Low frequency extrapolation with deep learning

Hongyu Sun and Laurent Demanet, Massachusetts Institute of Technology*

SUMMARY

The lack of the low frequency information and good initial model can seriously affect the success of full waveform inversion (FWI) due to the inherent cycle skipping problem. Reasonable and reliable low frequency extrapolation is in principle the most direct way to solve this problem. In this paper, we propose a deep-learning-based bandwidth extension method by considering low frequency extrapolation as a regression problem. The Deep Neural Networks (DNNs) are trained to automatically extrapolate the low frequencies without preprocessing steps. The band-limited recordings are the inputs of the DNNs and, in our numerical experiments, the pretrained neural networks can predict the continuous-valued seismograms in the unobserved low frequency band. For the numerical experiments considered here, it is possible to find the amplitude and phase correlations among different frequency components by training the DNNs with enough data samples, and extrapolate the low frequencies from the band-limited seismic records trace by trace. The synthetic example shows that our approach is not subject to the structural limitations of other methods to bandwidth extension, and seems to offer a tantalizing solution to the problem of properly initializing FWI.

INTRODUCTION

It is recognized that the low frequency data are essential for FWI since the low wavenumber components are needed for FWI to avoid convergence to a local minimum, in case the initial models miss the reasonable representation of the complex structure. However, because of the acquisition limitation and low-cut filters in seismic processing, the input data for seismic inversion are typically limited to a band above 3Hz. With assumptions and approximations to make inference from tractable but simplified models, geophysicists have started estimating the low wavenumber components from the band-limited records by signal processing methods. For example, they recover the low frequencies by the envelope of the signal (Wu et al., 2014; Hu et al., 2017) or the inversion of the reflectivity series and convolution with the broadband source wavelet (Wang and Herrmann, 2016; Zhang et al., 2017). However, the low frequencies recovered by these methods are still far away from the true low frequency data and can only be used during the construction of the initial model for FWI. Li and Demanet (2016) attempt to extrapolate the true low frequency data based on phase tracking method (Li and Demanet, 2015). Unlike the explicit parameterization of phases and amplitudes of atomic events, here we propose an approach that deals with the raw band-limited records. The deep convolutional neural networks (CNNs) are trained to automatically recover the missing low frequencies from the input band-limited data.

Because of the state-of-the-art performance of machine learning in many fields, geophysicists have begun borrowing such ideas in seismic processing and interpretation (Chen et al., 2017; Guitton et al., 2017). Machine learning techniques attempt to leverage the concept of statistical learning associated with different types of data characteristics. Lewis and Vigh (2017) investigate convolutional neural networks (CNNs) to incorporate the long wavelength features of the model in the regularization term, by learning the probability of salt geobodies being present at any location in a seismic image. Araya-Polo et al. (2018) directly produce layered velocity models from shot gathers with DNNs. Richardson (2018) constructs FWI as recurrent neural networks.

In the case of bandwidth extension, the data characteristics are the amplitudes and phases of seismic waves, which are dictated by the physics of wave propagation. Among many kinds of machine learning algorithms, we have selected DNNs for low frequency extrapolation due to the increasing community agreement (Grzeszczuk et al., 1998; De et al., 2011; Araya-Polo et al., 2017) in favor of this method as a reasonable surrogate for physics-based process. The universal approximation theorem also shows that the neural networks can be used to replicate any function up to our desired accuracy if the DNNs have enough hidden layers and nodes (Hornik et al., 1989). Although training is therefore expected to succeed arbitrarily well, only empirical evidence currently exists for the performance of testing a network out of sample.

In this paper, we choose to focus on CNNs. The idea behind CNNs is to mine the hidden correlations among different frequency components. The raw band-limited signals in the time domain are directly fed into the CNNs for regression and bandwidth extension. The limitations of neural networks for such signal processing tasks, however, are (1) the lack of generalizability guarantees and (2) the absence of a physical interpretation for the operations performed by the networks. Even so, the preliminary results shown here for the synthetic dataset demonstrate a new direct method that attempts to extrapolate the true values of the low frequencies rather than simply estimating and compensating the low frequency energy.

THEORY AND METHOD

A neural network defines a mapping $\mathbf{y} = f(\mathbf{x}, \mathbf{w})$ and learns the value of the parameters $\mathbf{w}$ that result in a good fit between $\mathbf{x}$ and $\mathbf{y}$. DNNs are typically represented by composing together many different functions to find complex nonlinear relationships. The chain structures are the most common structures in DNNs (Goodfellow et al., 2016):

$$\mathbf{y} = f(\mathbf{x}) = f_L(...f_2(f_1(\mathbf{x}))), \tag{1}$$

where f_1, f_2 and f_L are the first, the second and the L_{th} layer of the network. The overall length of the chain L gives the depth of the deep learning model. The final layer is the output layer, which defines the size and type of the output data. The training sets specify directly what the output layer must do at each point **x** but not specify the behavior of other layers. They are hidden layers and computed by activation functions. The nonlinearity of the activation function enables the neural network to be a universal function approximator. Rectified activation units are essential for the recent success of DNNs because they can accelerate convergence of the training procedure. Numerical experiment shows that, for bandwidth extension, Parametric Rectified Linear Unit (PReLU)(He et al., 2015) works better than the Rectified Linear Unit (ReLU). The formula of PReLU is

$$g(\alpha,\mathbf{y}) = \begin{cases} \alpha\mathbf{y}, & if \quad \mathbf{y} < 0 \\ \mathbf{y}, & if \quad \mathbf{y} \geq 0 \end{cases}, \tag{2}$$

where α is also a learnable parameter and would be adaptively updated for each rectifier during training.

Unlike the classification problem that trains the DNNs to produce a probability distribution, regression problem trains the DNNs for the continuous-valued output. It evaluates the performance of the model by calculating the mean-squared error (MSE) of the predicted outputs $f(\mathbf{x}_i;\mathbf{w})$ and the actual outputs $\mathbf{y}_i$:

$$J(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m} L(\mathbf{y}_i, f(\mathbf{x}_i;\mathbf{w})), \tag{3}$$

where the loss L is the squared error between the true low frequencies and the estimated outputs of the neural networks. The cost function J is usually minimized over $\mathbf{w}$ by stochastic gradient descent (SGD) algorithm using a subset of the training set. This subset is called a mini-batch. Each evaluation of the gradient using the mini-batch is an iteration. The full pass of the training algorithm over the entire training set using mini-batches is an epoch. The learning rate η (step-size) is a key parameter for deep learning and required to be fine-tune. Adaptive moment estimation (Kingma and Ba, 2014) is one of the state-of-the-art SGD algorithms which can adapt the learning rate for each parameter by dividing the learning rate for a weight by a moving average for that weight. Both of the gradients and the second moments of the gradients are used to calculate the moving average.

$$\begin{aligned}
\mathbf{w}^{t+1} &= \mathbf{w}^{t} - \eta\frac{\hat{\mathbf{m}}_w}{\sqrt{\hat{\mathbf{v}}_w}+\varepsilon}, \\
\hat{\mathbf{m}}_w &= \frac{\mathbf{m}_w^{t+1}}{1-\beta_1^t}, \\
\hat{\mathbf{v}}_w &= \frac{\mathbf{v}_w^{t+1}}{1-\beta_2^t}, \\
\mathbf{v}_w^{t+1} &= \beta_2^t\mathbf{v}_w^t + (1-\beta_2^t)(\frac{\partial J(\mathbf{w}^t)}{\partial \mathbf{w}})^2, \\
\mathbf{m}_w^{t+1} &= \beta_1^t\mathbf{m}_w^t + (1-\beta_1^t)\frac{\partial J(\mathbf{w}^t)}{\partial \mathbf{w}},
\end{aligned} \tag{4}$$

where β_1, β_2 are the forgetting factors for gradients and second moments of gradients, respectively. They control the decay rates of the exponential moving averages. ε is a small number used to prevent division by zero. The gradients $\frac{\partial J(\mathbf{w}^t)}{\partial \mathbf{w}}$ of the neural networks are calculated by the backpropagation method (Goodfellow et al., 2016)

One typical architecture of DNNs that uses the convolution to extract spatial features is CNNs. CNNs characterized by local connections and weight sharing can exploit the local correlation of the input image. The hidden units are connected to a locally limited subset of units in the input, which is the receptive field of the filter. The size of the receptive field increases as we stack multiple convolutional layers, so the CNNs can also learn the global features. The CNNs are normally designed to deal with image classification problems. For bandwidth extension, the data to be learned are the one-dimensional time-domain seismic signals, so we directly consider the amplitude at each sampling point as the pixel value of the image to be fed into the CNNs. The basic construction of CNNs in this paper is the convolutional layer with N filters of size $n \times 1$ followed by a batch normalization layer and a PReLU layer. The filter number in each convolutional layer determines the number of the feature map or the channel of its output. Each output channel of the convolutional layer is obtained by convolving the channel of the previous layer with one filter, summing, adding a bias term. The batch normalization layer can speed up training of CNNs and reduce the sensitivity to network initialization by normalizing each input channel across a mini-batch. Although the pooling layer is typically used in the conventional architecture of CNNs, we leave it out because both the input and output signals have the same length, so downsampling is unhelpful for bandwidth extension in our experiments.

Since CNNs belong to supervised learning methods, we need to firstly train the CNNs from a large number of samples to determine the coefficients of the network, and, secondly, use the network for testing. According to the statistical learning theory, the generalization error is the difference between the expected and empirical error. It can be approximately measured by the difference between the errors on the training and test sets. For the purpose of generalization, the models to create the large training sets should be able to represent many subsurface structures, including different types of reflectors and diffractors, so we can find a common set of parameters for data from a specific region. The performance of the neural networks is sensitive to the architecture and hyperparameters, so we should design them carefully. Next, we illustrate the specific choice of the architecture and hyperparameters for bandwidth extension along with the numerical example.

NUMERICAL EXAMPLE

We demonstrate the reliability of the low frequency extrapolation with deep learning method on the Marmousi model (Figure 1). With the synthetic data, we can evaluate the extrapolation accuracy by the comparison with the true low frequencies. The full-size model is unseen during the training process and used to synthesize the test set. To collect the training set, we randomly select nine parts of the Marmousi model (Figure 2) with different size and structure, and then interpolate the submodels to the same size as the original model. In this way, the depth and distance of each velocity model are the same. We believe that the randomized models produced in this manner are realistic enough to demonstrate the generalization of the neural network if the number of submodels is large enough, so the pretrained network can be exposed to the new data collected on the new model (full-size Marmousi model) with a certain generalization level.

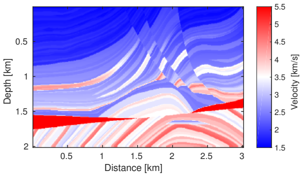

Figure 1: The Marmousi velocity model used to collect the test dataset (unseen during the training process).

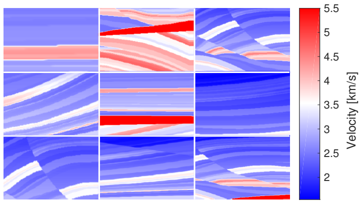

Figure 2: The nine submodels extracted from the Marmousi model to collect the training dataset.

The acquisition geometry has 30 sources and 300 receivers evenly spaced on the surface. We use the finite-difference modelling method with PML to solve the 2D acoustic wave equation in the time domain to generate the full-bandwidth wavefields of both the training and test datasets. The Ricker wavelet's dominant frequency is 20Hz and its maximum amplitude is one. The sampling interval and the total recording time are 1ms and 2.9s, respectively. Each time series or trace is considered as one sample in the dataset, so we have $81,000$ training samples and $9,000$ test samples. For each sample, we use the data in the band above 5Hz as the inputs and the data in the low frequency band (0.3-5Hz) as the outputs of the neural network.

The architecture of our neural network is a feed-forward stack of five sequential combinations of the convolution, batch normalization and PReLU layers, and finally followed by one fully connected layer which outputs continuous-valued amplitude of the time-domain signal in the low frequency band. The filter numbers of the five convolutional layers are 128, 64, 128, 64 and 1, respectively. We use only one filter in the last convolutional layer to reduce the number of channel to one. The variation of the channel number can add nonlinearity to our model. The filter size of all the filers in our neural networks are 80×1. Unlike the small filer size commonly used in image classification problem, it is essential for bandwidth extension to use large filer. The large filter size enables CNNs to have enough feasibility to learn the ability of reconstructing the long-wavelength information from the mapping between the band-limited data and their true low frequencies. The stride of the convolution is one and the zero padding is used to make the output length of each convolution layer the same as its input. The initial value of the bias is zero. The weight initialization is via the Glorot uniform initializer (Glorot and Bengio, 2010). It randomly initializes the weights from a truncated normal distribution centered on zero with the standard deviation $\sqrt{2/n_1+n_2}$ where n_1 are n_2 are the numbers of input and output units in the weight tensor, respectively. With this architecture, we train the network with the Adam optimizer and use a mini-batch of 20 samples at each iteration. The initial learning rate and forgetting rate of the Adam are the same as the original paper (Kingma and Ba, 2014).

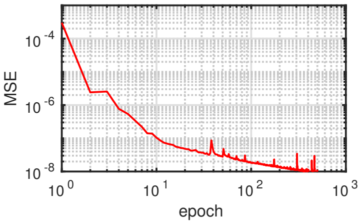

Figure 3: The training error (MSE) on the Marmousi training dataset with the proposed neural network.

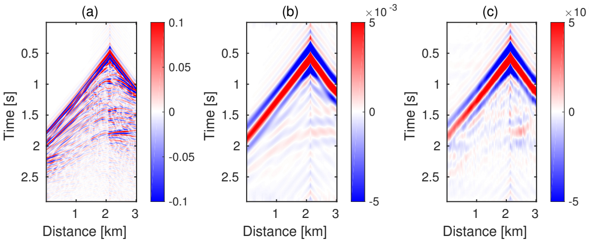

Figure 4: Comparison between the (a) band-limited recordings $(5-50Hz)$, (b) true and (c) predicted low frequency recordings $(0.3-5Hz)$. The band-limited data in (a) are the inputs of CNNs to predict the low frequencies in (b).

The training process of the 500 epochs is shown in Figure 3. After training, we test the performance of the neural networks by feeding the band-limited data in the test set into the model and obtain the extrapolated low frequencies of the full-size Marmousi model. Figure 4 compares the shot gathers between the band-limited data $(5-50Hz)$, true and extrapolated low frequencies $(0.3-5Hz)$ where the source is located at the horizontal distance $x = 2.2km$. The extrapolated results in Figure 4(c) show that the neural networks accurately predict the recordings in the low frequency band, which are totally missing before the test. Figure 5 compares two individual seismograms where the receivers are located at the horizontal distance $x = 1.73km$ and $x = 2.25km$, respectively. The extrapolated low frequency data match the true recordings well. Then we combine the extrapolated low frequencies with the band-limited data and compare the amplitude spectrum in the frequency band $0.3-20Hz$ between the data without low frequencies, with true low frequencies and with extrapolated low frequencies in Figure 6. The pretrained neural networks successfully recover the low frequency information from the band-limited data in Figure 6(a). The amplitude spectrum comparison of the single trace where the receiver is located at $x = 2.25km$ (Figure 7) clearly shows that the neural networks reconstruct the true low frequency energy very well.

Although our method is not based on any physical model, some limitations can still deteriorate the extrapolation accuracy. The most important limitation is the inevitable generalization error. As a data-driven statistical optimization method, deep learning requires a large number of samples (usually millions) to become an effective predictor. Since the training dataset in this example is small (81,000 samples) but the model capacity is large (3,290,946 trainable parameters after downsampling the signals by factor of three), it is very easy for the neural network to be overfitting, which seriously constrains the extrapolation accuracy. Therefore, in practice, it is standard to use regularization, dropout or even collect larger training set to relieve this problem. In addition, the training time of deep learning is highly related to the size of dataset and the model capacity, and thus is very demanding. For instance, the training process in this example takes one day on eight GPUs for the 500 epochs. To speed up the training by reducing the number of weights of neural networks, we can downsample both the inputs and outputs, and then use band-limited interpolation method to recover the signal after extrapolation. Another limitation in deep learning is due to the unbalanced data. The energy of the direct wave is very strong compared with the reflected waves, which biases the neural networks towards fitting the direct wave and having less contribution to the reflected waves. So the extrapolation accuracy of the reflected waves is not as good as the primary wave in this example. Moreover, as we perform bandwidth extension trace by trace, the accumulation of the predicted errors reduce the coherence of the event across traces. Hence, it is probably better to extrapolate multi-trace seismograms simultaneously. Finally, the effects of the architecture and hyperparameters of neural networks on the performance of bandwidth extension still need to be studied in detail, and thus we can further improve the extrapolation accuracy by exploring DNNs that are more suitable.

CONCLUSIONS

In this paper, we have applied deep learning method to the challenging bandwidth extension problem that is essential for FWI. We formulate bandwidth extension as a regression problem in machine learning and propose an end-to-end trainable model for low frequency extrapolation. Without any preprocessing on the input (the band-limited data) and postprocessing on the output (the extrapolated low frequencies), DNNs have the ability to recover the low frequencies, which are totally missing in the seismic data in our experiments. The choice of the architectural parameters is non-unique. The extrapolation accuracy can be further modified by adjusting the architecture and hyperparameters of the neural networks.

ACKNOWLEDGMENTS

The authors thank Total SA for support. LD is also supported by AFOSR grant FA9550-17-1-0316, and NSF grant DMS-1255203.

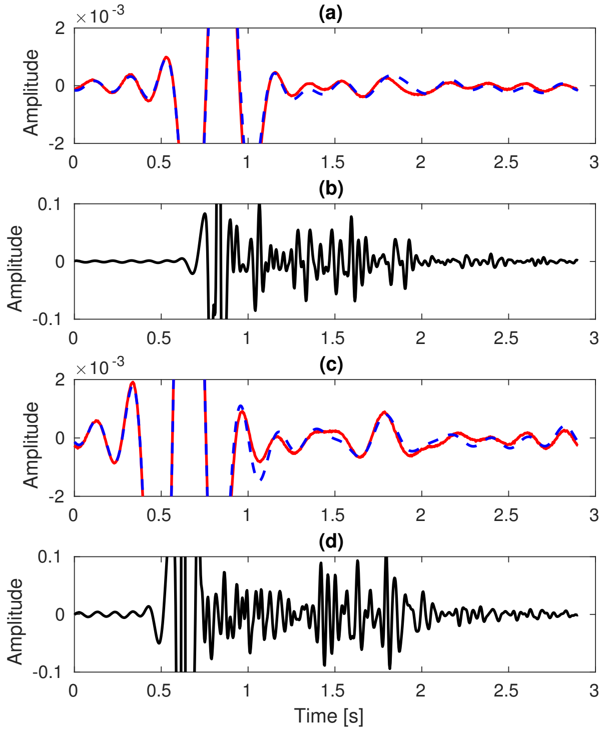

Figure 5: Comparison between the predicted (red line), the true (blue dash line) recording in the low frequency band $(0.3 - 5Hz)$ and the band-limited recording (black line) $(5 - 50Hz)$ at the horizontal distance (a) (b) $x = 1.73km$ and (c) (d) $x = 2.25km$.

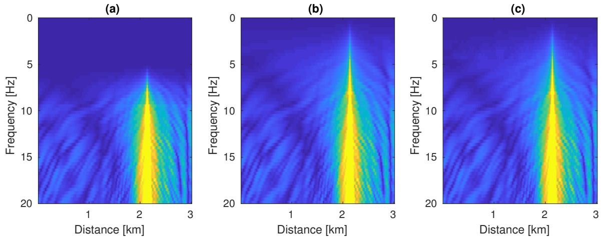

Figure 6: Comparison of the amplitude spectrum between (a) the band-limited recordings $(5 - 20Hz)$, the recordings $(0.3 - 20Hz)$ with (b) true and (c) predicted low frequencies $(0.3 - 5Hz)$.

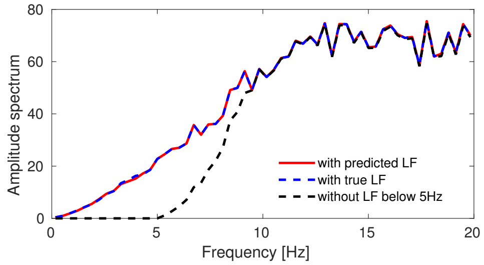

Figure 7: Comparison of the amplitude spectrum at $x = 2.25km$ between the band-limited recording $(5 - 20Hz)$, the recording $(0.3 - 20Hz)$ with true and predicted low frequencies $(0.3 - 5Hz)$.

REFERENCES

Araya-Polo, M., T. Dahlke, C. Frogner, C. Zhang, T. Poggio, and D. Hohl, 2017, Automated fault detection without seismic processing: The Leading Edge, **36**, 208–214.

Araya-Polo, M., J. Jennings, A. Adler, and T. Dahlke, 2018, Deep-learning tomography: The Leading Edge, **37**, 58–66.

Chen, Y., J. Hill, W. Lei, M. Lefebvre, J. Tromp, E. Bozdag, and D. Komatitsch, 2017, Automated time-window selection based on machine learning for full-waveform inversion: Society of Exploration Geophysicists.

De, S., D. Deo, G. Sankaranarayanan, and V. S. Arikatla, 2011, A physics-driven neural networks-based simulation system (phyn-ness) for multimodal interactive virtual environments involving nonlinear deformable objects: Presence, **20**, 289–308.

Glorot, X., and Y. Bengio, 2010, Understanding the difficulty of training deep feedforward neural networks: Proceedings of the thirteenth international conference on artificial intelligence and statistics, 249–256.

Goodfellow, I., Y. Bengio, and A. Courville, 2016, Deep learning, **1**.

Grzeszczuk, R., D. Terzopoulos, and G. Hinton, 1998, Neuroanimator: Fast neural network emulation and control of physics-based models: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, ACM, 9–20.

Guitton, A., H. Wang, and W. Trainor-Guitton, 2017, Statistical imaging of faults in 3d seismic volumes using a machine learning approach: Society of Exploration Geophysicists.

He, K., X. Zhang, S. Ren, and J. Sun, 2015, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification: Proceedings of the IEEE international conference on computer vision, 1026–1034.

Hornik, K., M. Stinchcombe, and H. White, 1989, Multilayer feedforward networks are universal approximators: Elsevier, **2**.

Hu, Y., L. Han, Z. Xu, F. Zhang, and J. Zeng, 2017, Adaptive multi-step full waveform inversion based on waveform mode decomposition: Elsevier, **139**.

Kingma, D. P., and J. Ba, 2014, Adam: A method for stochastic optimization: arXiv preprint arXiv:1412.6980.

Lewis, W., and D. Vigh, 2017, Deep learning prior models from seismic images for full-waveform inversion: Society of Exploration Geophysicists.

Li, Y. E., and L. Demanet, 2015, Phase and amplitude tracking for seismic event separation: Society of Exploration Geophysicists, **80**.

———, 2016, Full-waveform inversion with extrapolated low-frequency data: Society of Exploration Geophysicists, **81**.

Richardson, A., 2018, Seismic full-waveform inversion using deep learning tools and techniques: arXiv preprint arXiv:1801.07232.

Wang, R., and F. Herrmann, 2016, Frequency down extrapolation with tv norm minimization: Society of Exploration Geophysicists.

Wu, R.-S., J. Luo, and B. Wu, 2014, Seismic envelope inversion and modulation signal model: Society of Exploration Geophysicists, **79**.

Zhang, P., L. Han, Z. Xu, F. Zhang, and Y. Wei, 2017, Sparse blind deconvolution based low-frequency seismic data reconstruction for multiscale full waveform inversion: Elsevier, **139**.