Th_P01_03

Realistically Textured Random Velocity Models for Deep Learning Applications

V. Kazei[1]*, O. Ovcharenko[1], T. Alkhalifah[1], F. Simons[2]

[1] KAUST; [2] Princeton

Summary

Deep learning can be used to help reconstruct low frequencies in seismic data, and to directly infer velocity models in simple cases. In order to succeed with deep learning, a good training set of velocity models is critical. We present a new way to design random models that are statistically similar to a given guiding model. Our approach is based on shuffling the coefficients of a wavelet packet decomposition (WPD) of the guiding model, allowing us to replicate realistic textures from a synthetic model. We generate realistically random models from the BP 2004 and Marmousi II models for neural network training, and utilize the trained network to extrapolate low frequencies for the SEAM model. We apply full-waveform inversion to the extrapolated data to understand the limitations of our approach.

Introduction: Full-waveform inversion (FWI) and synthetic model generation

Full-waveform inversion (FWI) uses the complete observed seismograms, and can provide very high-resolution results. However, the low-frequency data that are critical to the accurate convergence of FWI are hard to acquire in the field. Machine learning, and deep learning in particular, can be used to reconstruct the missing low frequencies (Ovcharenko et al., 2017; Sun and Demanet, 2018; Ovcharenko et al., 2018). Deep learning can also be used to infer velocity models directly from the seismic data (Araya-Polo et al., 2018; Wu et al., 2018). Both of these deep learning applications require training sets of seismic velocity models: ideally, a large set of seismic velocity models that are both realistic and sufficiently representative of possible geological features, while maintaining a small effective dimensionality of the model space.

Tools for generating velocity models for machine learning include horizontally layered models with faults (Wu et al., 2018), layered models with smooth boundaries (Araya-Polo et al., 2018), stretched chunks of known synthetic models (Sun and Demanet, 2018), and random Gaussian fields added to random but realistic velocity gradients (Ovcharenko et al., 2017, 2018). However, layered models oversimplify seismic models and push FWI back towards classic seismic velocity analysis. Using chunks of synthetic models shrinks the dimensions of the explored model space to very few parameters and, therefore, might limit the deep-learning generalization abilities. Finally, Gaussian random fields may result in unrealistic models of the subsurface.

To fulfill the requirements for a machine-learning training set, we exploit wavelet transforms. Thresholding a set of wavelet coefficients that represent existing synthetic models enforces low image dimensionality while keeping it realistic (Daubechies, 1992; Taubman and Marcellin, 2002). This has been recognized and utilized to improve FWI results (Aravkin et al., 2011; Lin et al., 2012; Ray et al., 2017). To make the models realistic, we propose decomposing the model into trends and textures. Trends can easily be captured by a spline/polynomial fit and randomized within realistic limits. Textures are harder to capture, but can be captured with wavelet packet decomposition (WPD) (Laine and Fan, 1993). We present a method to generate realistic models with WPD and apply the generated model set to extrapolate low frequencies with a deep convolutional network.

Optimizing the wavelet packet decomposition (WPD)

Standard wavelet analysis projects an image onto a multiresolution set of scale spaces, which we show in an example of the Marmousi II model (Fig. 1(a)) using the Haar basis. Scaling functions filter the image down to a coarse level of approximation, and wavelet functions extract the details at the current level. Hence, the model is decomposed into horizontal, vertical, and diagonal features (Fig. 1(b)). We observe that the Marmousi II model, with all its complexity, is mostly dominated by horizontal structure. The procedure is repeated on the low-pass component, which brings us to a standard pyramid representation of the image at two levels (Fig. 1(b)). The wavelet packet decomposition (WPD) goes further and breaks down each type of feature extracted by the wavelet transform with a single-level wavelet transform of that feature (Fig. 1(d)). This procedure is repeated recursively and leads to 4^n blocks of the coefficients, where n is the number of transformation levels. Such trees lead to better image compression (Christophe et al., 2008) and, more importantly for our application, expose the local textures of the image.

The choice of wavelet and the level of WPD remain subject to our investigation. We select optimally compressing wavelets out of Daubechies, symlets, coiflets, and biorthogonal families, and test different levels of WPD up to level six. Symlets with four vanishing moments (sym4) perform the best; the Daubechies wavelets with two vanishing moments (db2) have a similar performance. we truncate the Marmousi II models and chunks of the BP 2004 model to 10,000 WPD coefficients. Level five WPD with db2 and sym4 outperform other configurations in terms of the RMS error. Fig. 2 compares these wavelets with the most simple Haar wavelet. Fig. 3 shows a comparison between the basis functions in WPD for different wavelets and gives examples of the textures utilized to decompose the models. Note that with the standard wavelet transform, only scaled versions of the first row in Fig. 3 would be used.

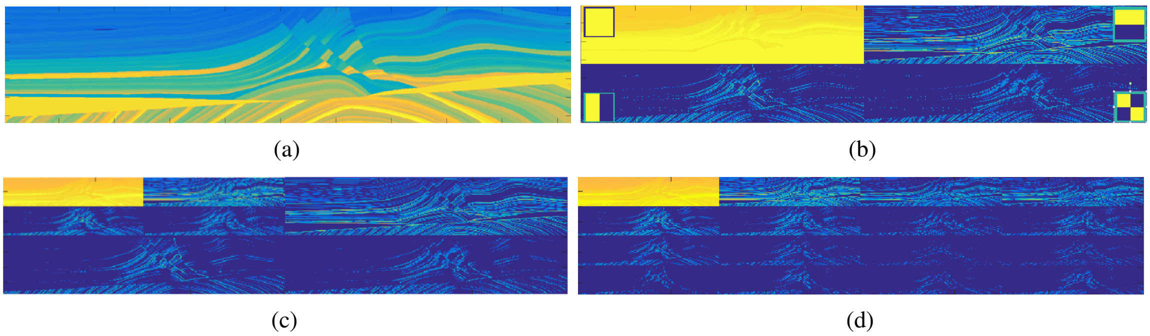


Figure 1 *(a) Marmousi II model velocity model. (b) Wavelet transform of the Marmousi model to the first level decomposes it into the next scale (left top), the horizontal (right top), the vertical (left bottom) and the diagonal (right bottom) features. (c) Two-level wavelet transform of the Marmousi model (coarse features of the first level are decomposed). (d) Wavelet packet decomposition (WPD) of the second level (all features of the first-level wavelet transform are decomposed).*

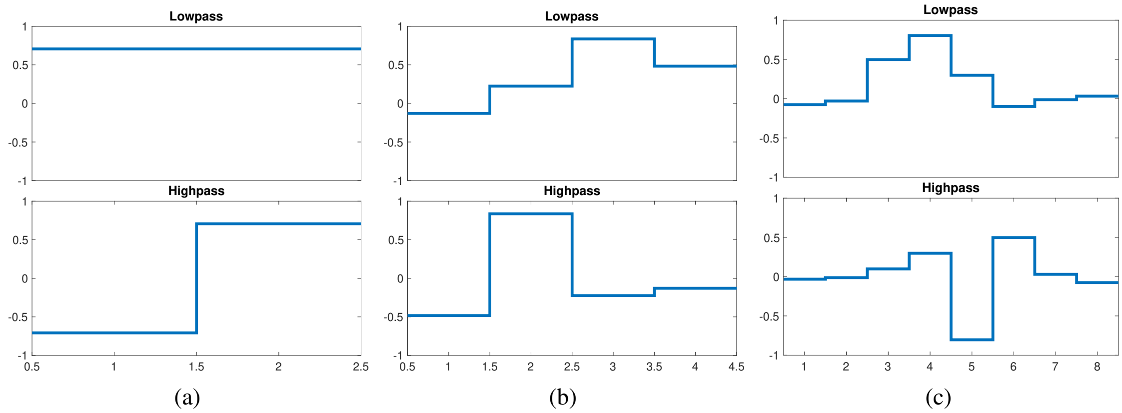


Figure 2 *(a) Haar (two-point support and zero mean and one vanishing moment), (b) db2 (D4, four-point support and two vanishing moments) and (c) sym4 (four vanishing moments and eight-point support) wavelets (bottom row) and scaling functions (top row).*

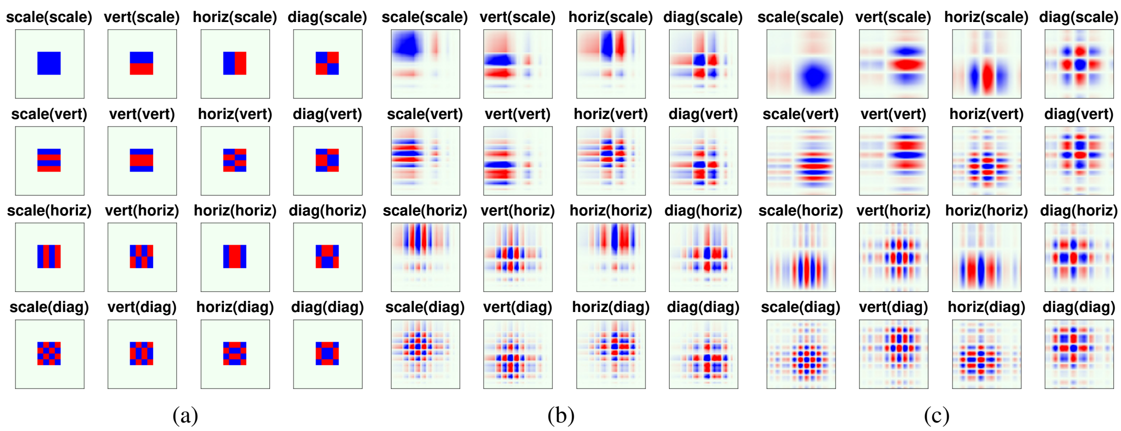
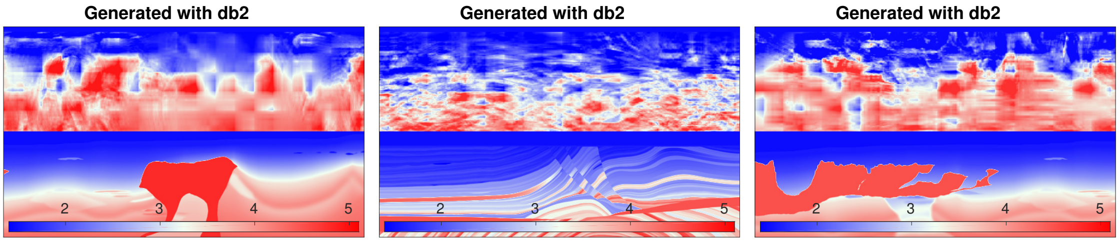
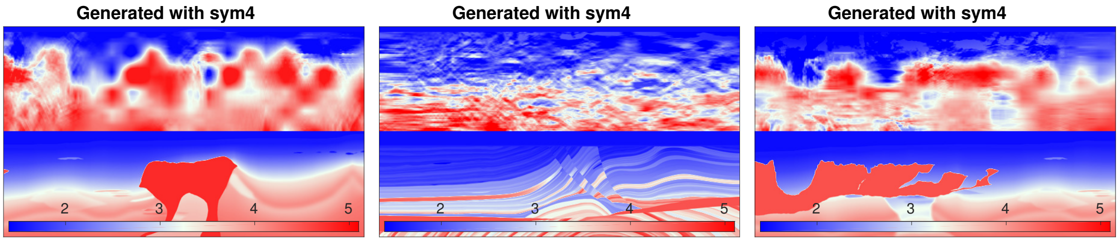


Figure 3 *First 16 out of 1024 (4^5) basis functions at level 5 for a wavelet packet decomposition (WPD) with (a) Haar, (b) db2 or D4 and (c) sym4 wavelets. For comparison, the standard wavelet transform relies only on the first rows of coefficients, scaling them to different levels.*



(a) (b) (c)

Figure 4 *Original guiding (bottom) and sample random (top) models generated by shuffling coefficients of the db2 (D4) wavelet packet decomposition (WPD).*



(a) (b) (c)

Figure 5 *As Fig. 4, but for a wavelet packet decomposition (WPD) based on sym4 wavelets.*

Model synthesis and application

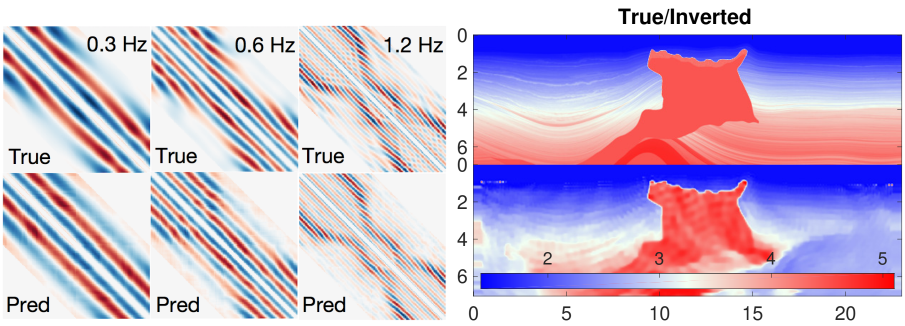
We propose randomly shifting the textures within the guiding model, which are captured by WPD, while randomizing the velocity trend separately. First, we shuffle the WPD coefficients of the guiding model. We use the same purely random shuffling procedure for all the blocks of the coefficients, except for the pure scaling coefficients. Then, we shuffle the scaling coefficients in such a way that they cannot move further than one third of the full model size. This limitation partly preserves the trends that exist in the guiding model. We set to zero all but the largest 10,000 WPD coefficients. Then, we apply wavelet packet reconstruction. We notice that any shuffling of the scaling coefficients effectively reduces the large scale trends in the model. Therefore, we replace the vertical linear trend of the generated model with a more realistic randomized trend.

To test the model generator, we use the Marmousi II model and chunks of the BP 2004 model as guiding models to generate a set of 200 realistically random models. Fig. 4 and Fig. 5 show random models generated by the process described above, based on the db2 (Fig. 2(b)) and sym4 (Fig. 2(c)) wavelets. The generated models look texturally similar to the guiding models. The models generated with sym4 wavelets look structurally more continuous, which is probably due to the better compression and extended length of the wavelet.

We model full-bandwidth data for the set of generated velocity models and use it as a training set for deep learning. We reconfigure the MobileNet (Howard et al., 2017) neural network to extrapolate low-frequency data for individual seismic shot gathers. We use 2–4.5 Hz data as inputs, and 0.3–1.2 Hz data as outputs. Previously, we used a simple random-model generator based on random Gaussian fields (Ovcharenko et al., 2018), which worked well for the extrapolation of the ultra-low-frequency data, but not for frequencies of 1 Hz and above. With the new training set, it is possible to predict the full range of low frequencies necessary for FWI to converge for the SEAM model, which was not used as a guiding model to train the data generation (Fig. 6).

Conclusions

We proposed a new method of generating realistically random models by shuffling and distorting the wavelet packet decomposition (WPD) coefficients of guiding models. Filtering out the low values of



(a) (b)

Figure 6 *(a) Extrapolated low-frequency data (real part) for 0.3, 0.6, and 1.2 Hz, from 2-4.5 Hz synthetic data for the SEAM model. (b) SEAM velocity model (top) and FWI started with extrapolated frequencies and finalized with known frequencies (bottom).*

the WPD coefficients allows us to ensure a low effective dimensionality of the problem. The generated models are texturally similar to the guiding models: the guiding model textures are simply translated to new locations within the model. The generated models can be useful in applications of machine learning to seismic inverse problems.

Acknowledgements

We thank King Abdullah University of Science and Technology (KAUST) for support.

References

Aravkin, A.Y., Van Leeuwen, T., Burke, J.V. and Herrmann, F.J. [2011] A Nonlinear Sparsity Promoting Formulation and Algorithm for Full Waveform Inversion. In: *73rd EAGE Conference and Exhibition Incorporating SPE EUROPEC 2011*.

Araya-Polo, M., Jennings, J., Adler, A. and Dahlke, T. [2018] Deep-Learning Tomography. *The Leading Edge*, **37**(1), 58–66.

Christophe, E., Mailhes, C. and Duhamel, P. [2008] Hyperspectral Image Compression: Adapting SPIHT and EZW to Anisotropic 3-D Wavelet Coding. *IEEE Trans. Image Proc.*, **17**(12), 2334–2346.

Daubechies, I. [1992] *Ten Lectures on Wavelets*, SIAM.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H. [2017] Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv:1704.04861*.

Laine, A. and Fan, J. [1993] Texture Classification by Wavelet Packet Signatures. *IEEE Trans. Pattern Analysis Machine Intelligence*, **15**(11), 1186–1191.

Lin, Y., Abubakar, A. and Habashy, T.M. [2012] Seismic Full-Waveform Inversion Using Truncated Wavelet Representations. In: *SEG Tech. Progr. Expanded Abstracts 2012*, Society of Exploration Geophysicists, 1–6.

Ovcharenko, O., Kazei, V., Peter, D. and Alkhalifah, T. [2017] Neural Network Based Low-Frequency Data Extrapolation. In: *3rd SEG FWI Workshop: What Are We Getting?*

Ovcharenko, O., Kazei, V., Peter, D., Zhang, X. and Alkhalifah, T. [2018] Low-Frequency Data Extrapolation Using a Feed-Forward ANN. In: *80th EAGE Conference and Exhibition 2018*.

Ray, A., Kaplan, S., Washbourne, J. and Albertin, U. [2017] Low Frequency Full Waveform Seismic Inversion within a Tree Based Bayesian Framework. *Geophys. J. International*, **212**(1), 522–542.

Sun, H. and Demanet, L. [2018] Low Frequency Extrapolation with Deep Learning. In: *SEG Tech. Progr. Expanded Abstracts 2018*, Society of Exploration Geophysicists, 2011–2015.

Taubman, D.S. and Marcellin, M.W. [2002] JPEG2000: Standard for Interactive Imaging. *Proc. IEEE*, **90**(8), 1336–1357.

Wu, Y., Lin, Y. and Zhou, Z. [2018] InversionNet: Accurate and Efficient Seismic Waveform Inversion with Convolutional Neural Networks. In: *SEG Tech. Progr. Expanded Abstracts 2018*, Society of Exploration Geophysicists, 2096–2100.