

Neural networks for geophysicists and their application to seismic data interpretation



Bas Peters¹, Eldad Haber¹, and Justin Granek²

<https://doi.org/10.1190/tle38070534.1>

Abstract

There has been a surge of interest in neural networks for the interpretation of seismic images over the last few years. Network-based learning methods can provide fast and accurate automatic interpretation, provided that there are many training labels. We provide an introduction to the field for geophysicists who are familiar with the framework of forward modeling and inversion. We explain the similarities and differences between deep networks and other geophysical inverse problems and show their utility in solving problems such as lithology interpolation between wells, horizon tracking, and segmentation of seismic images. The benefits of our approach are demonstrated on field data from the Sea of Ireland and the North Sea.

Introduction

Deep neural networks (DNNs) have revolutionized computer vision, image processing, and image understanding (e.g., Deng et al., 2009; Krizhevsky and Hinton, 2009; Ronneberger et al., 2015; Goodfellow et al., 2016). In particular, deep convolutional networks have solved long-standing problems such as image classification, segmentation, deblurring, denoising, and more. Most of the applications are based on supervised learning — we are given some data and corresponding interpretation or labels. The goal of the network is to empirically find the connection between the data and labels.

Seismic interpretation can be viewed as a type of image understanding in which the 3D image is the seismic cube, and interpretation of the seismic data (e.g., horizons, faults, etc.) are the labeled features that need to be recovered. Using deep convolutional networks is therefore a straightforward extension of existing neural-network technology. It has been studied recently by many authors (e.g., Poulton, 2002; Leggett et al., 2003; Lowell and Paton, 2018; Waldebrand et al., 2018; Wu and Zhang, 2018; Zhao, 2018; Peters et al., 2019a, 2019b).

However, while it seems straightforward to use such algorithms, there are some fundamental differences between vision-related applications to seismic processing. First, and maybe most importantly, is the amount of labeled or annotated data available. While in computer vision, labeled data are easy to obtain, it is much more difficult to do so for seismic applications. Second, while labeled data are likely to be correct in vision, it is much more uncertain in seismic interpretation. For example, when viewing an image, it is usually obvious if an object such as a car exists within a frame. On the other hand, two geologists may argue about the existence or the exact location of a particular fault

or a deep horizon. This makes data for the seismic problem biased. Third, even for labeled data, in most applications, the data are not fully labeled and only small portions have been annotated. Finally, while most vision data are 2D, seismic data are typically 3D and therefore should be learned in 3D when possible. This makes using graphical processing units challenging due to memory restrictions, especially when the networks are deep and wide.

In this paper, we review and discuss some recent work that we and others have done to tackle some of the challenges when attempting to use deep networks for problems that arise from seismic interpretation. In particular, we address DNNs from a geophysicist's point of view in terms of network design and optimization. We show that the network can be interpreted as a forward problem, while learning can be interpreted as the inverse problem. Any geophysicist that is familiar with the process of modeling and inversion can understand the process and draw from previous experiences.

In the rest of the paper, we give background information about deep networks. In particular, we discuss the connection between deep networks to differential equations and show that the machine learning problem is similar to other well-studied problems in geophysics such as full-waveform inversion or electromagnetic forward and inverse problems. This should make it easy for any geophysicist with such background to understand and contribute to the field. We then discuss two different applications that can be tackled using this framework. First, we explain how DNNs can interpolate lithology, given sparse borehole information and seismic data. Next, we show how networks can predict multiple horizons including branching horizons. We then summarize the paper and discuss and suggest future applications.

Deep neural networks — A geophysicist's view

Suppose we are given data D and the corresponding label map C . If there is a physical basis to obtain C from D , then we should use it. For example, assume that D is a velocity model and C is a seismic cube. In this case, we can use the wave equation to obtain C from D . However, such a physical mapping is unavailable for many problems in science and engineering. Since there is no physical basis to recover C from D , we turn to an empirical relationship. Many empirical models work well for different applications. For problems where D and C have spatial interpretation, DNNs have been successful in capturing the information and generating empirical relationships that hold well in practice.

A deep network is a chain of nonlinear transformations of the data. In particular, we turn to recent work (He et al., 2016;

¹University of British Columbia, Vancouver, Canada. E-mail: bpeters@eoas.ubc.ca; ehaber@eoas.ubc.ca.

²Computational Geosciences Inc., Vancouver, Canada. E-mail: justin@compgeoinc.com.

Haber and Ruthotto, 2017; Chang et al., 2018) that uses residual networks that have the form:

$$Y_{j+1} = Y_j - K_j^T \sigma(K_j Y_j + B_j), j = 1, \dots, n \text{ with } Y_1 = D. \quad (1)$$

Here, Y_j is state, K_j is convolution kernel, B_j is bias vector, and $\sigma(\cdot)$ is a nonlinear activation function. Examples of point-wise activation functions are the sigmoid, hyperbolic tangent, or ReLU function: $\text{ReLU}(K_j Y_j + B_j) = \max(0, K_j Y_j + B_j)$. It is crucial that the activation functions are nonlinear, otherwise the entire network would simplify to a linear operator. In this work, we use the ReLU function, which sets all negative values in its input to zero. This is equivalent to projection onto the space of positive numbers — a halfspace. We thus see that every network state Y depends on the previous network state and a nonlinearly transformed version of the convolved and shifted previous network state.

Given the network (equation 1), we push the data forward through n layers to obtain Y_n . Given Y_n , it is possible to predict the label C by simply multiplying Y_n by a matrix W :

$$C = W Y_n. \quad (2)$$

Let us review the process from a geophysicist's point of view and show that it is equivalent to many other forward problems in geophysics. To this end, the deep network (equation 1) can be viewed as a discretization of a physical process (e.g., the wave or Maxwell's equations). From this point of view, Y_j is the field (e.g., acoustic or electromagnetic), and K_j and B_j are model parameters such as seismic velocity or electric conductivity. Just like in any other field, when considering the forward problem, we assume that we know the model parameters, and therefore we can predict the fields Y . The classification process in equation 2 can be interpreted as projecting the fields to measure some of their properties. A similar process in geophysics is when W is a projection matrix that measures the field at some locations, that is, in receiver positions.

It is important to stress that the network presented in equation 1 is just one architecture that we can use. For problems of semantic segmentation, it has been shown that coupling a few of these networks, each on a different resolution, gives much better results than using a single resolution. The idea behind such networks is plotted in Figure 1. We refer the reader to Ronneberger et al. (2015) for more detail on efficient network architectures that deal with data with multiple scales.

In general, the model parameters K_j and B_j are unknown in practice and need to be calibrated from the data. This process is similar to finding the seismic velocity model or electric conductivity from some measured geophysical data. To this end, we assume

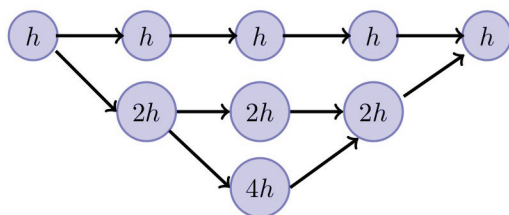


Figure 1. A number of resnets with scales h (original image), $2h$ (coarse image), and $4h$. The networks are coupled by restriction and prolongation and are used to deal with data at different resolutions.

that we have some observed labels C^{obs} . The learning problem can be framed as a parameter estimation problem or an inverse problem, where we fit the observed labels by minimizing an objective function, for example the ℓ_1 -norm that we use for the horizon prediction example later in this work:

$$\min_{\theta} \|C(\theta) - C^{obs}\|_1 + \alpha R(\theta). \quad (3)$$

Here, we introduce the cumulation of model parameters $\theta = \{K_1, \dots, K_n, B_1, \dots, B_n\}$ and a regularization term $R(\theta)$. Most literature assumes that $R(\theta)$ is a simple Tikhonov regularization or, in the language of deep learning, weight decay:

$$R(\theta) = \frac{1}{2} \sum_j \|K_j\|_F^2 + \|B_j\|^2. \quad (4)$$

As we will show next, such basic regularization may not be sufficient for problems that arise from seismic applications, and we review other more appropriate regularization for the problems presented here.

Depending on the applications, we may also require a different objective function than the ℓ_1 objective (equation 3). For the lithology interpolation example presented later, which is classification based, we use the multiclass cross-entropy loss. We note that different objective functions have very similar structure in terms of input, output, what parameters we learn, and how we can regularize.

While we have emphasized similarities between the training problem and other geophysical problems, at this point, it is worthwhile pointing out two fundamental differences between deep learning and geophysical inverse problems. First, and most important, in geophysics we are interested in the model θ . Such a model generally has some physical attributes that we are interested in. The model typically represents velocity, conductivity, porosity, or other physical properties. In machine learning, on the other hand, the model parameters do not have an obvious physical meaning (that we know of), and therefore it is hard to know what is a “reasonable” model. Second, optimizing the objective function in equation 3 is typically done using stochastic gradient descent (Bottou and Bousquet, 2008), whereas quasi-Newton and Gauss-Newton methods are more common in seismic and electromagnetic nonlinear geophysical inverse problems. In the following sections, we show how we use the setting discussed earlier to solve a number of practical problems that arise in seismic interpretation.

Applications to seismic interpretation

In this section, we discuss the application of deep networks to two seismic applications. All applications share the same forward propagation process, and the main difference is the way we set up the loss function (misfit) and regularization. We find it rather remarkable that similar network architectures work for such different problems. This emphasizes the strength of deep learning applied to seismic interpretation.

One common feature that most geophysical problems share is that the labels C^{obs} are not present for the whole seismic image. For example, it is common to have part of the image labeled but not all of it. Another example is that we know only part of a horizon. This is in stark contrast to most computer vision problems where the images are fully labeled. This difference results from the technical difficulty and expertise that is needed to label seismic data. While most nonspecialists can identify a car in an image, an expert may be needed to classify a seismic unit. However, we note that most applications in geophysics share this type of sparse measurement. For example, we never have a fully observed wavefield when considering the full-waveform inversion, and the misfit is calculated only on the observable point (where we record the data). We therefore modify common loss functions in DNN training to return the misfit only from the locations where the image is labeled.

Interpolation of lithology between wells using seismic data

Consider some boreholes, and assume that geologic lithology is observed within the boreholes. Our goal is to use lithology information from the wells to interpret the seismic image (Figure 2a).

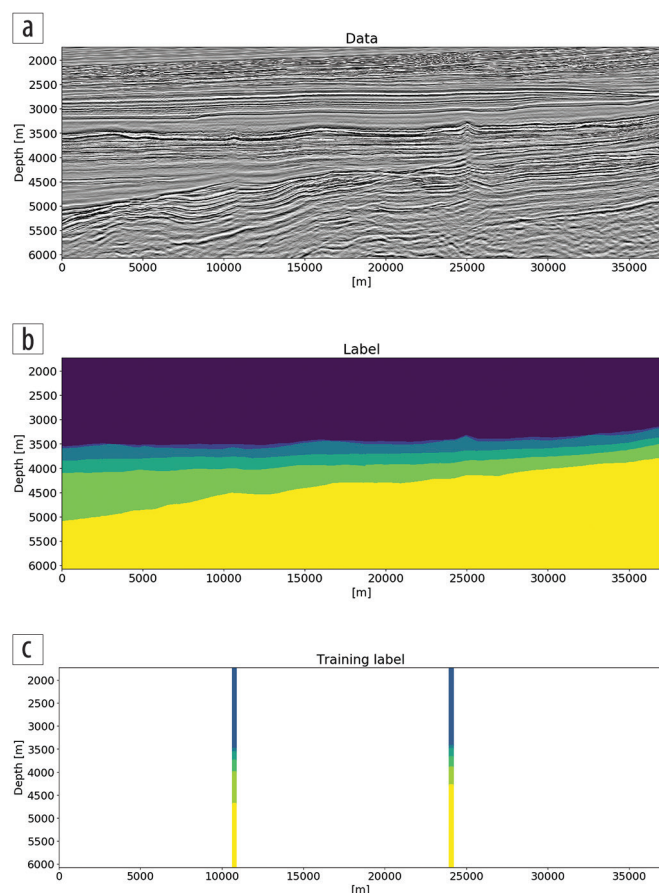


Figure 2. (a) A slice from a 3D seismic model. This is an example of an input for the network. (b) A fully annotated label image where each color indicates a rock/lithology type of interest. We do not use full labels as the target for our networks because they are time consuming to generate. (c) An example of a type of label that we use in our examples. The information corresponds to the lithological units derived from logs in two wells. The white space is not used to measure the misfit or compute a gradient; it is unknown information not used for training the network.

Specifically, we illustrate the benefits of being able to train on sparse labels (such as in Figure 2c) and predict fully annotated images (Figure 2b).

When minimizing the loss (equation 3) discussed earlier, artifacts typically appear in the prediction. These artifacts are a result of the lack of labels everywhere. To overcome this problem, we propose to add new regularization terms to the loss. This regularization penalizes unwanted oscillations in the prediction maps.

Note that the true label images we hope to predict are blocky (Figure 2b). Such an end result is obtained as follows. If we seek to classify our seismic image into, for example, six geologic units, the network output consists of six images (see Figure 3a for an example of the predicted probability for one class). Each of the six output images conveys the probability of that particular class being present at each pixel. The sum over all six output probability images for each pixel is 1. This implies that the underlying probability of each lithological unit should be smooth. The probability of a particular class changes smoothly from low to high across the interface if the network is well trained. When there are not many labeled pixels available, for example, only in a few boreholes, the predicted probabilities are often highly oscillatory and result in erratic classifications that do not look geologically plausible.

We propose to mitigate a lack of labels everywhere by using the prior knowledge that the prediction per class should be smooth. This type of prior information fits in the neural-network training process as a penalty function on the output of the network. To this end, consider solving an optimization problem of the form:

$$L(C(\theta), C^{obs}) = \ell(C(\theta), C^{obs}) + \alpha R(Y_n(\theta)). \quad (5)$$

We minimize the multiclass cross-entropy ℓ plus a regularization term. The regularization $R(\cdot)$ is chosen as

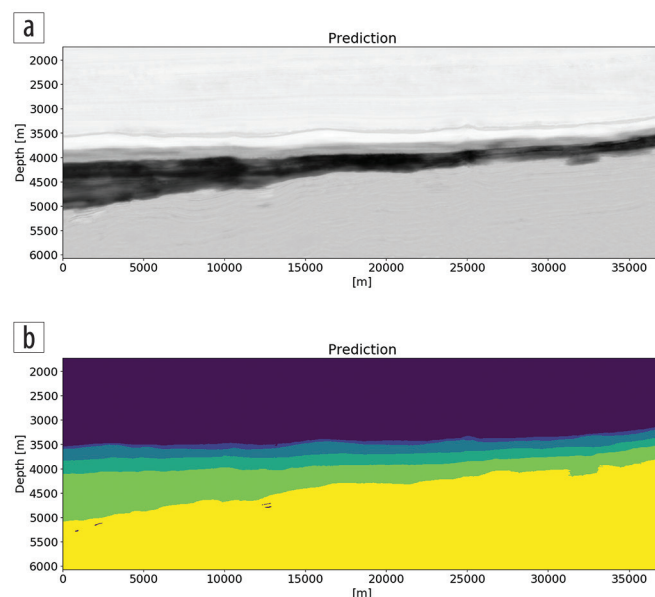


Figure 3. (a) Prediction for a single class and (b) maximum predicted class probability per pixel. Both are the result of training including regularization on the network output.

$$R(C) = \frac{1}{2} \|\nabla_h Y_n(\theta)\|^2, \quad (6)$$

where ∇_h is a discrete gradient matrix (Haber, 2014) that can be implemented using convolutions with kernels of ± 1 . This means we apply quadratic smoothing regularization to the last network state (i.e., the output probability maps). This is different from standard Tikhonov regularization in geophysical inverse problems or weight-decay regularization on the convolutional kernels in neural networks. In the current case, the regularization applies to state that represents an image that we know should be smooth. Tikhonov and weight-decay regularization apply to the model parameters. We thus see that the concepts that apply to model parameters in geophysical inversion are analogous to the network state in network-based image interpretation.

Note that regularization always applies to the full network output. The output is a full image regardless of sparse sampling of data and/or labels. We can still subsample to introduce randomization or for computational reasons.

To apply the objective and regularization tools presented so far, we train a network using the loss function defined in equation 5 with quadratic smoothing regularization (equation 6) applied to the network output. The prediction in Figure 3a is smooth, and the maximum predicted class probability per pixel in Figure 3b is a good approximation to the true map as verified by Figure 4. Without regularization, the prediction contains many oscillatory artifacts.

Horizon tracking by interpolation of scattered picks

Our second application is tracking a horizon from a small number of horizon picks (seed points) in a few large seismic images. Horizon tracking using neural networks has seen a few time periods of varying activity (Liu et al., 1989; Harrigan et al., 1992; Kusuma and Fish, 1993; Veezhinathan et al., 1993; Alberts et al., 2000; Huang et al., 2005). Algorithms that are not based on learning have also made progress (see Wu and Fomel [2018] for recent work that combines and extends multiple concepts on deterministic horizon tracking).

It was shown previously (Peters et al., 2019a) that it is possible to track a single horizon using U-net-based networks and loss functions that compute losses and gradients based on the sparse labels only. Therefore, there was no need to work in small patches around labeled points or manually generate fully annotated label images. Here, we answer two follow-up questions. (1) Can we

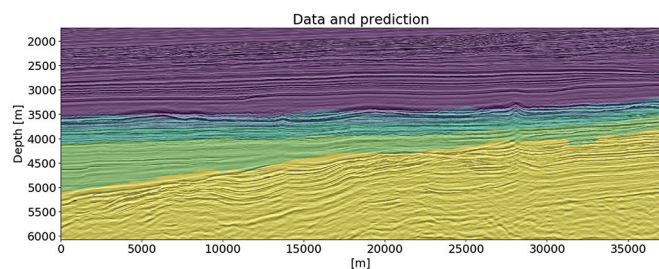


Figure 4. The predicted segmentation from Figure 3b (using network output regularization) overlaid on the seismic input data.

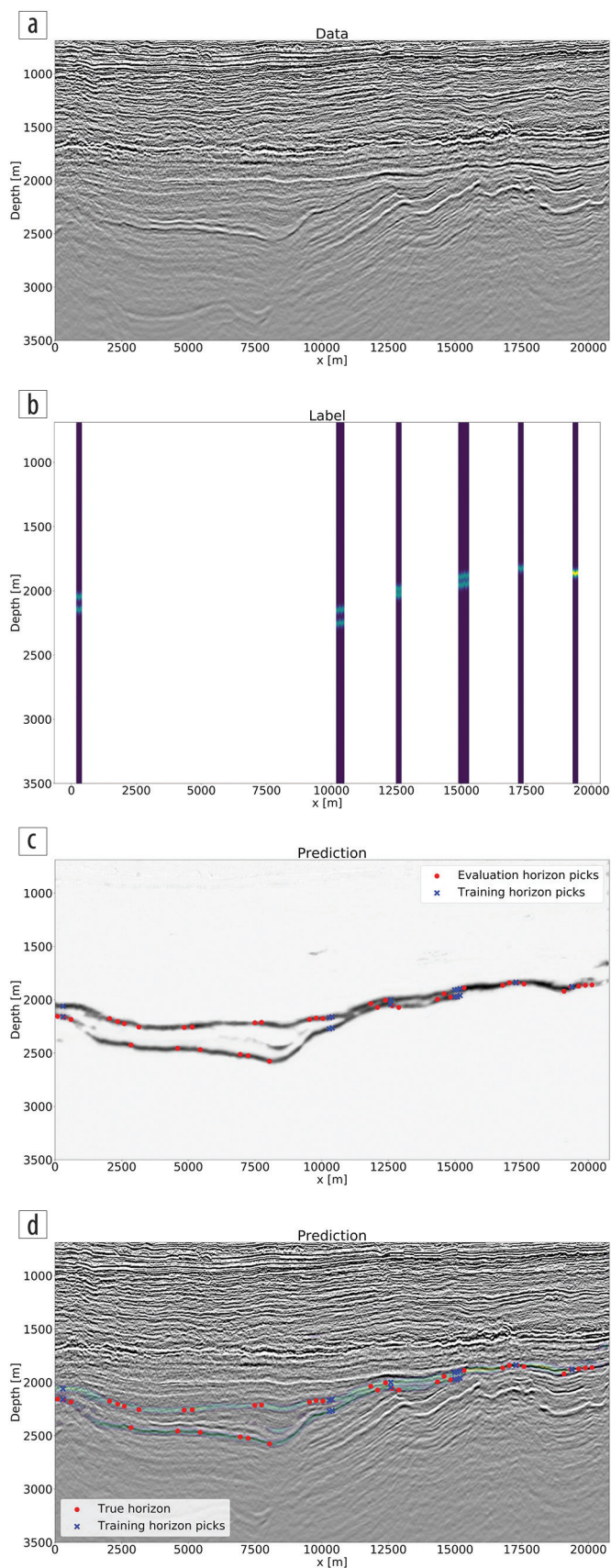


Figure 5. (a) One of the data images. (b) A label image. About 10 columns per image are known; the network never uses the white space. The labels are the convolutions of a Gaussian kernel with the horizon picks. (c) Network output with training and testing picks. (d) Color-coded network horizon prediction on top of the data.

Table 1. Overview of similarities and differences between geophysical forward/inverse modeling and neural networks for interpretation of geophysical data/images.

	Geophysical forward problem	Geophysical inverse problem	Neural network
Discrete problem structure	Discretized differential operators in time-stepping scheme		Network structure w.r.t. Y_j , e.g., $Y_{j+1} = Y_j - K_j^T \sigma(K_j Y_j + B_j)$
Model parameters	Known physical parameters	Unknown physical parameters	Unknown convolutional kernels with unclear meaning
Model parameter regularization		Tikhonov regularization on physical model parameters	Weight decay on kernels and biases
Output/state regularization	Would be equivalent to: regularization of final elastic/electromagnetic field		Tikhonov regularization on final network state (probability maps)

train a network to track more than one horizon simultaneously? (2) How do networks deal with multiple horizons that merge and split? These two questions warrant a new look at the automatic horizon tracking/interpolation problem because results with merging horizons are rarely published. Especially since there is a renewed surge of interest in using neural networks for seismic interpretation, we need to test the promise of networks against the more challenging situation posed in these two questions.

We demonstrate our method using a 3D seismic data set from the North Sea. One of the 100 slices is shown in Figure 5a. An industrial partner provided us the horizon x-y-z locations, picked by seismic interpreters because their auto-tracking algorithms had difficulty tracking the deeper horizons. We create a label image by convolving the horizon picks (seed points) with a Gaussian kernel in the vertical direction. This procedure adds a sense of uncertainty to the pick. We use approximately 10 locations per slice for training, as shown in Figure 5b. Only the colored columns are used to train the network; in the white space, it is unknown if and where the horizon is. The loss function only uses information in the known label columns. We see that there are two horizons of interest that merge near the right side of the figure and also get close to each other at the left end. We train a single network to predict both horizons simultaneously using the nonlinear regression and optimization approach detailed in Peters et al. (2019a). The network design is as described earlier in this work.


Figure 5c displays the network output, which ideally is the true horizon everywhere convolved with the Gaussian kernel that we used to generate training label images. The training and evaluation picks are plotted on top and validate that the network is able to predict both horizons accurately including the point where they merge. In Figure 5d, we show the network output prediction plotted on top of the seismic data to provide some more insight. The color coding corresponds to the grayscale intensity of the previous figure. The colors and vertical spread indicate how “sure” the network thinks it is about the prediction.

From the results, we conclude that we can train a single network to simultaneously predict the location of multiple horizons that merge and branch. The symmetric convolutional U-net variant, with the same network architecture as in the previous example, trained by a partial loss function on a small number of known horizon x-y-z locations, achieves excellent results. Data augmentation and regularization, as described earlier, can reduce the number of required training x-y-z picks.

Conclusions

In this paper, we introduced DNNs from an inverse problem point of view. We have shown that the network can be considered as the forward problem and the training as the inverse problem. We have explored the connection between deep networks to other geophysical inverse problems. We believe that approaching the learning problem in this way allows us to better understand the role of data fitting, regularization, the stability of the network itself, the propagation of noise within the network, and the associated uncertainties — all topics that have received ample treatment in geophysical inverse problems.

We demonstrated the capability of deep networks to deal with problems that arise from seismic interpretation. In our experience, neural networks can do exceptionally well for such problems, given some thought about appropriate regularization and loss or misfit functions. The similarities and differences between the neural networks and geophysical inverse problems, sometimes subtle, should offer a bridge to better understand the potential and success of neural networks for geophysical interpretation.

When solving a particular problem, it is important to realize that geophysical problems are very different from common vision problems. The availability of accurate training data is key to training the network, and this can be difficult to obtain in many applications. Another important aspect is the size of the data. While vision problems are typically 2D, many geophysical problems are 3D. We believe that new algorithms should be developed to deal with the size of geophysical images as well as with the uncertainty that is an inherent part of geophysical processing. 

Data and materials availability

Data associated with this research are confidential and cannot be released.

Corresponding author: bpeters@eoas.ubc.ca

References

- Alberts, P., M. Warner, and D. Lister, 2000, Artificial neural networks for simultaneous multi horizon tracking across discontinuities: 70th Annual International Meeting, SEG, Expanded Abstracts, 651–653, <https://doi.org/10.1190/1.1816150>.
- Bottou, L., and O. Bousquet, 2008, The tradeoffs of large scale learning: Proceedings of the Conference on Neural Information Processing Systems, 161–168.

- Chang, B., L. Meng, E. Haber, L. Ruthotto, D. Begert, and E. Holtham, 2018, Reversible architectures for arbitrarily deep residual neural networks: 32nd Conference on Artificial Intelligence, AAAI.
- Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, 2009, ImageNet: A large-scale hierarchical image database: Conference on Computer Vision and Pattern Recognition, IEEE, 248–255, <https://doi.org/10.1109/CVPR.2009.5206848>.
- Goodfellow, I., Y. Bengio, and A. Courville, 2016, Deep learning: MIT Press.
- Haber, E., 2014, Computational methods in geophysical electromagnetics: SIAM.
- Haber, E., and L. Ruthotto, 2017, Stable architectures for deep neural networks: Inverse Problems, **34**, no. 1, <https://doi.org/10.1088/1361-6420/aa9a90>.
- Harrigan, E., J. R. Kroh, W. A. Sandham, and T. S. Durrani, 1992, Seismic horizon picking using an artificial neural network: International Conference on Acoustics, Speech, and Signal Processing, IEEE, <https://doi.org/10.1109/ICASSP.1992.226265>.
- He, K., X. Zhang, S. Ren, and J. Sun, 2016, Deep residual learning for image recognition: Proceedings of the Conference on Computer Vision and Pattern Recognition, IEEE, 770–778, <https://doi.org/10.1109/CVPR.2016.90>.
- Huang, K.-Y., C.-H. Chang, W.-S. Hsieh, S.-C. Hsieh, L. K. Wang, and F.-J. Tsai, 2005, Cellular neural network for seismic horizon picking: Proceedings of the 9th International Workshop on Cellular Neural Networks and their Applications, 219–222, <https://doi.org/10.1109/CNNA.2005.1543200>.
- Krizhevsky, A., and G. Hinton, 2009, Learning multiple layers of features from tiny images: Technical Report, University of Toronto.
- Kusuma, T., and B. C. Fish, 1993, Toward more robust neural-network first break and horizon pickers: 63rd Annual International Meeting, SEG, Expanded Abstracts, 238–241, <https://doi.org/10.1190/1.1822449>.
- Leggett, M., W. A. Sandham, and T. S. Durrani, 2003, Automated 3-D horizon tracking and seismic classification using artificial neural networks, *in* W. A. Sandham and M. Leggett, eds., Geophysical Applications of Artificial Neural Networks and Fuzzy Logic: Springer, 31–44, https://doi.org/10.1007/978-94-017-0271-3_3.
- Liu, X., P. Xue, and Y. Li, 1989, Neural network method for tracing seismic events: 59th Annual International Meeting, SEG, Expanded Abstracts, 716–718, <https://doi.org/10.1190/1.1889749>.
- Lowell, J., and G. Paton, 2018, Application of deep learning for seismic horizon interpretation: 88th Annual International Meeting, SEG, Expanded Abstracts, 1976–1980, <https://doi.org/10.1190/segam2018-2998176.1>.
- Peters, B., J. Granek, and E. Haber, 2019a, Multi-resolution neural networks for tracking seismic horizons from few training images: Interpretation, **7**, no. 3, 1–54, <https://doi.org/10.1190/int-2018-0225.1>.
- Peters, B., J. Granek, and E. Haber, 2019b, Automatic classification of geologic units in seismic images using partially interpreted examples: 81st Conference and Exhibition, EAGE, Extended Abstracts.
- Poulton, M. M., 2002, Neural networks as an intelligence amplification tool: A review of applications: Geophysics, **67**, no. 3, 979–993, <https://doi.org/10.1190/1.1484539>.
- Ronneberger, O., P. Fischer, and T. Brox, 2015, U-Net: Convolutional networks for biomedical image segmentation: Medical Image Computing and Computer-Assisted Intervention, 234–241, https://doi.org/10.1007/978-3-319-24574-4_28.
- Veezhinathan, J., F. Kemp, and J. Threet, 1993, A hybrid of neural net and branch and bound techniques for seismic horizon tracking: Proceedings of the Symposium on Applied Computing: States of the Art and Practice: ACM/SIGAPP, 173–178, <https://doi.org/10.1145/162754.162863>.
- Waldeland, A. U., A. C. Jensen, L.-J. Gelius, and A. H. S. Solberg, 2018, Convolutional neural networks for automated seismic interpretation: The Leading Edge, **37**, no. 7, 529–537, <https://doi.org/10.1190/tle37070529.1>.
- Wu, H., and B. Zhang, 2018, A deep convolutional encoder-decoder neural network in assisting seismic horizon tracking: arXiv:1804.06814.
- Wu, X., and S. Fomel, 2018, Least-squares horizons with local slopes and multigrid correlations: Geophysics, **83**, no. 4, IM29–IM40, <https://doi.org/10.1190/geo2017-0830.1>.
- Zhao, T., 2018, Seismic facies classification using different deep convolutional neural networks: 88th Annual International Meeting, SEG, Expanded Abstracts, 2046–2050, <https://doi.org/10.1190/segam2018-2997085.1>.