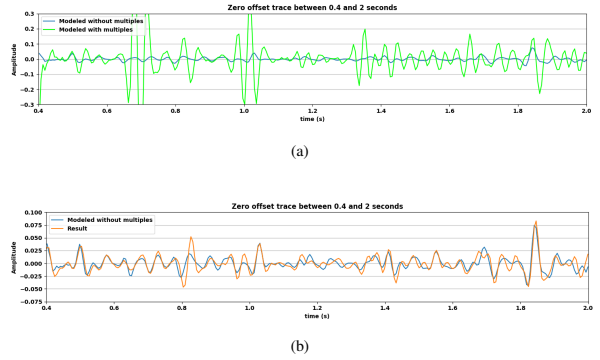


Figure 1: Comparing reflections present in zero-offset trace modeled with and without free surface multiples and result obtained by eliminating the multiples using trained CNN on velocity model with different water depth. a) comparison between zero-offset trace modeled with and without free surface multiples. b) comparison between zero-offset trace modeled without free surface multiples and the obtained result.

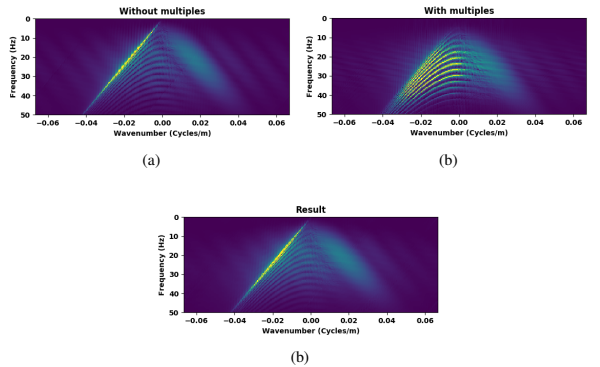


Figure 2: F-K spectrum of shot gathers without multiples, with multiples, and the multiple elimination result.

Example 2. Removal of numerical dispersion

While the results of the previous example are certainly encouraging, deghosting and surface-related removal are relatively mundane operations compared to modeling for inversion that involves the solution of the wave equation. We use the term “modeling for inversion” to underline that we are often work-

ing with modeling kernels that are designed to fit observed data during an iterative inversion process. For obvious reasons, wave-equation based inversions are built on the premise that our simulations for inversion capture the physics accurately, an assumption that may not always be justified. To mimic situations where we are not capturing the right physics, we simulate waves in acoustic models (depicted in Figure 3) with only a two-point spatial finite difference stencil. By doing this, we deliberately violate the stability criterion of this numerical approximation of the wave equation. Because of this violation, we expect this low-fidelity approximation to be dispersive compared to high-fidelity simulations carried out with a 20-point stencil in the spatial directions (cf. Figures 6a and 6b).

Our goal now is to train a CNN to map dispersed wavefields to non-dispersed wavefields, a task that is not simple given the complexities of the wavefields and the non-trivial imprint of numerical dispersion. To make things more difficult and also more realistic, and to make sure the CNN is not simply memorizing what the correct non-dispersed wavefield should look like, we train the network on a different velocity model from the same geological area. To this end, we extract two slices from the 3D Marmousi model that — aside from having different water depths (this is where expect a lot of dispersion) — exhibit a completely different velocity structure albeit they were both drawn from the same geological area. We use the velocity model in Figure 3 (top) to train the CNN on pairs of wavefields (Figure 6a and 6b) and then test the trained CNN on wavefields generated from Figure 3 (bottom).

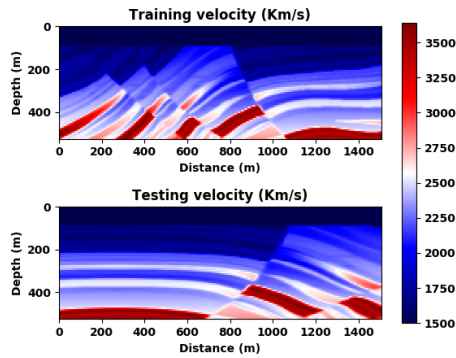
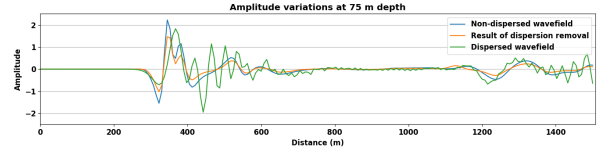
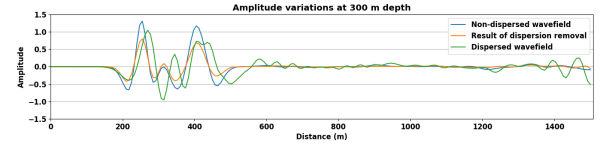


Figure 3: Training (top) and testing velocity (bottom).

A representative time-snapshot of the non-dispersed high-fidelity test data is plotted in Figure 6c. Its low-fidelity counterpart is plotted in Figure 6d and shows, as expected, a heavy imprint of numerical dispersion. In Figure 6e, we plot the mapped result by the CNN. While certainly not perfect (see also Figure 4), the CNN demonstrates a remarkable ability to carry out this non-trivial mapping even through it was trained on a different velocity model. While there are differences in the amplitude, the phase of the different events in this complex wavefield are mostly restored and the detrimental effects of the numerical dispersion is indeed removed.



(a) Amplitudes at 75 m depth.



(b) Amplitudes at 340 m depth.

Figure 4: Comparing the amplitude variations of non-dispersed wavefield, dispersed wavefield, and result obtained by numerical dispersion removal using trained CNN on training velocity model.

DISCUSSION & CONCLUSIONS

While it still early days on the application of deep convolutional neural networks to “pres-stack seismic”, we feel that the examples included in this work indicate that this technology has the potential to be truly transformational. Of course its eventual success will depend on our ability to train the networks; to define networks that scale to large problem sizes and that can handle noise; and above all to come up with networks that generalize to a sufficient degree. If we succeed in meeting these goals, there is no doubt in our minds that this technology will have the potential to completely revolutionize seismic data processing and inversion. Not only will we be able to remove certain unwanted components from the data (e.g. multiples + ghosts) but we will also be able to handle situations where we only have access to low-fidelity physics to carry out our inversions.

While pedagogical, the numerical dispersion example has clearly demonstrated to us that neural networks can compensate for unmodeled intricate physics. This means that as long as we are able to keep things “simple but not too simple” to quote Albert Einstein, we are in a position to make great strides forward by using networks to handle the often intractable intricacies of geophysical data while we focus on extracting valuable and interpretable information on the leading order physics. We have to say these are exciting times.

ACKNOWLEDGEMENTS

This research was carried out within Georgia Institute of Technology, School of Computational Science and Engineering.

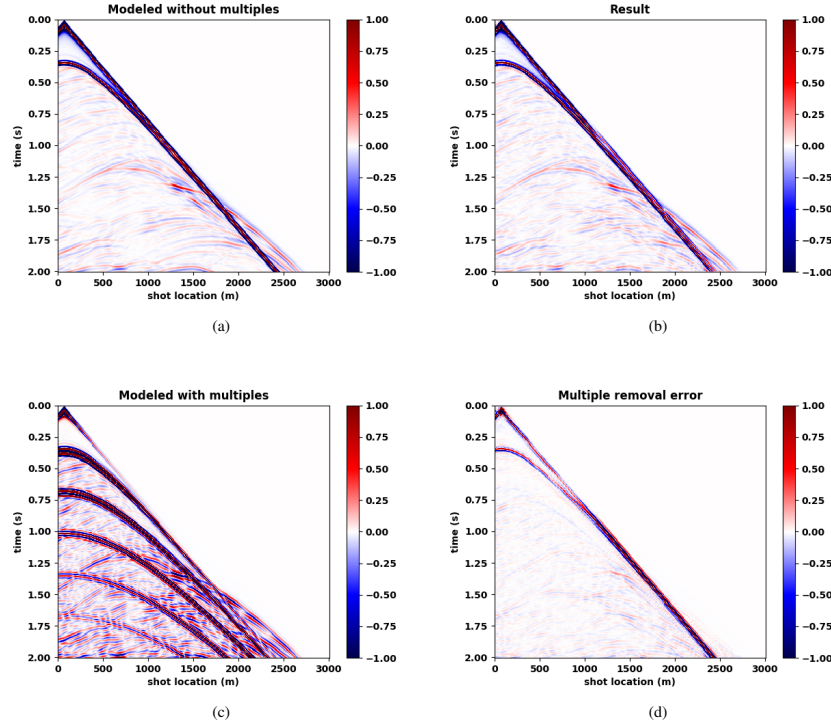


Figure 5: Multiple elimination results. a) Shot record modeled without surface related multiples. b) Multiple elimination result using CNN trained on velocity model with different water depth. c) Shot record modeled with surface related multiples. d) Difference between (a) and result.

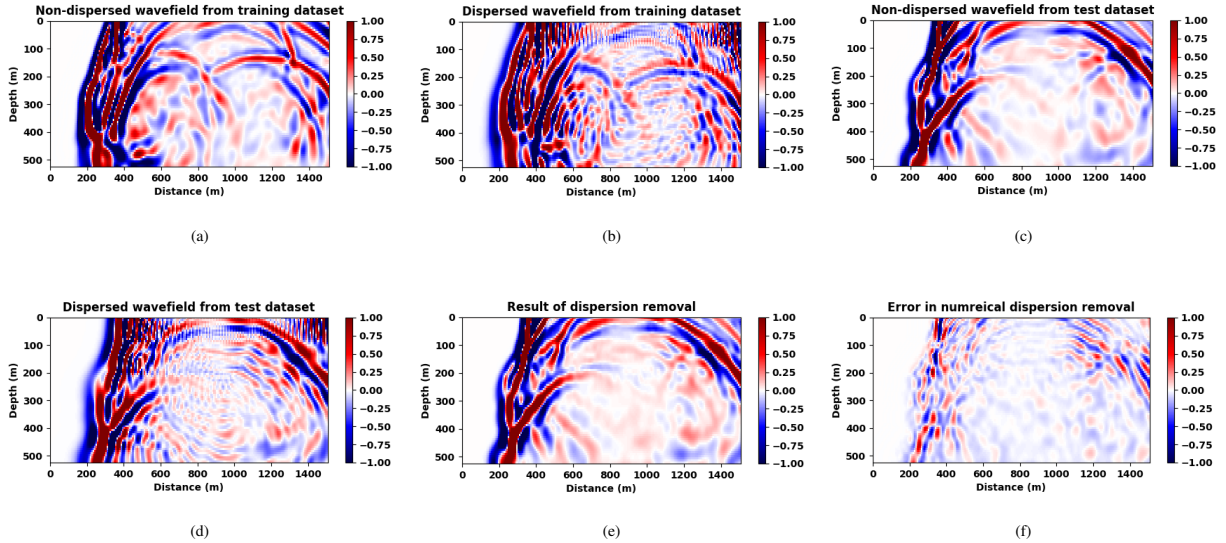


Figure 6: Numerical dispersion result. a) Wavefield generated with 20-point spatial finite difference stencil using training velocity. b) Wavefield generated with 2-point spatial finite difference stencil using training velocity at the same shot location and same time step as wavefield in (a). c) Wavefield generated with 20-point spatial finite difference stencil using testing velocity at the same shot location and same time step as wavefield in (a) and (b). d) Wavefield generated with 2-point spatial finite difference stencil using testing velocity at the same shot location and same time step as wavefield in (a) and (b). e) Removing numerical dispersion in (d) using CNN trained on training velocity with the aim to predict (c). f) Difference between (c) and (e).

REFERENCES

- Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., 2016, Tensorflow: A system for large-scale machine learning.: OSDI, 265–283.
- Adler, J., and O. Öktem, 2018, Learned primal-dual reconstruction: IEEE Transactions on Medical Imaging.
- Aizenberg, I., L. Sheremetov, L. Villa-Vargas, and J. Martinez-Muñoz, 2016, Multilayer neural network with multi-valued neurons in time series forecasting of oil production: Neurocomputing, **175**, 980–989.
- Anderson, R. N., B. Xie, L. Wu, A. A. Kressner, J. H. Frantz Jr, M. A. Ockree, and K. G. Brown, 2016, Petroleum analytics learning machine to forecast production in the wet gas marcellus shale: Presented at the , Unconventional Resources Technology Conference (URTEC).
- Brunton, S. L., J. L. Proctor, and J. N. Kutz, 2016, Discovering governing equations from data by sparse identification of nonlinear dynamical systems: Proceedings of the National Academy of Sciences, **113**, 3932–3937.
- Chen, Y., 2017, Automatic microseismic event picking via unsupervised machine learning: Geophysical Journal International, **112**, 88–102.
- Dokmanić, I., J. Bruna, S. Mallat, and M. de Hoop, 2016, Inverse problems with invariant multiscale statistics: arXiv preprint arXiv:1609.05502.
- Fai, K., Q. Wei, L. Carin, and K. Heller, 2017, An inner-loop free solution to inverse problems using deep neural networks.
- Hinton, G., L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., 2012, Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups: IEEE Signal Processing Magazine, **29**, 82–97.
- Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012, Imagenet classification with deep convolutional neural networks: Advances in neural information processing systems, 1097–1105.
- Lange, M., N. Kukreja, F. Luporini, M. Louboutin, C. Yount, J. Hückelheim, and G. J. Gorman, 2017, Optimised finite difference computation from symbolic equations: Proceedings of the 15th Python in Science Conference, 89 – 96.
- Li, X., C. Chan, and H. Nguyen, 2013, Application of the neural decision tree approach for prediction of petroleum production: Journal of Petroleum science and engineering, **104**, 11–16.
- Louboutin, M., P. Witte, M. Lange, N. Kukreja, F. Luporini, G. Gorman, and F. J. Herrmann, 2017, Full-waveform inversion, part 1: Forward modeling: The Leading Edge, **36**, 1033–1036.
- Mardani, M., H. Monajemi, V. Pappan, S. Vasanawala, D. Donoho, and J. Pauly, 2017, Recurrent generative adversarial networks for proximal learning and automated compressive image recovery: arXiv preprint arXiv:1711.10046.
- Mousavi, S. M., S. P. Horton, C. A. Langston, and B. Samei, 2016, Seismic features and automatic discrimination of deep and shallow induced-microearthquakes using neural network and logistic regression: Geophysical Journal International, **207**, 29–46.
- Rawles, C., and C. Thurber, 2015, A non-parametric method for automatic determination of p-wave and s-wave arrival times: application to local micro earthquakes: Geophysical Journal International, **202**, 1164–1179.
- Shafiq, M. A., Z. Long, H. Di, and G. AlRegib, 2018, Attention models in perception and interpretation of seismic volumes: Geophysics.
- Shahkoobi, A., R. Kumar, and F. J. Herrmann, 2018, Seismic data reconstruction with generative adversarial networks: Presented at the 80th EAGE Conference and Exhibition 2018.
- Sun, J., H. Li, Z. Xu, et al., 2016, Deep admm-net for compressive sensing mri: Advances in Neural Information Processing Systems, 10–18.
- Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, et al., 2015, Going deeper with convolutions: Presented at the , Cvpr.
- Zhao, T., V. Jayaram, A. Roy, and K. J. Marfurt, 2015, A comparison of classification techniques for seismic facies recognition: Interpretation, **3**, SAE29–SAE58.