Direct velocity estimation for seismic imaging using deep neural network

Stuart Farris
Department of Geophysics
Stanford University
sfarris@stanford.edu

Vishal Das
Department of Geophysics
Stanford University
vdas2@stanford.edu

Abhishek Tapadar
Department of Civil and Environmental Engineering
Stanford University
atapadar@stanford.edu

Abstract

We train a convolutional neural network (CNN), augmented by a long short-term memory (LSTM) unit, to predict one-dimensional, root-mean-squared (RMS) pressure wave velocity profiles of the earth's subsurface from two-dimensional geophysical data. The results from this network implementation are a promising proof of concept for approaching geophysical problems from a deep learning perspective.

1 Introduction

Retrieving the earth's material properties in the subsurface is necessary for a variety of applications including earthquake hazard prediction, landslide nucleation, shallow gas hazard identification, civil engineering projects, and hydrocarbon identification. A robust and widely used method in the earth science community to investigate the subsurface is exploration seismology [Yilmaz, 2001]. This method, analogous to an ultrasound in the medical community, uses as active source of energy to measure how elastic energy travels through the earth in order recover properties such as wave velocity, rock density, anisotropy, and fluid content.

Figure 1 illustrates a mock seismic survey, recorded data, and earth model inverse solution. The left plot in Figure 1 illustrates a slice of the earth with a seismic survey taken at the surface. Emitting from the controlled source point, energy travels through the subsurface, reflects off rock boundaries, and is recorded as it returns to the surface and point receivers. This process is repeated many times to produce recorded data gathers as seen in the center of Figure 1. Using many of these recorded gathers, the task of the geophysicists it to formulate and solve an inverse problem to recover the elastic properties of the earth.. The earth property of interest here, and in a large portion of the explorations geophysics community, is pressure wave velocity as described in Aki and Richards.

2 Related work: Geophysics baseline and similar deep learning projects

2.1 Geophysics baseline: normal moveout velocity analysis

The baseline method for estimating one-dimensional wave velocity profiles from two-dimensional seismic data is normal moveout (NMO) velocity analysis, as described in Claerbout [2010, chap. 4]. This method has the exact same inputs and outputs as our deep learning approach.

Pioneered in the 1970's, NMO analysis relies on redundant data measurements around a common illumination point in the earth and computes stacking power over a variety of hyperbolic wave trajectories to determine an optimal one-dimensional RMS

CS230: Deep Learning, Autumn 2018, Stanford University, CA.

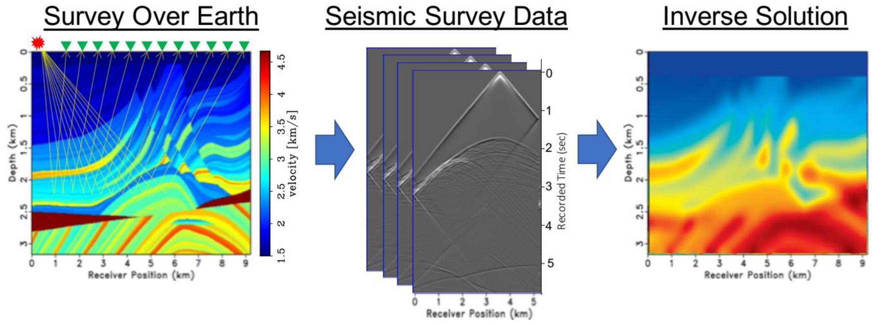

Figure 1: Left: An illustration of the two-dimensional wave velocity profile of the earth. At the surface of the earth, a controlled source of energy is emitted from the red dot. The energy travels though the earth and is recorded at point receivers, green triangles, on the surface. Middle: data recorded from many seismic experiments. Each two-dimensional panel is a recording from one seismic experiment. Right: a realistic reconstruction of the "true" earth model seen on the left from some geophysics inversion scheme.

velocity profile. The method is flawed for a variety of reasons. Namely, it is sensitive to noise in the data space, it assumes the earth is composed of homogeneous, flat layers (a gross assumption), and requires an experienced geophysicist to hand pick the final RMS velocity profile. This all makes NMO velocity analysis noisy, inaccurate, and time consuming. Despite this, it is still common practice to perform NMO analysis to make initial earth models that are inputs for more advanced geophysical inversion schemes. An outdated, tedious, human-intensive image processing task... maybe a perfect problem to apply a deep learning approach?

2.2 Deep learning baseline: a lack thereof

There is no deep learning baseline model for what we are attempting here. No one has attempted to solve this exact problem with neural networks or deep learning.

Although, the geophysics community has attempted to solve other classic earth science problems from a deep learning perspective. Das et al. [2018] showed the use of CNN for predicting impedance from 1D seismic data. Araya-Polo et al. [2018] predicted two-dimensional wave velocity profiles from a three-dimensional feature space that is a derivative of seismic data. But, we believe that by using a derivative of the seismic data space, this CNN solution is losing out on information contained in the raw seismic data. This is what lead us to use the raw seismic data as the feature space of our own solution. We will see that our approach is an end-to-end solution that solves a slightly simpler problem (we predict 1D profiles where they predict 2D profiles). Farris et al. [2018] showed that the solution of Araya-Polo et al. [2018] is attempting to solve the same problem as Full Waveform Inversion (FWI), the favored inversion scheme in the modern exploration community Tarantola [1984].

Ma et al. [2018] solves a problem very similar to ours with a CNN. They show good results predicting earth models with a very deep CNN. But they only predict very simple earth velocity profiles that leaves something to be desired. We hope to make predictions over more complex models.

3 Dataset and Features

3.1 An unlabeled earth

Many seismic experiments have been conducted in the real world. There is a plethora of data, but unfortunately it is all *unlabeled*. That is, since we never know the true earth model, we will never have an accurate label to pair with all the seismic data we produce. In fact, generating accurate labels (earth models) is the primary goal of exploration seismology. If we knew what the earth looked like beneath the surface their would be no need to generate all the seismological data!

3.2 Label and data generation

Fortunately, we have prior geophysical knowledge of realistic earth models and a firm grasp of how elastic energy travels through the earth, allowing us to generate our own data and labels [Aki and Richards]. We create psuedo-random, two-dimensional earth models that are a simplified version of the true earth, but fairly represent the wave velocity profile and layering structure we know exists in reality. The center of Figure 2 is an example of such an earth model. In each generated earth model, we solve the wave equation and propagate elastic energy, recording at the surface. This simulates a real life seismic experiment. An example

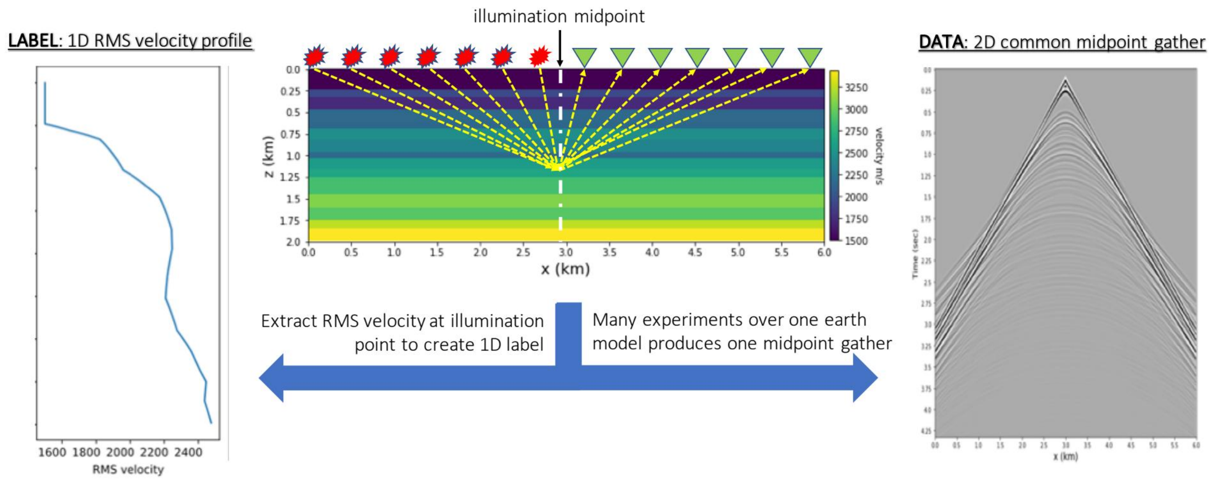

Figure 2: An example of a how a single label/data pair are created. The center image is a psuedo-random, two-dimensional earth model with assumed flat, homogeneous layers. Repeated seismic measurements are taken over the 2D model in order to produce a seismic midpoint gather. This gather is seen on the right hand side and is the input data generated from this example. At the center of the survey, at the common illumination point, a one-dimensional RMS velocity profile is extracted from the two-dimensional model and illustrated on the left. This is label generated from this example.

of our generated earth model and a corresponding seismic gather are illustrated in Figure 2. Since the earth models currently contain flat layers we can represent them with a one-dimensional RMS profile and use this as the label for each seismic gather. We are using an RMS profile of the velocity instead of true velocity because we are trying to directly compare our results to the baseline geophysical method, NMO velocity analysis, described above.

We generate 10,000 psuedo-random earth models and corresponding data/label pairings. The earth models are discretized with 10 meter spacing and are 200x600 samples in depth and width, respectively. To generate the seismic data, the wave equation code from Fabien-Ouellet and Gloaguen, Erwan Giroux [2016] was used to forward propagate a Ricker wavelet for 4.33 seconds at 0.01 second time discretization [SEG]. The wave equation propagation was run on eight Nvidia V100 GPUs and took about 24 hours to finish. Each data sample was then stored, for training the networks.

4 Methods

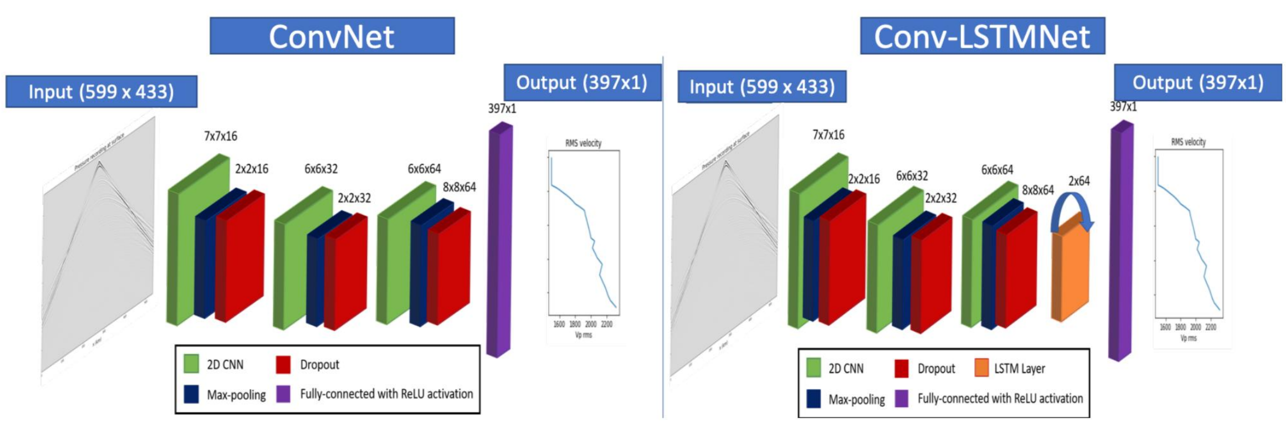

Figure 3: Neural network architectures used in the project. The left architecture is a deep convolutional neural network (DCNN) and the right architecture is DCNN with Long Short-term memory (LSTM). The dimensions of the input and output along with the dimensions of the layers are also specified in the figure.

4.1 Overall intuition

In seismic processing work-flow, curvature in the seismic gathers is a good indication of the velocity of seismic waves propagation in the earth's subsurface. In order to capture the curvature information in the seismic gather (input data), we used deep-convolutional layers in the first part of the neural network to encode the information. The data gathered from the seismic-shots, essentially being time-series, we have also used an Conv-LSTM architecture to encode this feature of the data. The last layer of both the architectures is a fully-connected layer that converts the encoded information to RMS velocity (output data). This can then be compared against the true features to yield the performance of the network on the dataset.

4.2 DCNN

We have implemented a Keras [Chollet et al. [2015]] version of the DCNN neural network architecture shown in Figure 3. The loss function is taken as the mean square error between the true and the predicted velocity trace. We have used Adam optimization method with a learning rate of 0.001 for training the weights. The training is done on Amazon EC2 p2.xlarge instance with 1 GPU and 4 CPUs. Each epoch for the training process for 600 training examples take 5 seconds of computational time. Figure4 shows the training and validation error for 500 epochs. The training and validation mean square error at the end of 500 epochs are referenced in table 5. The hyper-parameter tuning based on the number of Convolution Layers and Filter Width and number of filters are performed.

4.3 DCNN with LSTM

We have implemented a Keras [Chollet et al. [2015]] version of the LSTM neural network architecture shown in Figure 3 . The loss function is taken as the mean square error between the true and the predicted velocity trace. We have used Adam optimization method with a learning rate of 0.001 for training the weights. The training is done on Amazon EC2 p2.xlarge instance with 1 GPU and 4 CPUs. Each epoch for the training process for 600 training examples take 15 seconds of computational time. Figure4 shows the training and validation error for 500 epochs. The training and validation mean square error at the end of 500 epochs are referenced in table 5. The Conv-LSTM networks are tuned against the number of convolution layers before the LSTM layer

5 Experiments/Results/Discussion

Figure 5 shows the DCNN and DCNN with LSTM's (base case) prediction for one example from the training dataset and test dataset. Overall, both the networks successfully predicted the velocity trend. However, some of the subtle fluctuations in the velocity profile were not captured in the predictions for the test dataset.

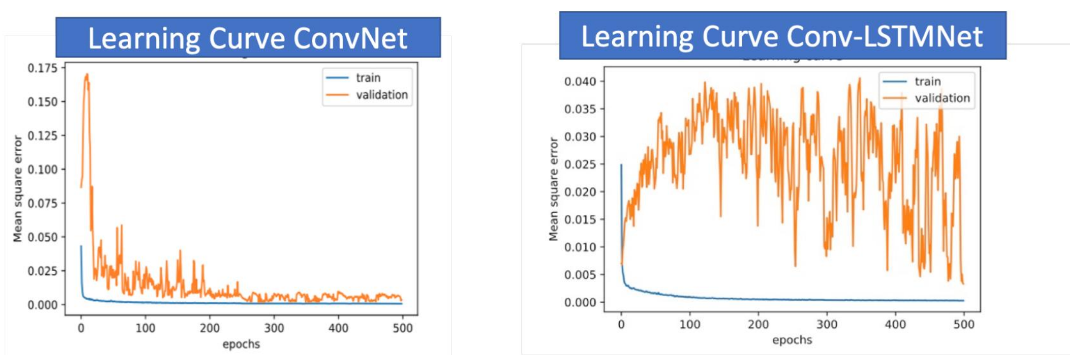

Figure 4: Learning curves for training and validation datasets for 100 epochs. The left figure shows the learning curve for the DCNN architecture and the right figure shows the learning curve for DCNN+LSTM architecture. Note the difference in y-axis range between the two figures.

6 Conclusion/Future Work

Deep neural networks showed promising results in predicting root mean square velocity from seismic gathers. DCNN with LSTM having three convolution layers, one LSTM layer and one fully-connected layer gave the least mean square error for the test dataset. However, the DCNN with LSTM architecture took five times more computation time than its DCNN equivalent. Some subtle fluctuations in the velocity predictions were not captured by the network. The deep learning approach presented in this work can help in substantially reducing the turn-around time for seismic processing in addition to improving the results.

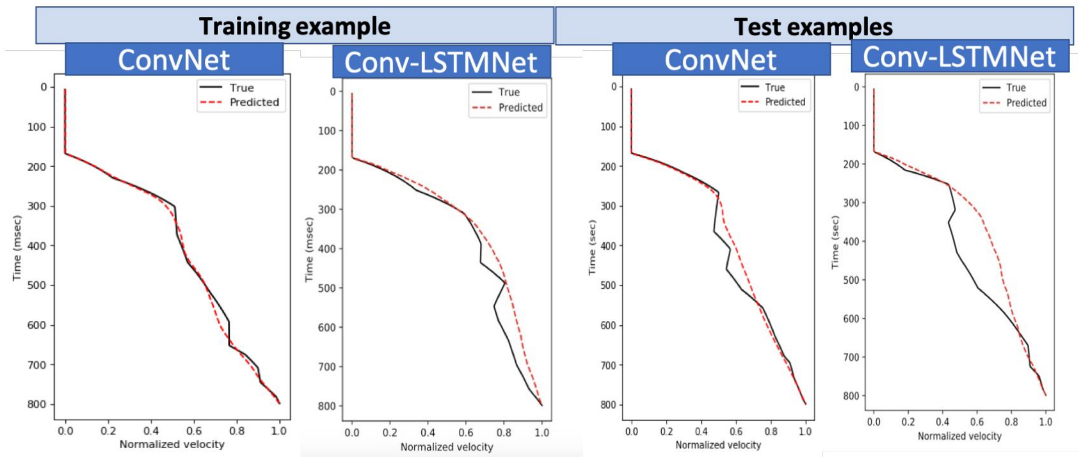

Figure 5: Examples from training and test datasets showing the true velocity and the velocity predicted from DCNN and DCNN+LSTM networks.

Sensitivity to # Layers			Sensitivity to # Filters			Sensitivity to Filter Width		
# CNN layers	Train MSE	Test MSE	# CNN Filters	Train MSE	Test MSE	Filter width	Train MSE	Test MSE
3	$6.86e-4$	0.0033	16,32,64	$6.86e-4$	0.0033	7,6,6	$6.86e-4$	0.0033
2	$3.19e-4$	0.0181	32,64,128	0.0023	0.0095	14,10,8	$6.14e-4$	0.006
1	0.0873	0.0846	8,16,32	0.0015	0.0013	5,3,3	0.0015	0.0054

Table 1: Training and validation dataset error corresponding to hyperparameters (number of layers, number of filters and filter width) tuning for the DCNN.

Sensitivity to # Layers in Conv-LSTM Network		
# CNN layers	Train MSE	Test MSE
3	0.0021	0.0002
2	0.0051	0.0003
1	0.0125	0.0133

Table 2: Training and validation dataset error corresponding to hyperparameter (number of convolution layers) tuning for the DCNN+LSTM architecture.

A more robust hyper-parameter tuning can be performed to improve the predictions and the computation time. Comparison of the deep network's predictions with velocity picks using baseline approach by professional seismic processing experts can help in better assessment of the network's performance and defining a Bayes error for the problem. As future work, we propose to use 3D data consisting of stack of 2D seismic gathers to capture the spatial information for better velocity predictions. Finally, as part of extension of the work, we plan to apply the method on real 3D seismic dataset.

7 Contributions

We would like to acknowledge Gabriel Fabien-Ouellet who thought of this idea as a postdoc with the Stanford Exploration Project. Furthermore the generous amount of Amazon Web Service credits provided by Amazon which allowed us to generate our own data/labels and train our models. Finally the open source software package Keras which made implementing deep learning models significantly less painful [Chollet et al., 2015].

All group members contributed equally to this project.

8 Code

Could not link bitbucket account in gradescope... please see the public repo @ https://bitbucket.org/stubacca1123/velocitynn

References

Ö Yilmaz. *Seismic Data Analysis*. Investigations in Geophysics. Society of Exploration Geophysicists, Jan 2001. ISBN 978-1-56080-094-1. doi: 10.1190/1.9781560801580. URL http://library.seg.org/doi/book/10.1190/1.9781560801580.

Keiiti Aki and Paul G. Richards. *Quantitative seismology*. ISBN 1891389637. URL https://books.google.com/books/about/Quantitative{_}Seismology.html?id=pWhEPgAACAAJ{&}source=kp{_}book{_}description.

Jon F Claerbout. Basic Earth Imaging. Technical report, 2010. URL http://sepwww.stanford.edu/sep/prof/bei11.2010.pdf.

Vishal Das, Ahinoam Pollack, Uri Wollner, and Tapan Mukerji. Convolutional neural network for seismic impedance inversion. In *SEG Technical Program Expanded Abstracts 2018*, pages 2071–2075. Society of Exploration Geophysicists, aug 2018. doi: 10.1190/segam2018-2994378.1. URL https://library.seg.org/doi/10.1190/segam2018-2994378.1.

Mauricio Araya-Polo, Joseph Jennings, Amir Adler, and Taylor Dahlke. Deep-learning tomography. *The Leading Edge*, 37(1): 58–66, 2018. doi: 10.1190/tle37010058.1. URL https://doi.org/10.1190/tle37010058.1.

S. Farris, M. Araya-Polo, J. Jennings, B. Clapp, and B Biondi. Tomography: a deep learning vs full-waveform inversion comparison. *EAGE Workshop on High Performance Computing*, sep 2018. ISSN 2214-4609. doi: 10.3997/2214-4609.201803073. URL http://www.earthdoc.org/publication/publicationdetails/?publication=94580.

Albert Tarantola. Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, 49(8):1259–1266, 1984. URL http://library.seg.org/doi/abs/10.1190/1.1441754.

Yue Ma, Xu Ji, Tong W Fei, and Yi Luo. Velocity picking with CNN Automatic velocity picking with convolutional neural networks. 2018. doi: 10.1190/segam2018-2987088.1. URL http://library.seg.org/.

Gabriel Fabien-Ouellet and Bernard Gloaguen, Erwan Giroux. Time-domain seismic modeling in viscoelastic media for full waveform inversion on heterogeneous computing platforms with OpenCL. *Computers nd Geoscience*, 2016.

Ricker wavelet - SEG Wiki. URL https://wiki.seg.org/index.php/Dictionary:Ricker{_}wavelet.

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