Deep learning for low frequency extrapolation of multicomponent data in elastic full waveform inversion

Hongyu Sun and Laurent Demanet

Massachusetts Institute of Technology, 77 Massachusetts Ave, Cambridge, MA 02139, hongyus@mit.edu, laurent@math.mit.edu

December, 2020

Abstract

Full waveform inversion (FWI) strongly depends on an accurate starting model to succeed. This is particularly true in the elastic regime: The cycle-skipping phenomenon is more severe in elastic FWI compared to acoustic FWI, due to the short S-wave wavelength. In this paper, we extend our work on extrapolated FWI (EFWI) by proposing to synthesize the low frequencies of multi-component elastic seismic records, and use those “artificial” low frequencies to seed the frequency sweep of elastic FWI. Our solution involves deep learning: we separately train the same convolutional neural network (CNN) on two training datasets, one with vertical components and one with horizontal components of particle velocities, to extrapolate the low frequencies of elastic data. The architecture of this CNN is designed with a large receptive field, by either large convolutional kernels or dilated convolution. Numerical examples on the Marmousi2 model show that the 2-4Hz low frequency data extrapolated from band-limited data above 4Hz provide good starting models for elastic FWI of P-wave and S-wave velocities. Additionally, we study the generalization ability of the proposed neural network over different physical models. For elastic test data, collecting the training dataset by elastic simulation shows better extrapolation accuracy than acoustic simulation, i.e., a smaller generalization gap.

1 Acknowledgments

The authors thank Total SA for support. Laurent Demanet is also supported by AFOSR grant FA9550-17-1-0316. Hongyu Sun acknowledges SEG scholarships (SEG 75th Anniversary Scholarship and Michael Bahorich/SEG Scholarship) for funding. Tensorflow (Abadi et al., 2015) and Keras (Chollet et al., 2015) are used for deep learning. Elastic FWI in this note is implemented using the open source code DENISE (https://github.com/daniel-koehn/DENISE-Black-Edition). Acoustic training datasets are simulated using Pysit (Hewett & Demanet, 2013).
2 Introduction

Full waveform inversion is well-known for its great potential to provide quantitative Earth properties of complex subsurface structures. Acoustic FWI is widely used and has been successfully applied to real seismic data. However, most seismic data have strong elastic effects (Marjanović et al., 2018). The acoustic approximation is insufficient to estimate correct reflections and introduces additional artifacts to FWI results (Plessix et al., 2013; Stopin et al., 2014). Therefore, it is desirable to develop a robust elastic FWI method for high-resolution Earth model building.

Foundational work has shown the ability of elastic FWI to retrieve realistic properties of the subsurface (Tarantola, 1986; Mora, 1987). However, it has difficulty handling real data sets. Elastic FWI is very sensitive to: accuracy of the starting model; correct estimation of density; proper definition of multi-parameter classes; and noise level (Brossier et al., 2010). The complex wave phenomena in elastic wavefields bring new challenges to FWI.

Among the many factors that affect the success of elastic FWI, the lowest starting frequency is an essential one, given that an accurate starting model is generally unavailable. Compared to acoustic FWI, the nonlinearity of elastic FWI is more severe due to the short S-wave propagating wavelength. Therefore, elastic FWI always requires a lower starting frequency compared to acoustic FWI. Additionally, the parameter cross-talk problem exists in elastic FWI and becomes more pronounced at higher frequencies, so ultra-low frequencies are required for a successful inversion of S-wave velocity and density.

In synthetic studies of elastic FWI, Brossier et al. (2009) invert the overthrust model (Aminzadeh et al., 1997) from 1.7Hz. Brossier et al. (2010) invert the Valhall model (Sirgue et al., 2009) from 2Hz. Both inversion workflows start from Gaussian smoothing of true models. Moreover, Choi et al. (2008) invert the Marmousi2 model (Martin et al., 2006) using a velocity-gradient starting model but a very low frequency (0.16Hz). For a successful inversion of the Marmousi2 density model, Köhn et al. (2012) use 0-2Hz in the first stage of multi-scale FWI. Jeong et al. (2012) invert the same model from 0.2Hz.

Few applications of elastic FWI to real data sets are reported in the literature (Crase et al., 1990; Sears et al., 2010; Marjanović et al., 2018). Vigh et al. (2014) use 3.5Hz as the starting frequency of elastic FWI given that the initial models are accurate enough. Raknes et al. (2015) apply 3D elastic FWI to update P-wave velocity and obtain S-wave velocity and density using empirical relationships. Borisov et al. (2020) perform elastic FWI involving surface waves in the band of 5-15Hz for a land data set.

New developments in acquisition enhance the recent success of FWI by measuring data with lower frequencies and longer offsets (Mahrooqi et al., 2012; Brenders et al., 2018). However, only acoustic FWI was applied to the land data set with low frequencies down to 1.5 Hz (Plessix et al., 2012). In addition to the expensive acquisition cost for the low-frequency signals, direct use of the field low-frequency data requires dedicated pre-processing steps, including travel-time tomography, for an accurate enough model to initialize FWI. The final inversion results strongly rely on the starting tomography model. Hence, attempting to retrieve reliable low-frequency data offers a sensible pathway to relieve the dependency of elastic FWI on starting models.

Deep learning is an emerging technology in many aspects of exploration geophysics. In seismic inversion, several groups have experimented with directly mapping data to model.
using deep learning (Araya-Polo et al., 2018; Yang & Ma, 2019; Wu & Lin, 2019; Zhang & Lin, 2020; Kazemi et al., 2020). Within Bayesian seismic inversion framework, deep learning has been applied for formulating priors (Herrmann et al., 2019; Mosser et al., 2020; Fang et al., 2020). Other groups use deep learning as a signal processing step to acquire reasonable data for inversion. For instance, Li et al. (2019) use deep learning to remove elastic artifacts for acoustic FWI. Siahkoohi et al. (2019) remove the numerical dispersion of wavefields by transfer learning.

Computationally extrapolating the missing low frequencies from band-limited data is the cheapest way for FWI to mitigate the cycle-skipping problem. Li & Demanet (2015, 2016) separate the shot gather to atomic events and then change the wavelet to extrapolate the low frequencies. Li & Demanet (2017) extend the frequency spectrum based on the redundancy of extended forward modeling. Recently, Sun & Demanet (2018); Ovcharenko et al. (2018); Jin et al. (2018) have utilized CNN to extrapolate the missing low frequencies from band-limited data. They have proposed different architectures of CNN to learn the mapping between high and low frequency data from different features in the training datasets. However, only acoustic data are considered in these studies.

Although the mechanism of deep learning is hard to explain, the feasibility of low frequency extrapolation has been discussed in terms of sparsity inversion (Hu et al., 2019) and wavenumber illumination (Ovcharenko et al., 2019). With multiple-trace extrapolation, the low wavenumbers of far-offset data have been proposed as the features in the frequency domain detected by CNN to extrapolate the missing low frequencies (Ovcharenko et al., 2019). In contrast, for trace-by-trace extrapolation (Sun & Demanet, 2020), the features to learn are the structured time series themselves. The feasibility of trace-by-trace frequency extrapolation has been mathematically proved in simple settings in Demanet & Nguyen (2015); Demanet & Townsend (2019), as a by-product of super-resolution.

In this paper, we extend our workflow of extrapolated FWI with deep learning (Sun & Demanet, 2020) into the elastic regime. We separately train the same neural network on two different training datasets, one to predict the low-frequency data of the horizontal components ($v_x$) and one to predict the low frequencies of the vertical components ($v_y$). The extrapolated low frequency data are used to initialize elastic FWI from a crude starting model. For the architecture design of CNN, a large receptive field is achieved by either large convolutional kernels or dilation convolution. Moreover, to investigate the generalization ability of neural networks over different physical models, we compare the extrapolation results of the neural networks trained on elastic data and acoustic data to predict the elastic low-frequency data. We also investigate several hyperparameters of deep learning to understand its bottleneck for low frequency extrapolation, such as mini-batch size, learning rate and the probability of dropout.

The organization of this article is as follows. In Section 3, we first briefly review elastic FWI and its implementation in this paper. Then, we present the architecture of neural networks and training datasets for low frequency extrapolation of elastic data. Section 4 describes the numerical results of low frequency extrapolation, extrapolated elastic FWI and investigation of hyperparameters. Section 5 discusses the limitations of the method. Section 6 comes to conclusions and future directions.
3 Method

We first give a brief review of elastic FWI as implemented in this paper. Then we illustrate the feasibility of low frequency extrapolation, and design two deep learning models for this purpose. Afterwards, the training and test datasets are provided to train and verify the performance of the proposed neural networks.

3.1 Review of elastic FWI

Elastic FWI is implemented in the time domain to invert the P-wave velocities ($v_p$), S-wave velocities ($v_s$) and density ($\rho$) simultaneously. The object function $E$ is formulated as

$$E = \frac{1}{2} \delta d^T \delta d = \frac{1}{2} \sum_s \sum_r \int [u_{cal} - u_{obs}]^2 dt,$$

(1)

where $d$ are the residuals between observed wavefields $u_{obs}$ and calculated wavefields $u_{cal}$. In 2D, both $u_{obs}$ and $u_{cal}$ contain the $v_x$ and $v_y$ components of elastic wavefields. The gradient $\frac{\delta E}{\delta m}$ relative to the model parameters $m$ is calculated in terms of $v_p$, $v_s$ and $\rho$ using the velocity-stress formulation of the elastic wave equation (Kohn et al., 2012). The starting models $m_0$ are updated using the L-BFGS method (Nocedal & Wright, 2006).

3.2 Deep learning models for low-frequency extrapolation

We choose CNN to perform the task of low-frequency extrapolation. By trace-by-trace extrapolation, the output and input are the same seismic recording in the low and high frequency band, respectively. In 2D, the elastic data contain horizontal and vertical components. As a result, we propose to separately train the same neural network twice on two different training datasets: one contains $v_x$ and the other contains $v_y$.

We design two kinds of CNN architectures with large receptive field for low-frequency extrapolation. A very large receptive field enables each feature in the final output to include a large range of input pixels. Since any single frequency component is related to the entire waveform in the time domain, extrapolation from one frequency band to the other requires a large receptive field to cover the entire input signal. Typically, the receptive field is increased by stacking layers. For example, stacking two convolutional layers (without pooling) with $3 \times 3$ filter results in a layer of $5 \times 5$ filter. However, it requires many layers to result in a large enough receptive field and is computationally inefficient. Therefore, we design the CNN architecture with two methods: directly use a large filter or a small filter with dilated convolution.

The first CNN architecture (ARCH1) directly employs a large filter on convolutional layers, which is the same as Sun & Demanet (2020) (Figure 1). Recall that ARCH1 is a feedforward stack of five convolutional blocks. Each block is a combination of a 1D convolutional layer, a PReLU layer and a batch normalization layer. The length of all filters is 200. On each convolutional layer, the channels number 64, 128, 64, 128 and 32, respectively. Although only one trace is plotted in Figure 1, the proposed neural network can easily explore multiple traces of the shot gather for multi-trace extrapolation by increasing the size of the kernels from $200 \times 1$ to $200 \times ntr$, where $ntr$ is the number of the input traces.
Figure 1: ARCH1. The first choice of deep learning architecture that directly employs a large filter on each convolutional layer (Sun & Demanet, 2020). The size and number of filters are labeled on the top of each convolutional layer.

The second CNN architecture (ARCH2) uses dilated convolution to increase the receptive field by orders of magnitude. A dilated convolution (convolution with holes) is a convolution where the filter is applied over an area larger than its length by skipping input values with a certain step (dilation). It effectively allows the network to operate on a coarser scale than with a normal convolution. This is similar to pooling or stride, but here the output has the same size as the input. As a special case, dilated convolution with dilation of one yields the standard convolution. Stacked dilated convolutions enable networks to have very large receptive fields with just a few layers. In addition to save computational cost, this method helps to preserve the input resolution throughout the network (Oord et al., 2016). Moreover, we use causal convolution to process time series (Moseley et al., 2018), although this choice does not appear to be essential in our case.

The architecture of ARCH2 (Figure 2) has two dilated convolutional blocks. Each block consists of 10 1D convolutional layers. Each convolutional layer is followed by a PReLU layer and a batch normalization layer. On each convolutional layer, there are 64 causal convolutional filters with a length of 2 (Figure 2b). The dilations of the ten convolutional layers are $2^0$, $2^1$, ..., $2^{10}$, respectively. The exponential increase in dilation results in exponential growth, with depth, of the receptive field (Yu & Koltun, 2015).

With the two proposed architectures, we can compare the specific receptive fields of both ARCH1 and ARCH2. Without pooling layer, the size of the receptive field $RF_{l+1}$ on the $l+1$ layer is

$$RF_{l+1} = RF_l + (k_{l+1} - 1) \times s_{l+1} \times d_{l+1}, \quad l = 0, ..., n,$$

$$RF_0 = 1,$$

where $k_{l+1}$ is the kernel size of the $l+1$ convolutional layer. $s_{l+1}$ is the stride size. $d_{l+1}$ is
Figure 2: (a) ARCH2. The second choice of deep learning architecture with dilated convolution. The filter length is 2 on each convolutional layer but two dilated convolutional blocks are stacked to increase the receptive field exponentially with depth. (b) The dilated causal convolutional block. Each block has 11 convolutional layers. The filter size, number of channel and dilation of each convolutional layer are labeled on the top.
the dilation on the \( l + 1 \) layer if the layer contains a dilated convolution. Otherwise, \( d_{l+1} \) equals 1 for regular convolutional layers.

The receptive field is 996 for ARCH1 and 4095 for ARCH2. Since a smaller kernel is used in ARCH2, the number of trainable parameters in ARCH2 \( (p_2=148,453,480) \) is much less than ARCH1 \( (p_1=294,602,648) \). Although both neural networks are able to perform low frequency extrapolation, convolution with dilation is more efficient than directly using a large convolutional kernel.

### 3.3 Training and test datasets

The training and test datasets are simulated on the elastic training and test models. The Marmousi2 elastic model (Figure 3) is referred to as the test model in deep learning. This is also the true model in the subsequent elastic FWI. The training models (Figure 4) are six batches randomly extracted from the Marmousi2 model. Our previous work [Sun & De- manet, 2020] has shown that the random selection of the training models on the Marmousi model provides enough generalization ability for the neural network to extrapolate low frequencies on the full-size Marmousi model. Here in the elastic regime, each model consists of three parameters: \( v_p, v_s \) and \( \rho \). The size of each model is 500 \( \times \) 174 with a grid spacing of 20m, including a water layer on the top of each model with a depth of 440m.

Both training and test datasets are simulated using a 2D time domain stress-velocity P–SV finite-difference (FD) code [Virieux, 1986; Levander, 1988] with an eighth-order spatial FD operator. A Ricker wavelet with a dominant frequency of 10Hz is used as the source signal. The sampling rate and the recording time is 0.02s and 6s, respectively. It is not necessary to collect the training and test datasets using the same acquisition geometry. To collect the test dataset, 50 shots are excited evenly from 800m to 8640m in the water layer at the same depth of 40m. 400 receivers are placed from 800m to 8780m under the water layer with a depth of 460m to record \( v_x \) and \( v_y \) of the elastic wavefields. However, for the training model, there are 100 shots excited evenly from 500m to 8420m with a depth of 40m on each training model. 400 receivers are placed from 480m to 8460m.

After the forward modeling, two training datasets are collected, one with a dataset of horizontal components and one with a dataset of vertical components. The 2D elastic data on the test model is also separated into two test datasets to process each component individually. By trace-by-trace extrapolation setup, there are \( 6 \times 100 \times 400 = 240,000 \) training samples in each training dataset and \( 1 \times 50 \times 400 = 20,000 \) test samples in each test dataset.

A simple preprocessing step can be used to improve the deep learning performance. Each sample in the training and test datasets is normalized to one by dividing the raw signal by its maximum. Then all the data are scaled with a constant (for instance, 100) to stabilize the training process. The values used to normalize and scale the raw data are recorded to recover the original observed data for elastic FWI. After this process, each sample in the training and test dataset is separated into a low-frequency signal and a high-frequency signal using a smooth window in the frequency domain. Then, each time series in the high-frequency band is fed into the neural network to predict the low-frequency time series.
Figure 3: The true Marmousi2 model: (a)$v_p$, (b)$v_s$ and (c)$\rho$. 
4 Numerical Examples

The numerical examples section is divided into four parts. In the first part, we train ARCH1 to extrapolate the low-frequency data of bandlimited multi-component recordings simulated on the Marmousi2 model (Figure 3). In the second part, we study the generalization ability of the proposed neural network over different physical models (acoustic or elastic wave equation). Then, we use the extrapolated low-frequencies of multi-component band-limited data to seed the frequency sweep of elastic FWI on the Marmousi2 model. In the last part, we investigate the hyperparameters of deep learning for low-frequency extrapolation using the proposed ARCH1 and ARCH2.

4.1 Low frequency extrapolation of multicomponent data

We first extrapolate the low frequency data below 5Hz on the Marmousi2 model (Figure 3) using 5-25Hz band-limited data. Each sample in the training and test datasets is separated into a 0-5Hz low-frequency signal and a 5-25Hz high-frequency signal using a smooth window in the frequency domain. The time series in the high-frequency band is directly fed into the neural network to predict the 0-5Hz low-frequency time series. To deal with the multicomponent data, the neural network ARCH1 is trained twice: once on the training dataset of $v_x$ and once on the training dataset of $v_y$. Both training processes use the ADAM method with a mini-batch of 32 samples. We refer readers to Sun & Demanet (2020) for more details about training. Figures 5(a) and 5(b) show the training processes over 40 epochs to predict...
Figure 5: The learning curves of ARCH1 trained to extrapolate the 0-5Hz low frequencies of (a) $v_y$ and (b) $v_x$ from the 5-25Hz band-limited elastic recordings.

the low frequencies of $v_x$ and $v_y$, respectively. The curves of training loss decay over epochs on both the training and test datasets, which indicate that the neural network does not overfit.

Figure 5 shows the extrapolation results of both $v_x$ and $v_y$ where the source is located at 7.04km. Figures 7(a) and 7(b) compare the amplitude and phase spectrum of $v_y$ and $v_x$ at $x = 6.1$km among the band-limited recording (5.0 – 25.0Hz), the fullband recording with true and predicted low frequencies (0.1 – 5.0Hz). Despite minor prediction errors on both amplitude and phase, the neural network ARCH1 can successfully recover the low frequencies of $v_x$ and $v_y$ recordings with satisfactory accuracy.

4.2 Generalization over physical models

To study the generalization ability of the proposed neural network over different physical models (acoustic wave and elastic wave), we train ARCH1 once on an acoustic training dataset and simultaneously predict the low frequencies of both $v_x$ and $v_y$ in the same elastic test dataset. The acoustic training dataset is simulated using the acoustic wave equation on only the P-wave velocity models in Figure 4. Figures 8(a) and 8(b) show the amplitude and phase spectrum of $v_y$ and $v_x$ at $x = 6.1$km after training with the same procedure. Compared with the results in Figure 7, the extrapolation accuracy of the same trace in the test data is much poorer on acoustic training dataset than elastic training dataset. Even the extrapolation of the vertical component is not successful when the neural network is trained on acoustic data. This is an indicator that the neural network has difficulty generalizing to different physical models.

4.3 Extrapolated elastic full waveform inversion

We perform extrapolated elastic FWI using 4-20Hz band-limited data on the Marmousi2 model. The lower band of the band-limited data is 4Hz. Figures 9a, 10a and 11a show the initial models of $v_p$, $v_s$ and $\rho$, respectively. Unlike in the previous examples, a free surface boundary condition is applied to the top of the model to simulate the realistic marine
Figure 6: The extrapolation results of ARCH1 for Marmousi2 model: comparison among the (a) band-limited recordings (5.0 \text{−} 25.0\text{Hz}), (b) predicted and (c) true low-frequency recordings (0.1 \text{−} 5.0\text{Hz}) of $v_y$ and (d) band-limited recordings (5.0 \text{−} 25.0\text{Hz}), (e) predicted and (f) true low-frequency recordings (0.1 \text{−} 5.0\text{Hz}) of $v_x$. 
Figure 7: Extrapolation results of ARCH1 trained on elastic data: comparison of the amplitude and phase spectrum of (a) $v_y$ and (b) $v_x$ at $x = 6.1$km among the bandlimited recording ($5.0 - 25.0$Hz), the recording ($0.1 - 25.0$Hz) with true and predicted low frequencies ($0.1 - 5.0$Hz).

Figure 8: Extrapolation results of ARCH1 trained on acoustic data: comparison of the amplitude and phase spectrum of (a) $v_y$ and (b) $v_x$ at $x = 6.1$km among the bandlimited recording ($5.0 - 25.0$Hz), the recording ($0.1 - 25.0$Hz) with true and predicted low frequencies ($0.1 - 5.0$Hz).
Figure 9: Comparison among (a) the initial $v_p$ model, the inverted low-wavenumber velocity models using (b) $2.0 - 4.0Hz$ extrapolated data and (c) $2.0 - 4.0Hz$ true data. The inversion results in (b) and (c) are started from the initial model in (a).
Figure 10: Comparison among (a) the initial $v_s$ model, the inverted low-wavenumber velocity models using (b) $2.0 - 4.0 Hz$ extrapolated data and (c) $2.0 - 4.0 Hz$ true data. The inversion results in (b) and (c) are started from the initial model in (a).
Figure 11: Comparison among (a) the initial $\rho$ model, the inverted low-wavenumber density models using (b) $2.0 - 4.0Hz$ extrapolated data and (c) $2.0 - 4.0Hz$ true data. The inversion results in (b) and (c) are started from the initial model in (a).
Figure 12: Comparison of the inverted $v_p$ models from elastic FWI using $4 - 20Hz$ band-limited data. (a) The resulting model starts from the original initial model. (b) The resulting model starts from the inverted low-wavenumber velocity model using $2.0 - 4.0Hz$ extrapolated data. (c) The resulting model starts from the inverted low-wavenumber velocity model using $2.0 - 4.0Hz$ true data.
Figure 13: Comparison of the inverted $v_s$ models from elastic FWI using $4 - 20Hz$ band-limited data. (a) The resulting model starts from the original initial model. (b) The resulting model starts from the inverted low-wavenumber velocity model using $2.0 - 4.0 Hz$ extrapolated data. (c) The resulting model starts from the inverted low-wavenumber velocity model using $2.0 - 4.0 Hz$ true data.
Figure 14: Comparison of the inverted $\rho$ models from elastic FWI using 4$-$20Hz band-limited data. (a) The resulting model starts from the original initial model. (b) The resulting model starts from the inverted low-wavenumber density model using 2.0$-$4.0Hz extrapolated data. (c) The resulting model starts from the inverted low-wavenumber density model using 2.0$-$4.0Hz true data.
exploration environment. The free surface condition damages the low frequency data and thus the energy in the low frequency band 0-2Hz is close to zero in the simulated full-band data. This brings a new challenge to the low frequency extrapolation and introduces prediction errors to the extrapolated data. For this reason, we start elastic FWI using 2-4Hz extrapolated data before exploring the band-limited data.

Starting from the crude initial model in Figure 9(a), Figures 9(b) and 9(c) show the resulting P-wave velocity models after 30 iterations using extrapolated and true 2-4Hz low-frequency data, respectively. The inverted S-wave velocity models using extrapolated and true low frequencies are shown in Figures 10(b) and 10(c). Also, Figures 11(b) and 11(c) compare the inverted density models using the extrapolated and true 2-4Hz low frequencies. The inverted low-wavenumber models of $v_p$, $v_s$ and $\rho$ using extrapolated data are roughly the same as those using true data. However, the inversion of density model is not successful since 2-4Hz data are relatively high frequencies for the inversion of density model.

Then the inversion is continued with the 4-20Hz band-limited data. We utilize a multiscale method (Bunks et al., 1995) and sequentially explore the 4-6Hz, 4-10Hz and 4-20Hz band-limited data in the elastic FWI. In each frequency band, the number of iterations is 30, 30 and 20, respectively. Figures 12 and 13 show the resulting $v_p$ and $v_s$ models started from different low wavenumber models. The inversion results of $v_p$ and $v_s$ started from 2-4Hz extrapolated data are very close to the results started from 2-4Hz true data. Conversely, elastic FWI directly starting from the crude initial models using the band-limited data shows large errors.

Figure 14 shows the resulting density models using the 4-20Hz band-limited data but started from different models in Figure 11. Since the starting frequency band (2-4Hz) is relatively high for the inversion of density on the crude initial model (Figure 11(a)), the inverted models using 2-4Hz data (Figures 11(b) and (c)) only show high-wavenumber structure of the density model. With band-limited data involved in the inversion, the inverted density models (Figure 14) resemble migration results but show the density perturbation (Mora, 1987). For a successful inversion of the density model, a much lower starting frequency band is required to recover the low-wavenumber structures.

### 4.4 Investigation of hyperparameters

The performance of deep learning is very sensitive to the hyperparameters of training. However, choosing appropriate hyperparameters requires expertise and extensive trial and error. Here we discuss the influence of several hyperparameters on the training process, including mini-batch size, learning rate, and a layer-specific hyperparameter, i.e., dropout. We also compare the performance of ARCH1 and ARCH2 in terms of training cost and prediction accuracy. In each case, we compare the model performance of the new hyperparameter setting with the results predicted by the neural network ARCH1 in Section 4.3. The neural network is trained using a mini-batch of 32, learning rate of $10^{-3}$. The probability of dropout is 50%.

The first hyperparameter is the mini-batch size. A batch is a small subset of training data randomly selected by the optimizer to calculate the gradient. Choosing a suitable mini-batch size is a trade-off between training speed and test accuracy. Fewer samples in a batch slow down the training process but a large batch increases the instability of the neural network and lead to a poor performance. Figures 15(a) and 15(b) show the learning curves.
of ARCH1 when processing the $v_y$ and $v_x$ components using a batch size of 16, 32, 64 and 128. A mini-batch of 32 gives more reasonable decrease of the training loss among others. Therefore, we choose it as the mini-batch size in our experiments.

The second essential hyperparameter is the learning rate of the optimizer. Figure 16 compares the learning curves when the learning rates equal to $10^{-2}$, $10^{-3}$ and $10^{-4}$, respectively. If the learning rate is small, training is more reliable, but it will take significant time because steps towards the minimum of the loss function are tiny. The model may also miss the important patterns in the training data; Conversely, if the learning rate is high, training may not converge. Weight changes are so large that the optimizer overshoots the minimum and makes the loss worse. We observe that a learning rate of $10^{-3}$ seems to be an optimal learning rate and can quickly and stably find the minimum loss.

Moreover, we study a model-related hyper-parameter, i.e., dropout in ARCH1. Dropout prevents neural networks from overfitting by randomly dropping neurons during training. Each neuron is retained with a fixed probability $p$ independent of other neurons. $p$ can be chosen using a validation set or can simply be set at 0.5, which seems to be close to optimal for a wide range of networks and tasks (Srivastava et al., 2014). Figure 17 compares the learning curves of ARCH1 with a dropout rate of 0% (no dropout), 20% and 50%, respectively. It seems that for this training dataset, a case without dropout layer does not hurt the performance of ARCH1. All of the three cases give reasonably right learning curves on both training and test datasets.

Finally, we compare the performance of ARCH1 and ARCH2 using the same training dataset and hyperparameter. Each network is trained twice with the training dataset of $v_x$ and the training dataset of $v_y$. According to the learning curves in Figure 17, the training of ARCH2 is much more stable than that of ARCH1. ARCH2 also requires less training time, due to the less trainable parameters compared with ARCH1. Figure 18 shows the extrapolated elastic FWI results started from the 2-4Hz extrapolated low frequency data using ARCH2. According to the comparison of the inverted models started from the extrapolated data using ARCH1 and ARCH2, both neural networks are able to provide sufficient accuracy for the inversion of $v_p$ and $v_s$. 

20
Figure 15: The learning curves of ARCH1 trained using different batch sizes to extrapolate the 0-4Hz low frequencies of (a) \(v_y\) and (b) \(v_x\) from the 4-25Hz band-limited elastic recordings.

Figure 16: The learning curves of ARCH1 trained using different learning rates to extrapolate the 0-4Hz low frequencies of (a) \(v_y\) and (b) \(v_x\) from the 4-25Hz band-limited elastic recordings.
Figure 17: The learning curves of ARCH1 and ARCH2 trained to extrapolate the 0-4Hz low frequencies of (a) $v_y$ and (b) $v_x$ from the 4-25Hz band-limited elastic recordings.

5 Discussions and Limitations

Recovering the density using FWI is very challenging, independently of the bandwidth extension question, for the following reasons. (1) Cross-talk happens using short-offset data since P-wave velocity and density have the same radiation patterns at short apertures. (2) The variations in density are smaller than those in velocities. (3) Inversion of density requires ultra-low frequencies. Although elastic FWI does not always allow to correctly estimate density, it stands a better chance of properly reconstructing velocities, with either extrapolated or true low frequencies.

In extrapolated FWI, the choice of starting frequency is a trade-off between the accuracy of extrapolated low frequency data and the lowest frequency to mitigate the cycle-skipping problem. We start elastic FWI with 2-4Hz extrapolated low frequency data due to the insufficient extrapolation accuracy in the near-zero frequency range. The accuracy of the 2-4Hz extrapolated low frequency data is sufficient for elastic FWI of P-wave and S-wave velocities when starting from 4Hz band-limited data. However, the inversion of density model still suffers from the cycle-skipping problem and lack of the low-wavenumber structure.

The numerical example shows that the neural network cannot meaningfully generalize from the acoustic training dataset to the elastic test dataset. In addition to the wave propagation driven by different physics, another factor that makes the generalization fail could be numerical modeling. The synthetic elastic training and test datasets are simulated by solving the stress-velocity formulation of the wave equation using standard staggered grid with an eighth order FD operator. However, the acoustic training dataset is simulated by
Figure 18: Comparison of the inverted (a) $v_p$, (b) $v_s$ and (c) $\rho$ models from extrapolated elastic FWI. The resulting models start from the inverted low-wavenumber models using $2.0 - 4.0Hz$ data extrapolated using ARCH2.
solving the stress-displacement formulation using a sixth order FD operator.

The source signal is assumed to be known for extrapolated elastic FWI in this paper. However, for field data, the source signal may vary shot by shot. One solution could be to retrieve the source wavelet of the field dataset firstly, and then artificially boost the low-frequency energy after denoising. The new source signal can be used to synthesize the training dataset for low-frequency extrapolation. It can also be the source wavelet in the following elastic FWI using the extrapolated low-frequency data. In this way, the uncertainty of the source can be controlled to some extent.

We do not provide the numerical results of other factors that affect the performance of the deep learning models. For example, regularization of training loss, number of iterations, parameters of neural network, number of training samples and even inverse crime. Deep learning models contain many hyper-parameters and finding the best configuration for these parameters in a high dimensional space is challenging.

Finally, neural networks are trained using a stochastic learning algorithm. This means that the same model trained on the same dataset may result in a different performance. The specific results may vary, but the general trend should be the same, as reported in Section 4.4.

6 Conclusions

To relieve the dependency of elastic FWI on starting models, low-frequency extrapolation of multi-component seismic recordings is implemented to computationally recover the missing low frequencies from band-limited elastic data. The deep learning model is designed with a large receptive field in two different ways. One directly uses a large filter on each convolutional layer, the other utilizes dilated convolution to increase the receptive field exponentially with depth. By training the neural network twice, once with a dataset of horizontal components and once with a dataset of vertical components, we can extrapolate the low frequencies of multi-component band-limited recordings separately. The extrapolated 0-5Hz low frequencies match well with the true low-frequency data on the Marmousi2 model. Elastic FWI using 2-4Hz extrapolated data shows similar results to the true low frequencies. The accuracy of the extrapolated low frequencies is enough to provide low-wavenumber starting models for elastic FWI of P-wave and S-wave velocities on data band-limited above 4Hz.

The generalization ability of the neural network over different physical models is studied in this paper. The neural network trained on purely acoustic data shows larger prediction error on elastic test dataset compared to the neural network trained on elastic data. Therefore, collecting more realistic elastic training dataset will help to process the field data with strong elastic effects.
References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems, Software available from tensorflow.org.

Aminzadeh, F., Jean, B., & Kunz, T., 1997. SEG/EAGE 3-D Salt and Overthrust Models. SEG/EAGE 3-D Modeling Series, No. 1: Distribution CD of Salt and Overthrust models, SEG book series.

Araya-Polo, M., Jennings, J., Adler, A., & Dahlke, T., 2018. Deep-learning tomography, The Leading Edge, 37(1), 58–66.

Borisov, D., Gao, F., Williamson, P., & Tromp, J., 2020. Application of 2D full-waveform inversion on exploration land data, Geophysics, 85(2), R75–R86.

Brenders, A., Dellinger, J., Kanu, C., Li, Q., & Michell, S., 2018. The wolfspar® field trial: Results from a low-frequency seismic survey designed for FWI, in in 88th Annual SEG meeting, SEG Technical Program Expanded Abstracts, 1083–1087.

Brossier, R., Operto, S., & Virieux, J., 2009. Seismic imaging of complex onshore structures by 2D elastic frequency-domain full-waveform inversion, Geophysics, 74(6), WCC105–WCC118.

Brossier, R., Operto, S., & Virieux, J., 2010. Which data residual norm for robust elastic frequency-domain full waveform inversion?, Geophysics, 75(3), R37–R46.

Bunks, C., Saleck, F. M., Zaleski, S., & Chavent, G., 1995. Multiscale seismic waveform inversion, Geophysics, 60(5), 1457–1473.

Choi, Y., Min, D.-J., & Shin, C., 2008. Frequency-domain elastic full waveform inversion using the new pseudo-hessian matrix: Experience of elastic Marmousi-2 synthetic data, Bulletin of the Seismological Society of America, 98(5), 2402–2415.

Chollet, F. et al., 2015. Keras.

Crase, E., Pica, A., Noble, M., McDonald, J., & Tarantola, A., 1990. Robust elastic nonlinear waveform inversion: Application to real data, Geophysics, 55(5), 527–538.

Demanet, L. & Nguyen, N., 2015. The recoverability limit for superresolution via sparsity, arXiv preprint arXiv:1502.01385.

Demanet, L. & Townsend, A., 2019. Stable extrapolation of analytic functions, Foundations of Computational Mathematics, 19(2), 297–331.
Fang, Z., Fang, H., & Demanet, L., 2020. Deep generator priors for bayesian seismic inversion, *Advances in Geophysics*, 61, 179–216.

Herrmann, F. J., Siahkoohi, A., & Rizzuti, G., 2019. Learned imaging with constraints and uncertainty quantification, *arXiv preprint arXiv:1909.06473*.

Hewett, R. & Demanet, L., 2013. the pysit team, 2013, *PySIT: Python seismic imaging toolbox v0.5*.

Hu, W., Jin, Y., Wu, X., & Chen, J., 2019. Progressive transfer learning for low frequency data prediction in full waveform inversion, *arXiv preprint arXiv:1912.09944*.

Jeong, W., Lee, H.-Y., & Min, D.-J., 2012. Full waveform inversion strategy for density in the frequency domain, *Geophysical Journal International*, 188(3), 1221–1242.

Jin, Y., Hu, W., Wu, X., Chen, J., et al., 2018. Learn low wavenumber information in FWI via deep inception based convolutional networks, in *88th Annual SEG meeting*, SEG Technical Program Expanded Abstracts, 2091–2095.

Kazei, V., Ovcharenko, O., Plotnitskii, P., Peter, D., Alkhalifah, T., Silvestrov, I., Bakulin, A., & Zwartjes, P., 2020. Elastic near-surface model estimation from full waveforms by deep learning, in *90th Annual SEG meeting*, SEG Technical Program Expanded Abstracts, 3872–3876.

Köhn, D., De Nil, D., Kurzmann, A., Przebindowska, A., & Bohlen, T., 2012. On the influence of model parametrization in elastic full waveform tomography, *Geophysical Journal International*, 191(1), 325–345.

Levander, A. R., 1988. Fourth-order finite-difference P-SV seismograms, *Geophysics*, 53(11), 1425–1436.

Li, D., Gao, F., & Williamson, P., 2019. A deep learning approach for acoustic FWI with elastic data, in *89th Annual SEG meeting*, SEG Technical Program Expanded Abstracts, 2303–2307.

Li, Y. & Demanet, L., 2017. Extrapolated full-waveform inversion: An image-space approach, in *87th Annual SEG meeting*, SEG Technical Program Expanded Abstracts, 1682–1686.

Li, Y. E. & Demanet, L., 2015. Phase and amplitude tracking for seismic event separation, *Geophysics*, 80(6), WD59-WD72.

Li, Y. E. & Demanet, L., 2016. Full-waveform inversion with extrapolated low-frequency data, *Geophysics*, 81(6), R339–R348.

Mahrooqi, S., Rawahi, S., Yarubi, S., Abri, S., Yahyai, A., Jahdhami, M., Hunt, K., & Shorter, J., 2012. Land seismic low frequencies: acquisition, processing and full wave inversion of 1.5-86 Hz, in *82nd Annual SEG meeting*, SEG Technical Program Expanded Abstracts, 1–5.
Marjanović, M., Plessix, R.-É., Stopin, A., & Singh, S. C., 2018. Elastic versus acoustic 3-D full waveform inversion at the east pacific rise 9° 50’ N, *Geophysical Journal International*, **216**(3), 1497–1506.

Martin, G. S., Wiley, R., & Marfurt, K. J., 2006. Marmousi2: An elastic upgrade for marmousi, *The Leading Edge*, **25**(2), 156–166.

Mora, P., 1987. Nonlinear two-dimensional elastic inversion of multioffset seismic data, *Geophysics*, **52**(9), 1211–1228.

Moseley, B., Markham, A., & Nissen-Meyer, T., 2018. Fast approximate simulation of seismic waves with deep learning, *arXiv preprint arXiv:1807.06873*.

Mosser, L., Dubrule, O., & Blunt, M. J., 2020. Stochastic seismic waveform inversion using generative adversarial networks as a geological prior, *Mathematical Geosciences*, **52**(1), 53–79.

Nocedal, J. & Wright, S., 2006. *Numerical optimization*, Springer Science & Business Media.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K., 2016. Wavenet: A generative model for raw audio, *arXiv preprint arXiv:1609.03499*.

Ovcharenko, O., Kazei, V., Peter, D., Zhang, X., & Alkhalifah, T., 2018. Low-frequency data extrapolation using a feed-forward ANN, in *80th EAGE Conference and Exhibition 2018*, European Association of Geoscientists & Engineers, 1–5.

Ovcharenko, O., Kazei, V., Kalita, M., Peter, D., & Alkhalifah, T., 2019. Deep learning for low-frequency extrapolation from multioffset seismic data, *Geophysics*, **84**(6), R989–R1001.

Plessix, R., Milcik, P., Rynja, H., Stopin, A., Matson, K., & Abri, S., 2013. Multiparameter full-waveform inversion: Marine and land examples, *The Leading Edge*, **32**(9), 1030–1038.

Plessix, R.-É., Baeten, G., de Maag, J. W., ten Kroode, F., & Ruije, Z., 2012. Full waveform inversion and distance separated simultaneous sweeping: a study with a land seismic data set, *Geophysical Prospecting*, **60**(4), 733–747.

Raknes, E. B., Arntsen, B., & Weibull, W., 2015. Three-dimensional elastic full waveform inversion using seismic data from the sleipner area, *Geophysical Journal International*, **202**(3), 1877–1894.

Sears, T. J., Barton, P. J., & Singh, S. C., 2010. Elastic full waveform inversion of multi-component ocean-bottom cable seismic data: Application to Alba Field, UK North Sea, *Geophysics*, **75**(6), R109–R119.

Siahkoohi, A., Louboutin, M., & Herrmann, F. J., 2019. The importance of transfer learning in seismic modeling and imaging, *Geophysics*, **84**(6), A47–A52.
Sirgue, L., Barkved, O., Van Gestel, J., Askim, O., & Kommedal, J., 2009. 3D waveform inversion on Valhall wide-azimuth OBC, in 71st EAGE Conference and Exhibition incorporating SPE EUROPEC 2009, European Association of Geoscientists & Engineers, pp. cp–127.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research, 15(1), 1929–1958.

Stopin, A., Plessix, R.-É., & Al Abri, S., 2014. Multiparameter waveform inversion of a large wide-azimuth low-frequency land data set in Oman, Geophysics, 79(3), WA69–WA77.

Sun, H. & Demanet, L., 2018. Low frequency extrapolation with deep learning, in in 88th Annual SEG meeting, SEG Technical Program Expanded Abstracts, 2011–2015.

Sun, H. & Demanet, L., 2020. Extrapolated full-waveform inversion with deep learning, Geophysics, 85(3), R275–R288.

Tarantola, A., 1986. A strategy for nonlinear elastic inversion of seismic reflection data, Geophysics, 51(10), 1893–1903.

Vigh, D., Jiao, K., Watts, D., & Sun, D., 2014. Elastic full-waveform inversion application using multicomponent measurements of seismic data collection, Geophysics, 79(2), R63–R77.

Virieux, J., 1986. P-SV wave propagation in heterogeneous media; velocity-stress finite-difference method, Geophysics, 51(4), 889–901.

Wu, Y. & Lin, Y., 2019. InversionNet: An efficient and accurate data-driven full waveform inversion, IEEE Transactions on Computational Imaging, 6, 419–433.

Yang, F. & Ma, J., 2019. Deep-learning inversion: A next-generation seismic velocity model building method, Geophysics, 84(4), R583–R599.

Yu, F. & Koltun, V., 2015. Multi-scale context aggregation by dilated convolutions, arXiv preprint arXiv:1511.07122.

Zhang, Z. & Lin, Y., 2020. Data-driven seismic waveform inversion: A study on the robustness and generalization, IEEE Transactions on Geoscience and Remote Sensing, 58(10), 6900–6913