Data-driven geophysics: from dictionary learning to deep learning

Siwei Yu¹ and Jianwei Ma¹,²

¹ Center of Geophysics, Artificial Intelligence Laboratory, and School of Mathematics, Harbin Institute of Technology, Harbin, China. ² School of Earth and Space Sciences, Peking University, Beijing, China.

Abstract

Understanding the principles of geophysical phenomena is an essential and challenging task. ‘Model-driven’ approaches have supported the development of geophysics for a long time; however, such methods suffer from the ‘curse of dimensionality’ and may inaccurately model the subsurface. ‘Data-driven’ techniques may overcome these issues with increasingly available geophysical data. In this article, we review the basic concepts of and recent advances in data-driven approaches from dictionary learning to deep learning in a variety of geophysical scenarios, including seismic and earthquake data processing, inversion, and interpretation. We present a coding tutorial and a summary of tips for beginners and interested geophysical readers to rapidly explore deep learning. Some promising directions are provided for future research involving deep learning in geophysics, such as unsupervised learning, transfer learning, multimodal deep learning, federated learning, uncertainty estimation, and activate learning.

Introduction

‘Model-driven’ approaches are mathematical or geophysical modeling methods established based on physical causality, and they have played a vital role in the evolution of geophysical methods. Geophysics involves the physical mechanics and properties of the Earth and space environments, and the subject encompasses seismology, gravity fields, magnetic fields, electric fields, the atmosphere, and the internal structures of the Earth and other planets. In this paper, we focus on explorational geophysics and some seismological problems. Exploration geophysics aims to observe the subsurface of the Earth or other planets with physical fields collected at the surface, such as seismic fields and gravity fields. The process of exploration geophysics includes signal processing, modeling, inversion, and interpretation.

In the stage of geophysical signal processing, the simplest model assumption regarding the shape of underground layers is the linear assumption in small windows (Spitz [1]). Further assumptions include the sparsity (Herrmann and Hennenfent [2]) and low-rank (Oropeza and Sacchi [3]) assumptions, among others. However, the predesigned linear event assumption or sparse transform assumption is not adaptive to seismic data and may lead to low denoising or interpolation quality for data with complex structures. In the stages of seismic modeling and inversion, wave equations govern the kinematics and dynamics of seismic wave propagation. Acoustic, elastic, or viscoelastic wave equations introduce an increasing number of factors into the wave equations, and the generated wave field records can precisely approximate real scenarios. However, as the wave equation becomes increasingly complex, the numerical implementation of the equation becomes nontrivial, and the computational cost increases considerably for large-scale scenarios. In seismic
interpretation tasks, traditional methods rely on the experience of interpreters, leading to low efficiency and subjective bias. Therefore, we desire an adaptive and automated process in geophysical fields.

‘Data-driven’ methods were proposed to overcome the bottlenecks of model-driven approaches in situations in which the prior assumptions are not satisfied (Figure 1). Dictionary learning (Aharon et al. [4]) and deep learning (LeCun et al. [5]) are two widely adopted data-driven techniques. In dictionary learning, an adaptive dictionary is learned as a representation of the target data rather than utilizing a predefined dictionary. Dictionary learning is widely applied in seismic data denoising, interpolation, and inversion based on compressive sensing (Candes and Wakin [6]). The key features of dictionary learning are single-level decomposition, unsupervised learning, and linearity. Single-level decomposition means that one dictionary is used to represent a signal. Unsupervised learning means no labels are provided during dictionary learning. In addition, only the target data are used without an extensive training set. Linearity implies that the data decomposition on the dictionary is linear. The above features make the theory of dictionary learning simple; however, they limit the data representation ability of a dictionary. Unlike dictionary learning, deep learning decomposes data at a deep level and has an excellent representation ability.

Figure 1. Examples of model-driven and data-driven methods. (a) Model-driven methods. In random
denoising tasks, the curvelet denoising method (Herrmann and Hennenfent [2]) assumes that the signal is sparse under curvelet transform, and a matching method is used for denoising. In velocity inversion tasks, full-waveform inversion based on the wave equation is used for forward and adjoint modeling in the optimization algorithm. In fault interpretation tasks, faults are picked by interpreters. (b) Data-driven methods. The mentioned tasks are treated as regression problems that are optimized with neural networks. Note that different tasks may require different neural network architectures.

Deep learning methods train a deep neural network (DNN) through a complex nonlinear mapping process with adjustable parameters based on a large dataset. The trained DNN is used to predict the desired output of the target data for a specific task. Deep learning encompasses both supervised and unsupervised data-driven approaches depending on whether labels are available. Recently, deep learning methods have been widely adopted in various geophysical applications, such as solid Earth geoscience (Bergen et al. [7]), aftershock pattern analysis (DeVries et al. [8]), and Earth system analysis (Reichstein et al. [9]). In this paper, we focus on exploration geophysics and some seismological problems.

The contributions of this paper are as follows.

- We provide a review of data-driven methods, including dictionary learning and deep learning, rather than a review of only deep learning methods. In addition, we show the relationship between dictionary learning and deep learning.
- While introducing deep learning in geophysics, we group the references by the methods used instead of the applications to emphasize the usability of each neural network architecture. We omit the details of each architecture and only show the concepts required to understand each conceptual approach. We try to cover most network architectures rather than most references.
- A coding tutorial and summary of tips for beginners are given to allow interested geophysical readers to quickly understand deep learning.
- We provide several possible future research directions for using deep learning in geophysics, such as unsupervised learning, federated learning, activate learning.

Overall, we aim to pave the way for more geophysical researchers, students, and teachers to use data-driven techniques.

**General theory**

In geophysics, the goal is to invert unknown parameters \( \mathbf{x} \) from an available dataset \( \mathbf{y} = \mathbf{Lx} \). \( \mathbf{L} \) is a forward or degraded operator in geophysical applications, such as denoising, reconstruction, or full-waveform inversion. However, \( \mathbf{L} \) is usually ill-conditioned or not invertible. Compressive sensing approximates \( \mathbf{x} \) by forming an optimization problem with additional constraints:

\[
\hat{x} = \arg \min_{x} D(\mathbf{Lx}, \mathbf{y}) + R(x)
\]

where \( D \) is a similarity measurement function. Typically, the \( L_2 \)-norm \( \| \mathbf{Lx} - \mathbf{y} \|_2 \) is used for smooth measurement. \( R \) is a regularization term. Sparsity is a popular regularization term adopted in compressive sensing, where \( R(x) = \| \mathbf{Wx} \|_1 \). \( \mathbf{W} \) is a sparse transform with several components. The
goal of dictionary learning is to train an optimized sparse transform $W$, which is used for the sparse representation of $x$. Dictionary learning involves learning $W$ via matrix decomposition with constraints on the dictionary $W$ and coefficient $v$.

$$
\hat{W}, \hat{v} = \arg\min_{W, v} D(W^T v, x) + R_s(W) + R(v)
$$

Unlike dictionary learning, deep learning treats geophysical problems as regression or classification problems. A DNN $F$ is used to approximate $x$,

$$
x = F(y; \Theta)
$$

where $\Theta$ is the parameter set of the DNN. From the view of mathematics, a DNN provides a high-dimensional and nonlinear mapping from $y$ to $x$. From the perspective of biology, the architecture of a DNN is bionic and includes biological nerve cells in a multilayer structure. Each layer contains several neural units (neurons), with input, output, and nonlinear activation features that are analogous to axons, dendrites, and the cell unit. The input and output of a neuron are connected to the neurons in the neighboring layers, and $\Theta$ represents the connection weights. A particular connection format, the convolutional filter, is often used in modern DNNs to share the parameters among different neurons.

Deep learning aims to build a high-dimension approximation between two sets $X = \{x_i, i = 1 \ldots N\}$ and $Y = \{y_i, i = 1 \ldots N\}$, i.e., the inputs and labels, with a DNN. The approximation is achieved by minimizing the following loss function to obtain an optimized $\Theta$:

$$
E(\Theta; X, Y) = \sum_{i=1}^{N} \| x_i - F(y_i; \Theta) \|^2_2
$$

If $F$ is differentiable, a gradient-based method can be used to optimize $\Theta$. However, a large Jacobi matrix is involved when calculating $\nabla_{\Theta} E$, making it infeasible for large-scale datasets. A back-propagation method (LeCun et al. [10]) is proposed to compute $\nabla_{\Theta} E$ and avoid calculating the Jacobi matrix. In the following sections, we first introduce dictionary learning from the traditional K-means method and the widely used K-SVD approach; a recent fast algorithm with a tight data-driven framework and other deep learning methods are also presented. A mapping diagram of the primary references used in this paper is given in Figure 2.

**Dictionary learning**

**K-means**

Clustering is used to group geophysical attributes into several classifications. For example, we need to decide whether a region contains fluvial facies or faults based on stacked sections. K-means (Hartigan and Wong [11]), which is a classical clustering algorithm, can be treated as a dictionary learning method with an extremely sparse representation, where only one dictionary component is allowed, and the representation coefficient must be one. Though simple, we can briefly explore the basic steps in dictionary learning with this approach.
Figure 2. Conceptual map of the data-driven methods included in this paper.
K-means aims to cluster \( N \) given samples with \( M \) features into \( K \) groups. K-means applies two steps per iteration with \( K \) randomly initialized cluster centers. i) Assign the training samples to the nearest cluster center. ii) Update each cluster center based on the weighted center of the attached samples. Figure 3 shows an example of how K-means splits a dataset into two classes based on two selected features.

![Figure 3](image.png)

Figure 3. Illustration of the K-means method. Left: A randomly generated dataset with 300 samples \( (N=300) \) and two features \( (M=2) \). Right: The classification result \( (K=2) \).

K-means, and the corresponding improved methods, is used for signal classification in geophysics. Because K-means is sensitive to feature selection, Galvis et al. [12] suggested that seismic attributes be selected based on the notion of similarity and that these attributes can be used in the classification of surface waves with the K-means approach. K-means is also sensitive to outliers. Song et al. [13] propose an adaptive-phase K-means method. The advantage of using the phase distance as a similarity measure is that it provides robustness in the presence of horizon error. The classification result of K-means depends on the user-specified number of clusters. Waheed et al. [14] showed that the density-based spatial clustering method does not require a specification for the number of clusters and reduces the cost of automatic velocity selection compared to that in the traditional K-means approach. De Lima and Marfurt [15] proposed a combination of PCA and K-means for the classification of airborne gamma ray spectrometry.

**K-SVD**

Similar to K-means, most dictionary learning algorithms consist of two steps: i) sparse coding and ii) dictionary updating. Unlike K-means, the sparse representation is not limited to one component, and the number of representation coefficients is also not limited. The method of optimal directions (MOD) (Engan et al. [16]) uses orthogonal matching pursuit for sparse coding and a second-order Newtonian method for dictionary updating. Beckouche and Ma [17] use the MOD method for dictionary learning and sparse approximation in seismic denoising. The dictionary updating approach in MOD has favorable flexibility and simplicity; however, it is relatively impractical for large dictionaries since matrix inversion is involved.

K-SVD (where SVD is singular value decomposition) (Aharon et al. [4]) shares the same sparse coding structure as MOD, but several improvements in dictionary updating are given. First, one component of the dictionary is updated at a given time, and the remaining terms are fixed. Second, a rank-1 approximation SVD algorithm is used to obtain the updated dictionary and coefficients simultaneously, thereby accelerating convergence and reducing computational memory use compared to those in the MOD. K-SVD is applied in geophysics with preferred extensions. Nazari Siahsar et al. [18] split the training data into different slices and trained different dictionaries with a shared sparse coefficient matrix. Such a strategy allows 3-D datasets to processed with a reasonable...
time cost for all training patches.

**Data-driven tight frame**

Despite the success of K-SVD in signal enhancement and compression, dictionary updating is still time consuming in regard to high-dimensional and large-scale datasets, such as 3-D prestacked data in seismic exploration. K-SVD includes one SVD step to update one dictionary term. Can the entire dictionary be updated by one SVD for efficient improvement? Cai et al. [19] proposed a data-driven tight frame (DDTF) by enforcing a tight frame constraint on the dictionary. The tight frame condition is a slightly weaker condition than orthogonality, for which the perfect reconstruction property holds. With the tight frame property, dictionary updating in DDTF is achieved with one SVD, which is hundreds of times faster than K-SVD.

Liang et al. [20] first utilized 2-D DDTF in seismic data interpolation. The extension of dictionary learning to high dimensions is straightforward since the data are vectorized at the patch scale (patches are blocks generated from original data division into training samples). Yu et al. [21] extend DDTF to 3-D and 5-D with applications in seismic data interpolation. The training patches for dictionary learning are a random subset of all patches. An example of a learned dictionary with 3-D DDTF for a seismic volume is shown in Figure 4. Yu et al. [22] designed a Monte Carlo selection method based on a training set. The patches with high variance were selected with high probability to further improve efficiency. Liu et al. [23] proposed tensor DDTF, in which high-dimensional data are obtained by tensor products, to save computational resources and constrain data structures. Liu et al. [24] and Liu and Ma [25] proposed graph DDTF, in which a binary tree is used to cluster training patches. DDTF is implemented for each cluster to obtain a sparse dictionary with similar patches as the original dictionary. Wang and Ma [26] and Wang et al. [27] proposed adaptive DDTF and group-sparsity DDTF for the preservation of weak signals by considering the similarity among different patches.

![Figure 4](image.png)

**Figure 4.** An illustration of DDTF. The dictionary is initialized with a spline framelet. After training based on a post-stack seismic dataset, the trained dictionary exhibits apparent structures.

**From dictionary learning to deep learning**

Though both are data-driven methods, deep learning differs from dictionary learning in three aspects: the depth of decomposition, the amount of training data, and the nonlinear operators. Dictionary learning is usually a single-level matrix decomposition problem. Rubinstein et al. [28]
proposed double sparsity (DS) dictionary learning to explore deep decomposition. The motivation of DS is that the learned dictionary atoms still share some underlying sparse pattern for a generic dictionary. In other words, the dictionary is represented with a sparse coefficient matrix multiplied by a fixed dictionary, as in discrete cosine transform. Inspired by DS dictionary learning, can we propose triple, quadruple or even centuple dictionary learning? We know cascading linear operators are equivalent to a single linear operator. Therefore, using more than one fixed dictionary does not improve the signal representation ability compared to that ability of one fixed dictionary if no additional constraints are provided. In deep learning, nonlinear operators are combined in such a deep structure. A neural network with one hidden layer and nonlinear operators can represent any complex function with a sufficient number of hidden neurons. To fit a neural network with many hidden neurons, we need an extensive training set (notably, dictionary learning involves only one target data). A comparison of the learned features in dictionary learning and deep learning is shown in Figure 5.

![Figure 5. Comparison of the learned features in dictionary learning and deep learning. Dictionary learning obtains single-level decomposed features. Deep learning captures multilevel decomposed features.](image)

Deep learning is a machine learning method based on a DNN, which is an artificial neural network (ANN) with many layers. ANNs are widely in machine learning and date from the late 1940s. In a multilayer perceptron (MLP), a specific type of ANN, neurons are organized into different groups called layers. Neurons in adjacent layers are connected by connection weights. The output of a neuron is the weighted summation of the output of the neurons from the previous layers, and each output is then input into a nonlinear activation function. The simplest MLP includes an input layer, a hidden layer, and an output layer.

Poulton [29] published a review article on ANN methods in geophysics in 2000. Since 2000, many pioneers have applied machine learning methods in geophysics, and deep learning has slowly become popular. Limited by the length of this review, we only recall some such studies. Lim [30] characterized reservoir properties using fuzzy logic and an ANN for well data. Huang et al. [31] explored seismic data parameter determination and pattern detection with an ANN. Helmy et al. [32] applied hybrid computational models to characterize oil and gas reservoirs. Zhang et al. [33] proposed using a kernel-regularized least-squares (Evgeniou et al. [34]) method for fault detection from seismic records. Jia and Ma [35] suggest using supported vector regression (Cortes and Vapnik [36]) for seismic interpolation. The authors used linearly interpolated data and original data as the
inputs and outputs, respectively, and supported vector regression was used to obtain the relation between the inputs and outputs. They claimed that no assumptions were imposed on the data and that no parameter tuning was required for interpolation.

Our review will focus on DNNs. The number of layers in an ANN has a significant effect on the fitting and generalization abilities of an algorithm. Early ANNs were restricted to a few layers due to the computational capacity of the available hardware. With the development of hardware and optimization algorithms, ANNs are expanding towards deep layers. However, MLPs encounter computational and storage problems due to the massive number of parameters required when the network deepens or the size of the inputs increases in practical applications. In addition, an MLP requires preselected features as inputs into the neural network and ignores the structure of the input entirely, with full reliance on experience. Qi et al. [37] proposed feature selection for machine learning facies analysis. An exhaustive search method, which took 638 hours for the candidate attributes, was used to determine both the optimal number and combination of parameters. To reduce the number of parameters in an MLP, convolutional neural networks (CNNs) (McCann et al. [38]) were proposed to share parameters with convolutional filters.

**Convolutional neural networks**

CNNs were proposed to consider local coherency and reduce the number of weight parameters. CNN uses convolutional filters to restrict the inputs of a neural network to within a local range. The convolutional filters are shared by different neurons in the same layer. A CNN uses original data rather than selected features as an input set. Pooling layers are used in CNNs to extract key features by subsampling the input set. CNNs have developed rapidly since 2010 for image classification and segmentation, and some popular CNNs include VGGNet (Simonyan and Zisserman [39]) and AlexNet (Krizhevsky et al. [40]). CNNs are also used in image denoising (Zhang et al. [41]) and super-resolution tasks (Dong et al. [42]). The above CNNs are named vanilla CNNs, which are CNNs with simple sequential structures. Vanilla CNNs are used for regression and classification tasks. In regression tasks, the outputs are continuous variables; in classification tasks, the outputs are discrete variables. Compared to K-means, deep learning achieves complex classification with a multilayer feature extractor and simple classifiers.

Additional deep learning network architectures have been proposed for specific tasks based on MLPs or vanilla CNNs (Figure 6a,b). An autoencoder (AE) is a network in which the inputs and outputs are the same. Hidden layers in AEs extract deep features that are typically used for unsupervised classification with the help of K-means. A deep convolutional autoencoder (CAE, Figure 6c) is an AE with convolutional layers that acts as a feature extractor. U-Net (Ronneberger et al. [43]) (Figure 6d) uses skip connections to bring low-level features to a high level. The generative adversarial network (Goodfellow et al. [44], Creswell et al. [45]) (GAN, Figure 6e) aims to reproduce data examples with the same distribution as the training set. A GAN contains a generative network and a discriminative network. The generative network tries to produce a nearly real image. The discriminative network tries to distinguish whether the input image is real or generated (fake). Therefore, such a game will finally allow the generative network to produce fake images that the discriminative network cannot distinguish from real images. CycleGAN (Zhu et al. [46]) is a GAN with two generative networks and two discriminative networks, such that a cycle mapping between two datasets is trained. Recurrent neural networks (RNNs, Figure 6f) are commonly used for prediction tasks based on sequential data, and the prediction for the current input
depends on the history of inputs fed into the neural network. Long short-term memory (LSTM) (Hochreiter and Schmidhuber [47]) is a widely used RNN that considers how much historical information is forgotten or remembered. We introduce the concepts and applications of the above DNNs in detail in the following sections.

![Figure 6. Sketches of DNNs. We omit the details of the layers and maintain the shape of each network architecture. The blue lines indicate inputs, and the orange lines indicate outputs, the length of which represents the data dimension. The green lines indicate intermedia connections. Convolutional layers usually have the same size as the input and output, and pooling/unpooling layers will reduce/expand the data set size. (a) In regression tasks, such as denoising or interpolation, the output often has the same dimension as the input. (b) In classification tasks, the outputs are labels with a relatively small dimension. (c) The dimension of the latent feature space in the CAE is lower than that of the data space. (d) Skip connections are used to bring the low-level features to a high level in U-Net. (e) In a GAN, low-dimensional random vectors are used to generate a sample from the generator, and then the sample is classified as true or false by the discriminator. (f) In an RNN, the output of the network is used as input in a cycle.](image)

**Vanilla convolutional neural networks**

Vanilla CNNs are the most popular CNNs if many training samples and labels are available. Cascading convolutional layers, nonlinear layers, and data regularization layers provide a remarkable fit for the training samples in regression tasks. Pooling layers are used for feature extraction in classification tasks. Vanilla CNNs are reliable for most applications in geophysics, such as denoising, interpolation, velocity modeling, data interpretation, etc.

In the seismic denoising area, Yu et al. [48] proposed a denoising CNN (DnCNN) (Zhang et al. [41])-based method for three kinds of seismic noise. The DnCNN developed was composed of convolutional, batch normalization, and rectified linear unit layers. In this approach, the final output is equal to the network output plus the input, which is called residual learning, i.e., the output of the network represents noise. The concept of residual learning is similar to that used in a residual network (e.g., ResNet) (He et al. [49]) to avoid vanishing gradients. Random and linear noise are manually added to synthetic datasets, and multiple data sets are generated with the acoustic wave equation. In this case, transfer learning (Donahue et al. [50]) is used for field data denoising.
Different kinds of noise are processed with the same network architecture but different training sets. An example of scattered ground-roll attenuation is shown in Figure 7. Scattered ground roll is mainly observed in desert area, caused by the scattering of ground roll when the near surface is laterally heterogeneous. Scattered ground roll is difficult to remove because it occupies the same F-K domain as reflected signals. DnCNN was used to remove scattered ground roll successfully. Wu et al. [51] claimed that in a traditional CNN, the labels of clean data are difficult to obtain. They used multiple trials involving user-generated white noise to simulate real white noise. Additionally, the inputs were decomposed with the variational mode decomposition method (Dragomiretskiy and Zosso [52]) to obtain a few modes with different frequency supports, which were then fed into a CNN.

In the seismic interpolation field, Wang et al. [53] proposed the use of ResNet for the reconstruction of regularly missing data. The training set consisted of synthetic and field samples. The input of the network was preprocessed with a bicubic interpolation algorithm. Zhang et al. [54] trained a denoised neural network with a natural image dataset and used the trained network in the project onto a convex set (POCS) (Abma and Kabir [55]) framework for seismic data interpolation. Therefore, no new networks were required for the interpolation of other datasets or other tasks. Figure 8 gives the training set and a simple interpolation result from Zhang et al. [54].

Figure 7. Deep learning for scattered ground-roll attenuation (Yu et al. [48]). On the left is the original noisy dataset. On the right is the denoised dataset. The scattered ground roll marked by the green arrows are removed.
Figure 8. The training set and interpolation result from Zhang et al. [54]. (a) A subset of the natural image dataset. The natural image dataset was used to train a network for seismic data interpolation. (b) An under-sampled seismic record. (c) The interpolated record corresponding to (b). The regions 1.6-1.88 s and 1.0-1.375 km are enlarged at the top-right corner.

In seismic deblending, Zu et al. [56] constructed an end-to-end deblending CNN. The trained network was iteratively applied to blended data. The authors claimed that networks trained with both synthetic and field datasets perform well with real input datasets. Sun et al. [57] also used an end-to-end CNN for deblending. Different hyper-parameters, such as the number of layers, number of filters, and size of the filters, were used to construct a network that was optimal for seismic data. They used field datasets to construct the training set, which inevitably contained some noise contamination associated with the labels. Nakayama et al. [58] presented a method for designing acquisition parameters, including blending, source, and receiver positions, based on a genetic algorithm and CNN. The genetic algorithm was used to produce combinations of acquisition parameters, and the CNN was used to classify the combinations. Finally, a deblending algorithm was used to obtain a clean signal. If this signal was close to the original signal, the iteration stopped; if not, the algorithm proceeded with new parameters.

In velocity analysis and inversion, Arayapol et al. [59] use a vanilla DNN with an MLP structure for seismic tomography and obtained a promising result for synthetic 2D data. Wang and Ma [60] proposed a network combined with fully connected layers and fully convolutional layers (FCN). The FCN used a contracting path (encoder) and an expansive path (decoder) corresponding to feature extraction and function fitting, respectively. The FCN yielded outputs with the same size as the inputs. The input was a seismology data set with a cross-well geometry, and the output was a velocity model. Notably, the authors used smoothed natural images as seismic models, thus producing a large number of models to construct the training set. Figure 9 shows how Wang and Ma [60] converted a three-channel color image to a velocity model. Park and Sacchi [61] developed a CNN to directly estimate stacking velocities. The portions with different time slices as channels were used as inputs. The root square velocity was the output. For a much different dataset, the authors used transfer learning instead of network training from random initialization. Ovcharenko et al. [62] proposed a DL framework for extrapolating the frequency range of seismic data from high to low frequencies. The inputs and outputs of the network were multiple high-frequency and single-frequency seismic data.
low-frequency representations of a shot gather. Moreover, 0.25 Hz data were obtained from 2 to 4.5 Hz frequencies. Low-frequency information was used for the initialization of full-waveform inversion (FWI).

Figure 9. Converting a three-channel color image into a velocity model (Wang and Ma [60]). (a)-(c) are original color image, gray scale image, and corresponding velocity model. (d) is the seismic record generated from a cross-well geometry on (c).

In the attribute inversion area, Das et al. [63] proposed a 1-D CNN for seismic impedance inversion. The training set consisted of synthetic datasets and contained six output features, including the spherical variogram ranges of facies, phases, the central frequency, etc. The uncertainty was computed with an approximate Bayesian computational method. You et al. [64] predicted anisotropy information from conventional well logs based on a DNN, and the method was generalized for use with field data.

Convolutional autoencoder

A CAE is a type of CNN consisting of an encoder and a decoder. The encoder uses convolutional layers and pooling layers to extract critical features in a latent space from the inputs, resulting in a contracting path. The decoder uses deconvolutional layers and unpooling layers to decode the features into the original data space, resulting in an expanding path. A CAE works in both supervised and unsupervised ways. If labels are provided as outputs, a CAE is a supervised regression network. If the outputs are the same as the inputs, a CAE works in an unsupervised way, and the latent features are used for other tasks, such as clustering. The learned latent features can also be used for dimension reduction in large-scale tasks.

In seismic data processing, Wang et al. [65] proposed a CAE-based interpolation method for irregular sampling. The subsampled dataset is the input, and the complete dataset is the output. Transfer learning is used when the method is applied to field data. Wu et al. [66] treated first-arrival selection as an image segmentation problem with a CAE. Anything prior to the first arrival is set to zero, and all instances after the first arrival are set to one. This method works well for noisy situations and field datasets. For the training set, a subset of traces generated with a simple model is used. Gao et al. [67] used CAE for dimension reduction in FWI to estimate longwave information, where the latent parameters were optimized instead of the whole dataset.

In attribute analysis, Duan et al. [68] used a CAE to extract the features of 1D data and K-means for clustering. They used KL divergence to measure the similarities between the two distributions. He et al. [69] and Qian et al. [70] used an AE to extract seismic features in an
unsupervised way, and then a K-means clustering method was used to classify the seismic facies. Qian et al. [70] built a physical model and used field data from the Liziba survey to test the proposed method.

**U-Net**

U-Nets have U-shaped structures and skip connections. The skip connections bring low-level features to high levels. U-Net was first proposed for image segmentation and has been applied in seismic data processing, inversion, and interpretation. The U-structure with a contracting path and expanding path makes every data point in the output contain all information from the input, such that the approach is suitable for mapping data in different domains, such as inverting velocity from seismic records in FWI. The input size of the test set must be the same as that in the training set for a trained U-Net.

In seismic data processing and inversion, Mandelli et al. [71] used a U-Net for the interpolation of seismic data, and prestack images with and without missing traces were used as inputs and outputs. Hu et al. [72] proposed a U-Net-based first-arrival selection method by formulating a binary segmentation problem. Yang and Ma [73] proposed a new general velocity model construction method based on U-Net. The inputs were seismology data sets generated by the acoustic wave equation from surface survey, and labels were the velocity models. This method is useful for generating low-frequency models for the initialization of traditional FWI. Low-frequency information helps FWI converge. Figure 10 shows the velocity prediction results from Yang and Ma [73]. Unlike the conventional inversion method based on physical models, supervised deep learning methods are based on big-data training rather than prior-knowledge assumptions. Unlike in FWI, after network training is completed, the reconstruction costs are negligible. Moreover, little human intervention is needed, no initial velocities are involved, and no cycle-skipping problem exists. Instead of regression from seismic records to velocity models, Zhang and Alkhalifah [74] used deep learning to estimate the distribution of facies from the results of conventional FWI, and the facies were used to constrain a new iteration of the FWI approach several times. In this method, deep learning was integrated into FWI as part of a model constraint.
Figure 10. Predicting the velocity model with U-Net from raw seismological data (Yang and Ma [73]). The columns indicate different velocity models. From top to bottom are the ground truth velocity models, generated seismic records from one shot, and the predicted velocity models.

In seismic interpretation, Wu et al. [75] built an approximately realistic 3-D training dataset by randomly choosing folding and faulting parameters in a reasonable range. Then, the dataset was used to train a 3D U-Net for seismic structural interpretation for features such as faults, layers, and dips in field datasets. Building realistic synthetic datasets rather than handcrafted field datasets is more efficient and can produce similar results. Wu et al. [76] used an end-to-end U-Net for 3D seismic fault segmentation. The inputs were seismic images, and the outputs were ones, indicating faults, and zeros, indicating nonfaults. A class-balanced binary cross-entropy loss function was used to adjust the data imbalance so that the network was not trained to predict only zeros. A network trained only on synthetic data worked well when field datasets were considered. Wu et al. [77] treated the horizon interpretation problem as an image classification task; they used U-Net and post-stack traces located on a user-defined coarse grid as the inputs and manually picked horizons as labels. This approach is semi-automated because the horizons must be labeled for the coarse grid. The traces were processed individually. An example of synthetic post-stack image and field data fault analysis is shown in Figure 11, as published by Wu et al. [75].

![Figure 11](image-url)

Figure 11. (a) A post-stack dataset. (b) Prediction result of (a). (c) A synthetic dataset (Wu et al. [75]).

It is convenient to apply U-Net in various geophysical applications. Here, we give two examples: velocity picking and first-arrival picking. Figure 12 shows the results of using U-Net for velocity picking. The inputs are seismological data, and the outputs are ones where the picks are located and zeros elsewhere. Figure 13 shows the results of the phase picking based on U-Net. We used 8000 synthetic samples. A gradient constraint was added in the loss function to enhance the continuity of the picked positions. Seismological data sets were used as inputs. For the output, three classifications were set: zeros above the first arrival, ones below the first arrival, and twos for the first arrival. The training dataset was contaminated with strong noise and had missing traces. The predicted picking were close to the labels. First-arrival picking based on deep learning has been applied for realistic seismic data processing by using different neural networks (Hu et al. [72], Wu et al. [66]).
Figure 12. Velocity picking based on U-Net. The inputs are seismological data on the left. The outputs are the picking positions on the right. GT means ground truth. PD_REG and PD_CLS represent the predictions of the regression network and classification network, respectively.

Figure 13. Phase picking based on U-Net. The inputs are seismological data. The outputs are zeros above the first arrival (green area), ones below the first arrival (yellow area), and twos for the first arrival (blue line). The green line indicates the predicted first arrival. This experiment was performed based on the modified code from https://github.com/DaloroAT/first_break_picking.

Generative adversarial networks

GANs can be applied in adversarial training for two CNNs: one generator to produce a fake image and one discriminator to distinguish the produced image from real images. When training the discriminator, the real dataset and generated dataset correspond to labels 1 and 0, respectively. Additionally, when the generator is trained, all datasets correspond to the label 1. A GAN is used to generate samples with similar distributions as the training set. The generated samples are used for simulating realistic scenarios or expanding the training set. Zhu et al. [46] proposed an extended GAN, named CycleGAN, with two generators and two discriminators for signal processing. In CycleGAN, a two-way mapping is trained for mapping two datasets from one to the other. The training set CycleGAN is not necessarily paired, as in a vanilla CNN, which makes it relatively easy to construct training sets in geophysical applications with few labels.

To artificially expand labeled data sets, Wang et al. [78] proposed the GAN-based model EarthquakeGen. The detection accuracy was greatly improved by performing artificial sampling for the training set. Si et al. [79] proposed the use of CycleGAN for ground-roll attenuation. The training set consisted of synthetic and field-derived seismic data. Zhang et al. [80] proposed a seismic
Enhancement algorithm based on a GAN with time resolution improvements. Lipari et al. [81] used a GAN to map low-quality migrated images to high-quality images and the corresponding reflectivity images. Siahkoohi et al. [82] proposed a GAN to produce a high-quality wavefield from a low-quality wavefield in the context of surface-related multiples, ghosts, and dispersion. The training procedure consisted of initial training and transfer learning. The initial training was performed with datasets from nearby surveys. Transfer learning was performed with a small training set with high-fidelity data from the current dataset. Only two sets with and without high fidelity are needed. Wang et al. [83] proposed a 1D CycleGAN-based impedance inversion algorithm to mitigate the dependence of vanilla CNNs on the amount of labeled seismic data available.

**Recurrent neural networks**

In time-sequenced data processing applications, RNNs use the output of a network as the input of the subsequent process to consider the historical influence. RNNs are used for the prediction of new outputs from a sequential input, such as predicting new words from an input sentence. The prediction accuracy of LSTM increases with the amount of historical information considered. In geophysical applications, RNNs are used for predicting the next sample of a time-sequenced or spatially sequenced dataset. RNNs are also used for wavefield simulation based on a time-dependent network form.

In seismic data processing, Payani et al. [84] used an RNN to estimate the relationships among samples in a seismic trace; they found that 16 bits are needed for lossless representation instead of 32 bits per sample. Chen et al. [85] applied LSTM for the denoising of magnetotelluric data with prediction data samples in a trace. Li et al. [86] utilized an RNN to consider the spatial continuity and similarity of adjacent traces in facies analysis.

In seismic modeling and inversion, Sun et al. [87] constructed an RNN for wave modeling and inversion, and the network parameters corresponded to the selected velocity model. The structure of an RNN is similar to finite different time evolution. Therefore, optimizing an RNN is equivalent to seismic waveform inversion. Experiments with various optimization algorithms, including gradient descent, conjugate gradient, adaptive moment, and limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithms, have been performed. The results have indicated that first-order methods perform better than second-order methods. Liu [88] extends RNN for simultaneous inversion of velocity and density. Figure 14 shows the structure of a modified RNN based on the acoustic wave equation used in Liu [88]. The diagram represents the discretized wave equation with a flow chart implemented in an RNN. The inversion process in full waveform inversion is the training process of RNN. Fabien-Ouellet and Sarkar [89] combined three networks: a CNN with a CMP gather input and a semblance output, an RNN for data reduction, and LSTM for velocity decoding. This design was inspired by the information flow in semblance analysis. The proposed method works well for 1D layered velocity models. We give an example of using an RNN for simultaneous velocity and density inversion.
Figure 14. Modified RNN based on the acoustic wave equation for wave modeling (Liu [88]). The diagram represents the discretized wave equation implemented in an RNN. The auto-differential mechanics of a DNN help to efficiently optimize the velocity and density.

Earthquake-related works

The goal of earthquake data processing is quite different from that of exploration geophysics, therefore, this section focuses on deep learning-based earthquake signal processing. The preliminary processing of earthquake signals includes classification, to distinguish real earthquakes from noise, and arrival picking, to identify the arrival times of primary and secondary waves. Further applications involve earthquake early warning (EEW) analysis, Earth tomography, etc. Deep learning has shown promising results in these applications.

- Classification

Meier et al. [90] trained five DNNs for seismic signal and noise discrimination. The training set contained 374 thousand quake and 946 thousand noise records for three channels. Li et al. [91] trained machine learning algorithms on an extensive dataset to discriminate earthquake P waves from impulsive local noise. Linville et al. [92] used an RNN and a CNN to identify events as either quarry blasts or earthquakes. The purpose of volcano seismic detection is similar to that of EEW, i.e., deciding whether an event is dangerous. Malfante et al. [93] extracted 102 features from acoustic and seismic fields. Six classes were considered: long-period events, volcanic tremors, volcano-tectonic events, explosions, hybrid events, and tornados.

We provide an example of using the wavelet scattering transform (WST) (Mallat [94]) and a support vector machine for earthquake classification with a limited number of training samples. WST involves a cascade of wavelet transform, a module operator, and an averaging operator, corresponding to convolutional filters, a nonlinear operator, and a pooling operator in a CNN. The critical difference between WST and a CNN is that the filters are predesigned with wavelet transform in WST. In our case, only 100 records were used for training, and 2000 records were used for testing. We obtained a classification accuracy as high as 93% with the WST method. Figure 15 shows the architecture of WST.
Figure 15. (a) The architecture of WST. Unlike in a CNN, the outputs of WST are combined with the outputs of each layer. Then, the outputs of WST serve as features for a classifier.

- **Arrival picking**

  Wang et al. [95] proposed Picknet to choose natural seismic arrivals based on a deep residual network. The selected arrivals were used for seismic tomography, and the underground structure of Japan was reconstructed. Ross et al. [96] trained a CNN for arrival picking and polarity classification. The training set contained 19.4 million seismograms. They achieved remarkably high picking and classification accuracies close to or better than those obtained by human experts. Zhao et al. [97] proposed using U-Net for P- and S-phase arrival picking and achieved superior results compared to those of the short time average over long time average (STA/LTA) method. Zhou et al. [98] developed a hybrid event detection and phase-picking algorithm with both CNNs and RNNs.

- **Further applications**

  Zhang et al. [99] used a CNN to locate seismic sources from received waveforms at several stations. This method worked well for small earthquakes (M_L<3.0) with low SNRs, for which traditional methods fail. The prediction results and errors of earthquake source locations are indicated in Figure 16. Ross et al. [100] generated millions of synthetic sequences to train the PhaseLink network. Associating seismic phases involves classifying seismological records from the same source into one set. This concept is similar to seismic registration. Zhang and Curtis [101] used variational inference for seismic tomography. This method uses the mean value and variance as outputs. Yamaga and Mitsui [102] analyzed the relationship between a strong earthquake and postseismic deformation. The dataset was obtained with the global navigation satellite system and was relatively small, with 153 training points and 38 testing points. An RNN was used to learn the corresponding relationships, and the results were far more accurate than those of traditional regression methods. Rouet-LeDuc et al. [103] used random forest to predict earthquake in laboratory. They used acoustical signals as inputs and predicted laboratory fault failures. Machine learning identified a signal emitted from the fault zone previously thought to be low-amplitude noise.
Applications beyond exploration geophysics and earthquake

More deep learning applications in geophysics beyond exploration geophysics and earthquake are briefly introduced, such as in volcanology, climatology, atmospheric science, geodynamics, pedology, planetary science, and marine geophysics. Anantrasirichai et al. [104] used a CNN to classify interferometric fringes in wrapped interferograms with no atmospheric corrections for detecting volcanic deformation. Kadow et al. [105] chose a CNN-based inpainting algorithm to reconstruct missing values in global climate datasets like HadCRUT4. Li et al. [106] used a DNN to fuse satellite observations and station measurements for estimating ground-level PM2.5. Shahnas et al. [107] predicted mantle flow processes by employing support vector machine algorithms and training samples from numerical convection models. The authors claimed the proposed technique could be extended to more complex geodynamic problems by employing deep learning algorithms. Fang et al. [108] used LSTM to predict historical soil moisture with high fidelity from recent two years of satellite data, showing LSTM’s potential for hindcasting, data assimilation, and weather forecasting. Ruhunusiri et al. [109] developed an DNN to infer solar wind proxies at Mars using sheath measurements. Seven solar wind parameters were inferred simultaneously using spacecraft measurements. Clausen and Nickisch [110] proposed to classify auroral images with DNN. The authors used manually labeled images with six classes. Liu et al. [111] used U-Net to predict coastal inundation mapping from synthetic aperture radar imagery information, providing a better understanding of the geospatial and temporal characteristics of coastal flooding.

Revisiting the relationship between dictionary learning and deep learning

In this section, we discuss the relationship between dictionary learning and two specific deep learning methods, the deep image prior (DIP) (Lempitsky et al. [112]) and AE methods. The dictionary learning model in equation (1) is written in a specific form with a sparse constraint on the coefficient:
\[ W, C = \arg \min_{W,C} \left\| W^T C - X \right\|_F + \alpha \left\| C \right\|_1 \]

where \( \alpha \) is a weight parameter. Lempitsky et al. [112] proposed DIP, which uses a DNN with random inputs for regularization compared to traditional regularization. DIP uses the target data as the only training samples. Mathematically, DIP replaces the dictionary and the sparse constraint with a single, deep U-Net-based generator.

\[ \hat{\Theta} = \arg \min_{\Theta} \left\| \text{Generator}(v; \Theta) - X \right\|_{W^T \text{and sparsity}} \]

After obtaining an optimized \( \hat{\Theta} \), another round of forwarding generator propagation will produce a regularized result. In the AE, sparsity is achieved with an encoder that outputs a low-dimensional vector. A decoder corresponds to \( W^T \) in dictionary learning,

\[ \hat{\Theta}_E, \hat{\Theta}_D = \arg \min_{\Theta_E, \Theta_D} \sum_i \left\| \text{Decoder}_{W^T \text{C with sparsity}} \left( \text{Encoder}_{W^T}(X_i; \Theta_E); \Theta_D \right) - X_i \right\|_2^2 \]

where \( \Theta_E \) and \( \Theta_D \) are the parameters of the encoder and decoder, respectively. Figure 17 shows diagrams of the use of dictionary learning, DIP, and an AE for geophysical signal processing.
Figure 17. Comparison of dictionary learning and deep learning. (a) Dictionary learning has a shallow structure and is unsupervised and linear. A sparsity constraint is placed on the coefficients. (b) In DIP, the network architecture constrains the produced image. (c) An autoencoder has a deep structure, an extensive training set, and nonlinear operators.

**A deep learning tutorial for beginners**

**A coding example of a DnCNN**

The implementation of deep learning algorithms in geophysical data processing is quite simple based on existing frameworks, such as Caffe (https://github.com/BVLC/caffe), Pytorch (https://github.com/pytorch/pytorch), Keras (https://github.com/keras-team/keras) and TensorFlow (https://github.com/tensorflow/tensorflow). Here, we provide an example of how to use Python and Keras to construct a DnCNN for seismic denoising. The code requires 12 lines for dataset loading, model construction, training, and testing. The dataset is preconstructed and includes a clean subset and a noisy subset; the overall dataset includes 12800 samples x 64 x 64 x 1 (data shape) (the data can be downloaded from https://drive.google.com/file/d/1AZAi78TXb6Njazgm5jNvSvsfs4UhtNnt/view?usp=sharing).

```python
1. import h5py
2. from tensorflow.keras.layers import Input, Conv2D, BatchNormalization, ReLU, Subtract
3. from tensorflow.keras.models import Model
4. ftrain = h5py.File('noise_dataset.h5', 'r')
5. X, Y = ftrain['/X'][:], ftrain['/Y'][:]
6. input = Input(shape=(None, None, 1))
7. x = Conv2D(64, 3, padding='same', activation='relu')(input)
8. for i in range(15):
7. x = Conv2D(64, 3, padding='same', use_bias=False)(x)
10. x = ReLU()(BatchNormalization(axis=3, momentum=0.9, epsilon=0.001)(x))
11. x = Conv2D(1, 3, padding='same', use_bias=False)(x)
12. model = Model(inputs=input, outputs=Subtract()([input, x]))
13. model.compile(optimizer="rmsprop", loss="mean_squared_error")
14. model.fit(X[:-1000], Y[:-1000], batch_size=32, epochs=50, shuffle=True)
15. Y_ = model.predict(X[:-1000])
```

Any appropriate plotting tool can be used for data visualization. The training takes less than one hour on an NVidia 2080ti GPU. For further implementations, we suggest some public repositories of dictionary learning and deep learning information for interested readers, as listed in Table 1.

Table 1: Open sources of dictionary learning and deep learning methods for geophysical applications
Tips for beginners

We introduce some practical tips for beginners who want to explore deep learning in geophysics from the perspective of the three most critical steps in deep learning: data generation, network construction, and training.

- **Data generation**

  As noted by Poulton [29], ‘training a feed-forward neural network is approximately 10% of the effort involved in an application; deciding on the input and output data coding and creating good training and testing sets is 90% of the work’. In deep learning, we advise that the percentages of the effort for network construction and dataset preparation should be approximately 40% and 60%. First, most deep learning approaches use an original data set as the input, thus reducing coding decision efforts. Second, a wider variety of network architectures and parameters can be used in deep learning compared to those in traditional neural networks. Overall, constructing a proper training set plays a more prominent role in deep learning.

  Synthetic datasets can be effectively used in deep learning, which is advantageous since realistic datasets with labels are difficult to obtain. First, to assess the availability of deep learning in a specific geophysical application, using synthetic datasets is the most convenient method. Second, if a satisfactory result is obtained with synthetic datasets, realistic datasets can be used for network analysis via transfer learning with a few annotated realistic datasets. Third, if the synthetic datasets are sufficiently complicated, i.e., if the most important factors are considered when generating the datasets, the trained network may be able to process realistic datasets directly (Wu et al. [75] and Wu et al. [76]).

  A synthetic training set should be diverse. First, we suggest using an existing synthetic dataset with an open license (such as SEG open data) instead of generating a dataset. For specific tasks, such as FWI, a dataset may need to be generated based on a wave equation. Second, a dataset can be modified to increase the degrees of freedom. For example, noise, missing traces, and faults can be added to clean datasets depending on the considered task. Third, data augmentation can be used to expand a training set, such as via rotation, symmetry, scaling, translation, and other processes.
The goal is to generate extremely large synthetic datasets that are as close to realistic datasets as possible.

To generate realistic datasets, we suggest using existing methods to generate labels that should then be checked by a human. For example, in first-arrival picking, an automatic picking algorithm is used to preprocess the datasets, and the results are then provided to an expert who identifies the outliers. Such a procedure requires a human to check very notation. We also suggest using activate learning (Yoo and Kweon [113]) to provide a semiautomated labeling procedure. First, all datasets with machine annotation are used to train a DNN, and the samples with high predicted uncertainty are required to be manually annotated.

- **Network construction for different tasks**

  Beginners are suggested to use a DnCNN or U-Net for testing. DnCNNs are available for most tasks in which the input and output share the same domain, such as denoising, interpolation, and attribute analysis. The input size of a DnCNN can vary since there are no pooling layers involved. However, each output data point is determined by a local field from the input rather than from the entire input set. Additionally, U-Net contains pooling layers, and all input points are used to determine an output point. U-Nets are available for tasks even when the inputs and outputs are in different domains, such as in FWI. However, the input size of U-Net is fixed once trained.

  Combining a CAE and K-means is suggested for unsupervised clustering tasks, such as attribute classification. We do not suggest CycleGAN for geophysical tasks since the training process is extremely time consuming and the results are not stable. An RNN provides a high-performance framework for time-dependent tasks, such as forward wave modeling and FWI. RNNs are also used for regression and classification tasks involving temporal or spatial sequential datasets, such as in the denoising of a single trace.

  To adjust the hyperparameters of a DNN and optimization algorithms, we suggest using an autoML toolbox (such as Autokeras (https://github.com/keras-team/autokeras)) instead of manually adjusting the values. The basic objective is to search for the best parameter combination within a given sampling range. Such a search is exceptionally time consuming, and a random search strategy may accelerate the tuning process. Moreover, for most applications, the default architecture gives reasonable results.

- **Training, validation, and testing**

  The available dataset should be split into three subsets: one training set, one validation set, and one test set to optimize the network parameters. The proportions are suggested as 60%, 20%, and 20% based on experience. In a classification task, we suggest using one-hot coding in training. The validation set is used to test the network during training. Then, the model with the best validation accuracy is selected rather than the final trained model. If the validation accuracy does not improve during training, an early stopping strategy is suggested to avoid wasted time. Network hyperparameters should be tuned according to the validation accuracy. The validation set is used to guide training, and the test set is used to test the model based on unseen datasets; however, this set should not be used for hyperparameter tuning.
Two commonly seen issues during training are as follows: the validation loss is less than the training loss, and the loss is not a number (NaN). Intuitively, the training loss should be less than the validation loss since the model is trained with a training dataset. Some potential reasons for this issue are as follows: 1. regularization occurs during training but is ignored during validation, such as in the dropout layer; 2. the training loss is obtained by averaging the loss of each batch during an iteration, and the validation loss is obtained based on the loss after one iteration; and 3. the validation set may be less complicated than the training set. The potential reasons for NaN loss are as follows: 1. the learning rate is too high; 2. in an RNN, one should clip the gradient to avoid gradient explosion and 3. zero is used as a divisor, negative values are used in logarithm, or an exponent is assigned too large of a value.

**Future directions for deep learning in geophysics**

Deep learning, as an efficient artificial intelligence technique, is expected to discover geophysical concepts and inherit expert knowledge through machine-assisted mathematical algorithms. Despite the success of neural networks, their use as a tool for practical geophysics is still in its infancy. The main problems include a shortage of training samples, low signal-to-noise ratios, strong nonlinearity, etc. Among these issues, the critical challenge is the lack of training samples in geophysical applications compared to those in other industries. Several advanced deep learning methods have been proposed related to this challenge, such as semi/unsupervised learning, transfer learning, multimodal deep learning, federated learning, and active learning. We suggest that a focused be placed on the subjects below for future research in the coming decade.

**Semi/unsupervised learning**

In practical geophysical applications, obtaining labels for a large dataset is time consuming and can even be infeasible. Therefore, semisupervised or unsupervised learning is required to limit the dependence on labels. Dunham et al. [114] focused on the application of semisupervised learning in a situation in which the available labels were scarce. A self-training-based label propagation method was proposed, and it outperformed supervised learning methods in which unlabeled samples were neglected. Semisupervised learning takes advantage of both labeled and unlabeled datasets. The combination of AE and K-means is an efficient unsupervised learning method (He et al. [69] and Qian et al. [70]). An autoencoder is used to learn low-dimensional latent features in an unsupervised way, and then K-means is used to cluster the latent features.

**Transfer learning**

Usually, we must train one DNN for a specific dataset and a specific task. For example, a DNN may effectively process land data but not marine data, or a DNN may be effective in fault detection but not in facies classification. To increase the reusability of a trained network for different datasets or different tasks, transfer learning (Donahue et al. [50]) is suggested.

In transfer learning with different datasets, the optimized parameters for one dataset can be used as initialization values for learning a new network with another dataset; this process is called fine tuning. Fine tuning is typically much faster and easier than training a network with randomly initialized weights from scratch. In transfer learning involving different tasks, we assume that the extracted features should be the same in different tasks. Therefore, the first layers in a model trained for one task are copied to the new model for another task to reduce the training time. Another benefit of transfer learning is that with a small number of training samples, we can promptly transfer the
learned features to a new task or a new dataset. Diagrams of these two transfer learning methods are shown in Figure 18. Further topics in transfer learning include the relationship between the transferability of features (Yosinski et al. [115]) and the distance between different tasks and different data sets (Oquab et al. [116]).

Figure 18. Diagrams of transfer learning. (a) Transfer learning between different datasets. The parameters of one trained model can be moved to another model as initialization conditions. (b) Transfer learning between different tasks. The first layers of one trained model can be copied to another model.

**Combination of data-driven and model-driven methods**

To combine geophysical mechanics and deep learning, can we combine model-driven and data-driven approaches? Intuitively, such a combination will produce a more precise result than model-driven methods and a more reliable result than data-driven methods. In addition, with an additional physical constraint on deep learning methods, fewer training samples are required to obtain a more generalized prediction than those of traditional methods. Zhang et al. [117] proposed learning a denoising prior with a DNN and replacing the denoiser in the iteration optimization algorithm, such that different tasks use the same denoiser but different models. Raissi et al. [118] proposed a physical informed neural network that combines training data and physical equation constraints for training. Taking wave modeling as an example, the wavefield was represented with a DNN, 

\[ u(x,t) = F(x,t; \Theta) \], such that the acoustic wave equation was:

\[ u_{tt} - \frac{1}{c^2} u_{xx} = F_{uu}(x,t; \Theta) = c^2 \Delta F(x,t; \Theta) \]

The above equation can serve as a constraint while training the DNN. Another discussed deep learning technique, DIP, can be applied in different tasks with physical models. Similar to the idea of DIP, Wu and McMechan [119] showed that a DNN generator can be added to an FWI framework. First, a U-Net-based generator \( F(v; \Theta) \) with random input \( v \) was used to approximate a velocity model \( m \) with high accuracy. Then, \( m = F(v; \Theta) \) was inserted into the FWI objective function:

\[ E_{FWI}(\Theta) = \frac{1}{2} \| P(F(v; \Theta)) - d \|^2 \]

where \( d \) is the seismic record and \( P \) is the forward wavefield propagator. The gradient of \( E_{FWI} \) with respect to network parameters \( \Theta \) is calculated with the chain rule. U-Net is only used for
regularizing the velocity model. After training, one forward propagation of the network will produce a regularized result. We note that data-driven and model-driven methods are not independent; data-driven methods are also used for discovering physical concepts (Iten et al. [120]).

**Multimodal deep learning**

To improve the resolution of inversion, the joint inversion of data from different sources has been a popular topic in recent years (Garofalo et al. [121]). One of the advantages of DNNs is that they can fuse information from multiple inputs. In multimodal deep learning (Ngiam et al. [122], Ramachandram and Taylor [123]), inputs are from different sources, such as seismic data and gravity data. Collecting data from different sources can help relieve the bottleneck of a limited number of training samples. In addition, using multimodal datasets can increase the accuracy and reliability of deep learning methods. Figure 19 shows an illustration of multimodal deep learning.

**Federated learning**

To provide a practical training set in deep learning for geophysical applications, collecting available datasets from different institutes or corporations might be a possible solution. However, data transfer via the internet is time consuming and expensive for large-scale geophysical datasets. In addition, most datasets are protected and cannot be shared. Federated learning was first proposed by Google (Mcmahan et al. [124], Li et al. [125]) to train a DNN with user data from millions of cellphones (clients) without privacy or security issues. The encrypted gradients from different clients are assembled in a central server, thus avoiding data transfer. The server updates the model and distributes information to all clients (Figure 20). In a simple federated learning setting, the clients and the server share the same network architecture. We give a possible example of federated learning in geophysics based on the concept that some corporations do not share the annotations of first arrivals; however, they can benefit from federated learning by training a DNN together for first arrival picking.
Figure 19. An illustration of multimodal deep learning

Figure 20. Federated learning. The clients train the DNN with local datasets and uploads the model gradient to the server. The server aggregates the gradients and updates the global model. Then, the updated model is distributed to all the local clients. Many rounds of training are performed until the model meets a certain accuracy requirement.

**Uncertainty estimation**

One of the remaining questions associated with applying deep learning in geophysics is related to whether the results of deep learning-based model-driven methods with a solid theoretical foundation can be trusted. One trial in drilling may cost millions of dollars. What if a neural network can report high confidence in a prediction? Deep learning with uncertainty analysis was proposed
to assess reliability, such as through Markov chain Monte Carlo (MCMC) (Kortylewski et al. [126]), variational inference (Subedar et al. [127]), and Monte Carlo dropout (Gal and Ghahramani [128]) methods. For example, in Monte Carlo dropout, dropout layers are added to each original layer to simulate a Bernoulli distribution. With multiple realizations of dropout, the results are collected, and the variance is computed as the uncertainty.

Grana et al. [129] assessed the classification accuracy and uncertainty of RNN and MCMC methods. The RNN method yielded higher accuracy but relatively high uncertainty. The MCMC method provided similar accuracy and was robust to uncertainty through the use of prior spatial correlation models. In the RNN, the uncertainty was obtained through multiple runs of the same procedure with different training subsets, but the results were similar in each case. Maiti and Tiwari [130] used a Bayesian NN to predict the boundaries of lithofacies. Bayesian NNs provide low uncertainty compared to traditional deep learning methods. Cao et al. [131] proposed a sequence of fast seismic acquisitions for dispersion curve extraction and inversion for 3-D seismic models with uncertainty estimates using pretrained mixture density networks.

**Active learning**

To train a high-precision model using a small amount of labeled data, active learning is proposed to imitate the self-learning ability of human beings (Yoo and Kweon [113]). An active learning model selects the most useful data based on a sampling strategy for manual annotation and adds this data to the training set; then, the updated dataset is used for the next round of training (Figure 21). One of the sampling strategies is based on the uncertainty principle, i.e., the samples with high uncertainty are selected. Taking fault detection as an example, if a trained network is not sure whether a fault exists at a given location, we can annotate the fault manually and add the sample to the training set.

![Figure 21. An illustration of active learning. We choose samples with high uncertainty and manually annotate them to serve as training samples.](image)

**Summary**

Data-driven methods, especially deep learning methods, have created both opportunities and challenges in geophysical fields. Pioneer researchers have provided a basis for deep learning in geophysics with promising results; more advanced deep learning technologies and more practical problems must now be explored. To close this paper, we summarize a roadmap for applying deep learning in different geophysical tasks based on a three-level approach.

- Traditional methods are time consuming and require intensive human labor and expert
knowledge, such as in first-arrival selection and velocity selection.

- Traditional methods have difficulties and bottlenecks. For example, FWI requires good initial values and high accuracy modeling and suffers from local minimization.
- Traditional methods cannot handle some cases, such as multimodal data fusion and inversion.

With the development of new artificial intelligence models beyond deep learning and advances in research into the infinite possibilities of applying deep learning in geophysics, we can expect intelligent and automatic geophysical exploration, processing, and exploitation without human intervention soon.

Acknowledgments

We would like to thank Sitong Liu, Wenlong Wang, Xinming Wu, Fangshu Yang, Hao Zhang, and Jie Zhang for allowing us to reuse the original figures from their papers.

References

1. S. Spitz (1991). Seismic trace interpolation in the F-X domain. Geophysics. 56(6): 785-794.
2. F. J. Herrmann and G. Hennenfent (2008). Non-parametric seismic data recovery with curvelet frames. Geophysical Journal International. 173(1): 233-248.
3. V. Oropeza and M. Sacchi (2011). Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis. Geophysics. 76(3): V25-V32.
4. M. Aharon, M. Elad and A. Bruckstein (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing. 54(11): 4311-4322.
5. Y. LeCun, Y. Bengio and G. Hinton (2015). Deep learning. Nature. 521(7553): 436-444.
6. E. J. Candes and M. B. Wakin (2008). An introduction to compressive sampling. IEEE Signal Processing Magazine. 25(2): 21-30.
7. K. J. Bergen, P. A. Johnson, M. V. de Hoop and G. C. Beroza (2019). Machine learning for data-driven discovery in solid earth geoscience. Science. 363(6433): 1-10.
8. P. M. R. DeVries, F. Viegas, M. Wattenberg and B. J. Meade (2018). Deep learning of aftershock patterns following large earthquakes. Nature. 560(7720): 632-634.
9. M. Reichstein, G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais and Prabhat (2019). Deep learning and process understanding for data-driven earth system science. Nature. 566(7743): 195-204.
10. Y. LeCun, B. Boser, J. S. Denker, D. Henderson and L. D. Jackel (1997). Handwritten digit recognition with a back-propagation network. Advances in Neural Information Processing Systems. 2(2): 396-404.
11. J. A. Hartigan and M. A. Wong (1979). A k-means clustering algorithm. Journal of The Royal Statistical Society Series C-applied Statistics. 28(1): 100-108.
12. I. S. Galvis, Y. Villa, C. Duarte, D. A. Sierra and W. Agudelo (2017). Seismic attribute selection and clustering to detect and classify surface waves in multicomponent seismic data by using K-means algorithm. Geophysics. 36(3): 239-248.
13. C. Song, Z. Liu, Y. Wang, F. Xu, X. Li and G. Hu (2018). Adaptive phase K-means algorithm for waveform classification. Exploration Geophysics. 49(2): 213-219.
14. U. B. Waheed, S. Alzahrani and S. M. Hanafy (2019). Machine learning algorithms for automatic velocity picking: K-means vs. DBSCAN. SEG Technical Program Expanded Abstracts. 5510-5114.
15. R. P. De Lima and K. J. Marfurt (2018). Principal component analysis and K-means analysis of
airborne gamma-ray spectrometry surveys. SEG Technical Program Expanded Abstracts. 2277-2281.

16. K. Engan, S. O. Aase and J. H. Husoy (2002). Method of optimal directions for frame design. 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258).

17. S. Beckouche and J. Ma (2014). Simultaneous dictionary learning and denoising for seismic data. Geophysics. 79(3): A27-A31.

18. M. A. Nazari Siahsar, S. Gholtashi, A. R. Kahoo, W. Chen and Y. Chen (2017). Data-driven multitask sparse dictionary learning for noise attenuation of 3D seismic data. Geophysics. 82(6): V385-V396.

19. J. Cai, H. Ji, Z. Shen and G. Ye (2014). Data-driven tight frame construction and image denoising. Applied and Computational Harmonic Analysis. 37(1): 89-105.

20. J. Liang, J. Ma and X. Zhang (2014). Seismic data restoration via data-driven tight frame. Geophysics. 79(3): V65-V74.

21. S. Yu, J. Ma, X. Zhang and M. Sacchi (2015). Interpolation and denoising of high-dimensional seismic data by learning a tight frame. Geophysics. 80(5): V119-V132.

22. S. Yu, J. Ma and S. Osher (2016). Monte Carlo data-driven tight frame for seismic data recovery. Geophysics. 81(4): V327-V340.

23. L. Liu, G. Plonka and J. Ma (2017). Seismic data interpolation and denoising by learning a tensor tight frame. Inverse Problems. 33(10): 105011.

24. L. Liu, J. Ma and G. Plonka (2018). Sparse graph-regularized dictionary learning for suppressing random seismic noise. Geophysics. 83(3): V215-V231.

25. L. Liu and J. Ma (2019). Structured graph dictionary learning and application on the seismic denoising. IEEE Transactions on Geoscience and Remote Sensing. 57(4): 1883-1893.

26. X. Wang and J. Ma (2019). Adaptive dictionary learning for blind seismic data denoising. IEEE Geoscience and Remote Sensing Letters. 1-5.

27. X. Wang, B. Wen and J. Ma (2019). Denoising with weak signal preservation by group-sparsity transform learning. Geophysics. 84(6): V351-V368.

28. R. Rubinstein, M. Zibulevsky and M. Elad (2010). Double sparsity: Learning sparse dictionaries for sparse signal approximation. IEEE Transactions on Signal Processing. 58(3): 1553-1564.

29. M. M. Poulton (2002). Neural networks as an intelligence amplification tool: A review of applications. Geophysics. 67(3): 979-993.

30. J. S. Lim (2005). Reservoir properties determination using fuzzy logic and neural networks from well data in offshore korea. Journal of Petroleum Science and Engineering. 49(3-4): 182-192.

31. K. Huang, J. You, K. Chen, H. Lai and A. Don (2006). Neural network for parameters determination and seismic pattern detection. SEG Technical Program Expanded Abstracts. 2285-2289.

32. T. Helmy, A. Fatai and K. Faisal (2010). Hybrid computational models for the characterization of oil and gas reservoirs. Expert Systems with Applications. 37(7): 5353-5363.

33. C. Zhang, C. Froger, M. Araya-Polo and D. Hohl (2014). Machine-learning based automated fault detection in seismic traces. 76th EAGE Conference and Exhibition 2014. 2014(1): 1-5.

34. T. Evgeniou, M. Pontil and T. Poggio (2000). Regularization networks and support vector machines. Advances in Computational Mathematics. 13(1): 1-50.

35. Y. Jia and J. Ma (2017). What can machine learning do for seismic data processing? An interpolation application. Geophysics. 82(3): V163-V177.
36. C. Cortes and V. Vapnik (1995). Support-vector networks. Machine learning. **20**(3): 273-297.
37. J. Qi, B. Zhang, B. Lyu and K. Marfurt (2020). Seismic attribute selection for machine-learning-based facies analysis. Geophysics. **85**(2): O17-O35.
38. M. T. McCann, K. H. Jin and M. Unser (2017). Convolutional neural networks for inverse problems in imaging: A review. IEEE Signal Processing Magazine. **34**(6): 85-95.
39. K. Simonyan and A. Zisserman (2015). Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations.
40. A. Krizhevsky, I. Sutskever and G. E. Hinton (2017). Imagenet classification with deep convolutional neural networks. Communications of the Acm. **60**(6): 84-90.
41. K. Zhang, W. Zuo, Y. Chen, D. Meng and L. Zhang (2017). Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. IEEE Transactions on Image Processing. **26**(7): 3142-3155.
42. C. Dong, C. C. Loy, K. He and X. Tang (2014). Learning a deep convolutional network for image super-resolution. European Conference on Computer Vision. 184-199.
43. O. Ronneberger, P. Fischer and T. Brox (2015). U-net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer Assisted Intervention. 234-241.
44. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio (2014). Generative adversarial nets. Neural Information Processing Systems. 2672-2680.
45. A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta and A. A. Bharath (2018). Generative adversarial networks: An overview. IEEE Signal Processing Magazine. **35**(1): 53-65.
46. J. Zhu, T. Park, P. Isola and A. A. Efros (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. International Conference on Computer Vision. 2242-2251.
47. S. Hochreiter and J. Schmidhuber (1997). Long short-term memory. Neural Comput. **9**(8): 1735-1780.
48. S. Yu, J. Ma and W. Wang (2019). Deep learning for denoising. Geophysics. **84**(6): V333-V350.
49. K. He, X. Zhang, S. Ren and J. Sun (2016). Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition. 770-778.
50. J. Donahue, Y. Jia, O. Vinyals, J. Hoffman and T. Darrell (2014). Decaf: A deep convolutional activation feature for generic visual recognition. International Conference on Machine Learning. 647-655.
51. H. Wu, B. Zhang, T. Lin, F. Li and N. Liu (2019). White noise attenuation of seismic trace by integrating variational mode decomposition with convolutional neural network. Geophysics. **84**(5): V307-V317.
52. K. Dragomiretskiy and D. Zosso (2014). Variational mode decomposition. IEEE Transactions on Signal Processing. **62**(3): 531-544.
53. B. Wang, N. Zhang, W. Lu and J. Wang (2019). Deep-learning-based seismic data interpolation: A preliminary result. Geophysics. **84**(1): V11-V20.
54. H. Zhang, X. Yang and J. Ma (2020). Can learning from natural image denoising be used for seismic data interpolation? Geophysics. **85**(4): WA115-WA136.
55. R. Abma and N. Kabir (2006). 3D interpolation of irregular data with a POCS algorithm. Geophysics. **71**(6): 91-97.
56. S. Zu, J. Cao, S. Qu and Y. Chen (2020). Iterative deblending for simultaneous source data using the deep neural network. Geophysics. **85**(2): V131-V141.
57. J. Sun, S. Slang, T. Elboth, T. Larsen Greiner, S. McDonald and L.-J. Gelius (2020). A convolutional
neural network approach to deblending seismic data. Geophysics. 85(4): WA13-WA26.

58. S. Nakayama, G. Blacquiere and T. Ishiyama (2019). Automated survey design for blended acquisition with irregular spatial sampling via the integration of a metaheuristic and deep learning. Geophysics. 84(4): P47-P60.

59. M. Arayapolo, J. Jennings, A. Adler and T. Dahlke (2018). Deep-learning tomography. The Leading Edge. 37(1): 58-66.

60. W. Wang and J. Ma (2020). Velocity model building in a crosswell acquisition geometry with image-trained artificial neural network. Geophysics. 85(2): U31-U46.

61. M. J. Park and M. D. Sacchi (2019). Automatic velocity analysis using convolutional neural network and transfer learning. Geophysics. 85(1): V33-V43.

62. O. Ovcharenko, V. Kazei, M. Kalita, D. Peter and T. Alkhalifah (2019). Deep learning for low-frequency extrapolation from multioffset seismic data. Geophysics. 84(4): P47-P60.

63. V. Das, A. Pollack, U. Wollner and T. Mekerji (2019). Convolutional neural network for seismic impedance inversion. Geophysics. 84(6): R869-R880.

64. N. You, Y. E. Li and A. Cheng (2020). Shale anisotropy model building based on deep neural networks. Journal of Geophysical Research: Solid Earth. 125(2): e2019JB019042.

65. Y. Wang, B. Wang, N. Tu and J. Geng (2020). Seismic trace interpolation for irregular spatial sampled data using convolutional autoencoder. Geophysics. 85(2): V119-V130.

66. H. Wu, B. Zhang, F. Li and N. Liu (2019). Semiautomatic first-arrival picking of microseismic events by using the pixel-wise convolutional image segmentation method. Geophysics. 84(3): V143-V155.

67. Z. Gao, Z. Pan, J. Gao and Z. Xu (2019). Building long-wavelength velocity for salt structure using stochastic full waveform inversion with deep autoencoder based model reduction. SEG Technical Program Expanded Abstracts. 1680-1684.

68. Y. Duan, X. Zheng, L. Hu and L. Sun (2019). Seismic facies analysis based on deep convolutional embedded clustering. Geophysics. 84(6): IM87-IM97.

69. Y. He, J. Cao, Y. Lu, Y. Gan and S. Lv (2018). Shale seismic facies recognition technology based on sparse autoencoder. International Geophysical Conference. 1744-1748.

70. F. Qian, M. Yin, X. Liu, Y. Wang, C. Lu and G. Hu (2018). Unsupervised seismic facies analysis via deep convolutional autoencoders. Geophysics. 83(3): A39-A43.

71. S. Mandelli, F. Borra, V. Lipari, P. Bestagini, A. Sarti and S. Tubaro (2018). Seismic data interpolation through convolutional autoencoder. SEG Technical Program Expanded Abstracts. 4101-4105.

72. L. Hu, X. Zheng, Y. Duan, X. Yan, Y. Hu and X. Zhang (2019). First-arrival picking with a U-net convolutional network. Geophysics. 84(6): U45-U57.

73. F. Yang and J. Ma (2019). Deep-learning inversion: A next-generation seismic velocity model building method. Geophysics. 84(4): R585-R584.

74. Z. Zhang and T. Alkhalifah (2019). Regularized elastic full-waveform inversion using deep learning. Geophysics. 84(5): R741-R751.

75. X. Wu, Z. Geng, Y. Shi, N. Pham, S. Fomel and G. Caumon (2020). Building realistic structure models to train convolutional neural networks for seismic structural interpretation. Geophysics. 85(4): WA27-WA39.

76. X. Wu, L. Liang, Y. Shi and S. Fomel (2019). FaultSeg3D: Using synthetic data sets to train an end-to-end convolutional neural network for 3D seismic fault segmentation. Geophysics. 84(3): IM35-
77. H. Wu, B. Zhang, T. Lin, D. Cao and Y. Lou (2019). Semiautomated seismic horizon interpretation using the encoder-decoder convolutional neural network. Geophysics. 84(6): B403-B417.

78. T. Wang, Z. Zhang and Y. Li (2019). Earthquakegen: Earthquake generator using generative adversarial networks. SEG Technical Program Expanded Abstracts. 2674-2678.

79. X. Si, Y. Yuan, F. Ping, Y. Zheng and L. Feng (2020). Ground roll attenuation based on conditional and cycle generative adversarial networks. SEG 2019 Workshop: Mathematical Geophysics: Traditional vs Learning. 95-98.

80. H. Zhang, W. Wang, X. Wang, W. Chen and Z. Zhao (2019). An implementation of the seismic resolution enhancing network based on GAN. SEG Technical Program Expanded Abstracts 2019. 2478-2482.

81. V. Lipari, F. Picetti, P. Bestagini and S. Tubaro (2018). A generative adversarial network for seismic imaging applications. SEG Technical Program Expanded Abstracts. 2231-2235.

82. A. Siahkoohi, M. Louboutin and F. J. Herrmann (2019). The importance of transfer learning in seismic modeling and imaging. Geophysics. 84(6): A47-A52.

83. Y. Wang, Q. Ge, W. Lu and X. Yan (2019). Seismic impedance inversion based on cycle-consistent generative adversarial network. SEG Technical Program Expanded Abstracts. 2498-2502.

84. A. Payani, F. Fekri, G. Alregib, M. Mohandes and M. Deriche (2019). Compression of seismic signals via recurrent neural networks: Lossy and lossless algorithms. SEG Technical Program Expanded Abstracts 2019. 4082-4086.

85. H. Chen, R. Guo, J. Liu, Y. Wang and R. Lin (2020). Magnetotelluric data denoising with recurrent neural network. SEG 2019 Workshop: Mathematical Geophysics: Traditional vs Learning. 116-118.

86. L. Li, Y. Lin, X. Zhang, H. Liang, W. Xiong and S. Zhan (2019). Convolutional recurrent neural networks based waveform classification in seismic facies analysis. SEG Technical Program Expanded Abstracts. 2599-2603.

87. J. Sun, Z. Niu, K. A. Innanen, J. Li and D. O. Trad (2020). A theory-guided deep-learning formulation and optimization of seismic waveform inversion. Geophysics. 85(2): R87-R99.

88. S. Liu (2020). Multi-parameter full waveform inversions based on recurrent neural networks. Dissertation for the Master Degree in Science, Harbin Institute of Technology.

89. G. Fabien-Ouellet and R. Sarkar (2019). Seismic velocity estimation: A deep recurrent neural-network approach. Geophysics. 85(1): U21-U29.

90. M. A. Meier, Z. E. Ross, A. Ramachandran, A. Balakrishna, S. Nair, P. Kundzicz, Z. F. Li, J. Andrews, E. Hauksson and Y. S. Yue (2019). Reliable real-time seismic signal/noise discrimination with machine learning. Journal of Geophysical Research-Solid Earth. 124(1): 788-800.

91. Z. Li, M. A. Meier, E. Hauksson, Z. Zhan and J. Andrews (2018). Machine learning seismic wave discrimination: Application to earthquake early warning. Geophysical Research Letters. 45(10): 4773-4779.

92. L. Linville, K. Pankow and T. Draelos (2019). Deep learning models augment analyst decisions for event discrimination. Geophysical Research Letters. 46(7): 3643-3651.

93. M. Malfante, M. Dalla Mura, J. I. Mars, J. P. Metaxian, O. Macedo and A. Inza (2018). Automatic classification of volcano seismic signatures. Journal of Geophysical Research-Solid Earth. 123(12): 10645-10658.

94. S. Mallat (2012). Group invariant scattering. Communications on Pure and Applied Mathematics. 65(10): 1331-1398.
95. J. Wang, Z. Xiao, C. Liu, D. Zhao and Z. Yao (2019). Deep learning for picking seismic arrival times. Journal of Geophysical Research: Solid Earth. 124(7): 6612-6624.

96. Z. E. Ross, M.-A. Meier and E. Hauksson (2018). Pwave arrival picking and first-motion polarity determination with deep learning. Journal of Geophysical Research: Solid Earth. 123(6): 5120-5129.

97. M. Zhao, S. Chen, L. Fang and A. Y. David (2019). Earthquake phase arrival auto-picking based on U-shaped convolutional neural network. Chinese Journal of Geophysics. 62(8): 3034-3042.

98. Y. Zhou, H. Yue, Q. Kong and S. Zhou (2019). Hybrid event detection and phase-picking algorithm using convolutional and recurrent neural networks. Seismological Research Letters. 90(3): 1079-1087.

99. X. Zhang, J. Zhang, C. Yuan, S. Liu, Z. Chen and W. Li (2020). Locating induced earthquakes with a network of seismic stations in Oklahoma via a deep learning method. Scientific Reports. 10(1): 1941.

100. Z. E. Ross, Y. S. Yue, M. A. Meier, E. Hauksson and T. H. Heaton (2019). Phaselink: A deep learning approach to seismic phase association. Journal of Geophysical Research-Solid Earth. 124(1): 856-869.

101. X. Zhang and A. Curtis (2020). Seismic tomography using variational inference methods. Journal of Geophysical Research-Solid Earth. 125(4): e2019JB018589.

102. N. Yamaga and Y. Mitsui (2019). Machine learning approach to characterize the postseismic deformation of the 2011 Tohoku-Oki earthquake based on recurrent neural network. Geophysical Research Letters. 46(21): 11886-11892.

103. B. Rouet-LeDuc, C. L. Huibert, N. Lubbers, K. M. Barros, C. J. Humphreys and P. A. Johnson (2017). Machine learning predicts laboratory earthquakes. Geophysical Research Letters. 44(18): 9276-9282.

104. N. Anantrasirichai, J. Biggs, F. Albino, P. Hill and D. Bull (2018). Application of machine learning to classification of volcanic deformation in routinely-generated InSAR data. Journal of Geophysical Research. 123(8): 6592-6606.

105. C. Kadow, D. M. Hall and U. Ulbrich (2020). Artificial intelligence reconstructs missing climate information. Nature Geoscience. 13(6): 408-413.

106. T. Li, H. Shen, Q. Yuan, X. Zhang and L. Zhang (2017). Estimating ground-level PM2.5 by fusing satellite and station observations: A geo-intelligent deep learning approach. Geophysical Research Letters. 44(23): 11,985-11,993.

107. M. H. Shahnas, D. A. Yuen and R. N. Pysklywec (2018). Inverse problems in Geodynamics using machine learning algorithms. Journal of Geophysical Research-Solid Earth. 123(1): 296-310.

108. K. Fang, C. Shen, D. Kifer and X. Yang (2017). Prolongation of SMAP to spatiotemporally seamless coverage of continental U.S. Using a deep learning neural network. Geophysical Research Letters. 44(21): 11,030-011,039.

109. S. Ruhunusiri, J. S. Halekas, J. R. Espley, F. Eparvier, D. Brain, C. Mazelle, Y. Harada, G. A. DiBraccio, Y. Dong, Y. Ma, E. M. B. Thiemann, D. L. Mitchell and B. M. Jakosky (2018). An artificial neural network for inferring solar wind proxies at Mars. Geophysical Research Letters. 45(20): 10855-10865.

110. L. B. N. Clausen and H. Nickisch (2018). Automatic classification of auroral images from the oslo auroral themis (OATH) data set using machine learning. Journal of Geophysical Research-Space Physics. 123(7): 5640-5647.

111. B. Liu, X. Li and G. Zheng (2019). Coastal inundation mapping from bitemporal and dual-
polarization sar imagery based on deep convolutional neural networks. Journal of Geophysical Research-Oceans. **124**(12): 9101-9113.

112. V. Lempitsky, A. Vedaldi and D. Ulyanov (2018). Deep image prior. IEEE Conference on Computer Vision and Pattern Recognition, 9446-9454.

113. D. Yoo and I. S. Kweon (2019). Learning loss for active learning. IEEE Conference on Computer Vision and Pattern Recognition, 93-102.

114. M. W. Dunham, A. Malcolm and J. Kim Welford (2019). Improved well-log classification using semisupervised label propagation and self-training, with comparisons to popular supervised algorithms. Geophysics. **85**(1): O1-O15.

115. J. Yosinski, J. Clune, Y. Bengio and H. Lipson (2014). How transferable are features in deep neural networks. Neural Information Processing Systems. 3320-3328.

116. M. Oquab, L. Bottou, I. Laptev and J. Sivic (2014). Learning and transferring mid-level image representations using convolutional neural networks. IEEE Conference on Computer Vision and Pattern Recognition, 1717-1724.

117. K. Zhang, W. Zuo, S. Gu and L. Zhang (2017). Learning deep CNN denoiser prior for image restoration. IEEE Conference on Computer Vision and Pattern Recognition. 2808-2817.

118. M. Raissi, P. Perdikaris and G. E. Karniadakis (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics. **378**: 686-707.

119. Y. Wu and G. A. McMechan (2019). Parametric convolutional neural network-domain full-waveform inversion. Geophysics. **84**(6): R881-R896.

120. R. Iten, T. Metger, H. Wilming, L. Del Rio and R. Renner (2020). Discovering physical concepts with neural networks. Phys Rev Lett. **124**(1): 010508.

121. F. Garofalo, G. Sauvin, L. V. Socco and I. Lecomte (2015). Joint inversion of seismic and electric data applied to 2D media. Geophysics. **80**(4): EN93-EN104.

122. J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee and A. Y. Ng (2011). Multimodal deep learning. International Conference on Machine Learning. 689-696.

123. D. Ramachandram and G. W. Taylor (2017). Deep multimodal learning: A survey on recent advances and trends. IEEE Signal Processing Magazine. **34**(6): 96-108.

124. H. B. Memahan, E. Moore, D. Ramage, S. Hampson and B. A. Y. Arcas (2017). Communication-efficient learning of deep networks from decentralized data. International Conference on Artificial Intelligence and Statistics. 1273-1282.

125. T. Li, A. K. Sahu, A. Talwalkar and V. Smith (2020). Federated learning: Challenges, methods, and future directions. IEEE Signal Processing Magazine. **37**(3): 50-60.

126. A. Kortylewski, M. Wieser, A. Moreforster, A. Wieczorek, S. Parbhoo, V. Roth and T. Vetter (2018). Informed MCMC with Bayesian neural networks for facial image analysis. arXiv preprint arXiv:1811.07969.

127. M. Subedar, R. Krishnan, P. L. Meyer, O. Tickoo and J. Huang (2019). Uncertainty-aware audiovisual activity recognition using deep Bayesian variational inference. International Conference on Computer Vision. 6300-6309.

128. Y. Gal and Z. Ghahramani (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. International Conference on Machine Learning. 1050-1059.

129. D. Grana, L. Azevedo and M. Liu (2020). A comparison of deep machine learning and Monte Carlo methods for facies classification from seismic data. Geophysics. **85**(4): WA41-WA52.
130. S. Maiti and R. K. Tiwari (2010). Neural network modeling and an uncertainty analysis in Bayesian framework: A case study from the KTB borehole site. Journal of Geophysical Research-Solid Earth. **115**: 1-28.

131. R. Cao, S. Earp, S. A. L. de Ridder, A. Curtis and E. Galetti (2020). Near-real-time near-surface 3D seismic velocity and uncertainty models by wavefield gradiometry and neural network inversion of ambient seismic noise. Geophysics. **85**(1): KS13-KS27.