

Fast and Accurate Seismic Tomography via Deep Learning



Mauricio Araya-Polo, Amir Adler, Stuart Farris and Joseph Jennings

Abstract This chapter presents a novel convolutional neural network (CNN)-based approach to seismic tomography, which is widely used in velocity model building (VMB). VMB is a key step in geophysical exploration where a model of the subsurface is needed, such as in hydrocarbon exploration for the Oil & Gas industry. The VMB main product is an initial model of the subsurface that is subsequently used in seismic imaging and interpretation workflows. Existing solutions rely on numerical solutions of wave equations, and requires highly demanding computation and the resources of domain experts. In contrast, we propose and implement a novel 3D CNN solution that bypasses these demanding steps, directly producing an accurate subsurface model from recorded seismic data. The resulting predictive model maps relationships between the data space and the final earth model space. The subsurface models are reconstructed within seconds, namely, orders of magnitude faster than existing solutions. Reconstructed models are free of human biases since no initial model or numerical technique tuning is required. This chapter is a significant extension of previous published material and provides a detailed explanation of the seismic tomography problem, and of the previously unpublished 3D CNN architecture, training workflows and comparisons to state-of-the-art.

Keywords Deep learning · Exploration geophysics · Seismic imaging · Tomography · Inverse problems · Convolutional neural network

M. Araya-Polo · S. Farris · J. Jennings

Data Science and Machine Learning R&D, Shell International Exploration & Production Inc., Houston, TX, USA

A. Adler (✉)

Center for Brains, Minds and Machines (CBMM), MIT, Cambridge, MA, USA
e-mail: adleram@mit.edu

S. Farris · J. Jennings

Department of Geophysics, Stanford University, Stanford, CA, USA

© Springer Nature Switzerland AG 2020

W. Pedrycz and S.-M. Chen (eds.), *Deep Learning: Algorithms and Applications*, Studies in Computational Intelligence 865,
https://doi.org/10.1007/978-3-030-31760-7_5

1 Introduction

The main workflow of hydrocarbon exploration starts with acquiring field data, which consist of recordings of the response of the subsurface to artificial perturbations. Following data acquisition, several disciplines [1], geology, geophysics, petrophysics, etc., combine efforts to produce a model of the earth (see Fig. 1, top right) which may or may not have clear indications of hydrocarbon presence. In areas such as the Gulf of Mexico, hydrocarbons tend to accumulate near salt bodies making them a key geological structure in earth model building [2]. This earth model is a critical part of the decision making process and is given utmost importance during exploration projects. The average success ratio of the industry is low, thus avoiding unnecessary expenses, such as drilling wells, translates into saving millions of dollars. Therefore, techniques to accelerate the decision time and increase the success ratio are crucial.

What we propose in this chapter goes beyond what is currently making inroads in exploration geosciences, which is machine learning (ML) techniques being applied to specific well-known steps of the standard hydrocarbon exploration workflow (Fig. 1, red arrow). Most of the advances happen on the interpretation [3, 4] of the models rather than in the generation of them. Alternatively, our method is a end-to-end solution, producing earth models directly from unmanipulated seismic data. Our method differs from current velocity building methods, seismic tomography [5] (similar to medical tomography but the penetrating wave is seismic) or wave equation-based modeling/inversion, in that our method is automatic and without human intervention. The deep learning (DL) technique employed follows recent work [6, 7] that demonstrates this new approach, which uses a deep neural network (DNN) statistical model to transform raw input seismic data directly to the final mapping in 2D or 3D model space. The computational cost of the proposed approach is mostly due to the training phase, which occurs only once and offline. After training, velocity model reconstruction computational costs are negligible, thus making the overall computing requirements a fraction of those needed for traditional techniques, in particular the ones involving partial differential equations (PDE)-based simulations. As a preliminary step, velocity semblance [8] is used as the input feature space, which apparent seismic velocity (main attribute of an earth model) information for the training process. While we do perform feature extraction, rather than use the raw data, this feature extraction step is automated and not subject to human bias. Later, we extend the approach to work directly on the raw recordings thus freeing it from feature extraction and using the fully accepted unmanipulated seismic data as input.

The main design concern relates to the generalization capability of the DL-based solution, which basically indicates how much a trained model can accurately predict unseen data. To address that concern, we foresee models being trained with specific data belonging to different major exploration areas such as: pre-salt (Brasil offshore) or subsalt (Gulf of Mexico or West Africa). Regarding future hydrocarbon exploration workflows, one can imagine this technique being used just after data acquisition (field recording), then trained models are loaded up to the cloud from which interpreters can access realizations, thus performing online scenarios testing when feeding back

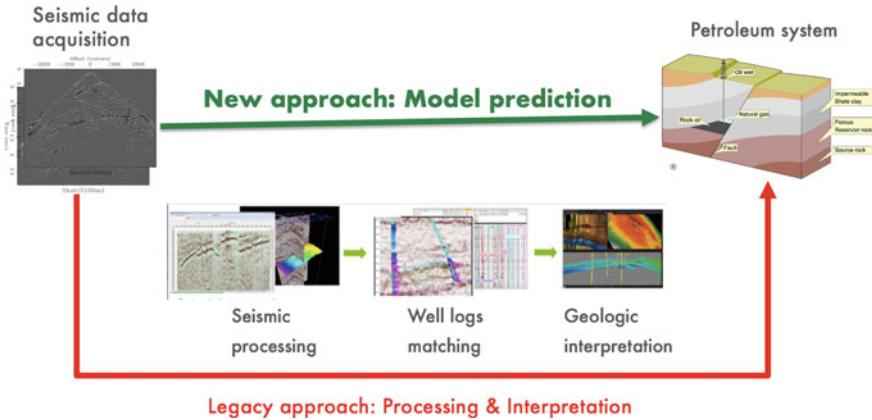


Fig. 1 Overall vision of the new exploration geophysics workflow (green arrow), where the classical way of approaching the problem is depicted in bottom following the red arrow

their model modifications. This anticipated workflow is fully ML-based, flexible and with the domain experts at the center of the critical decision making process. Finally, we envision that this technique can also be applied to other tomography problems that arise in the geosciences such as global seismology, shallow hazards, etc.

This chapter is organized as follows: Sect. 2 introduces the seismic tomography problem and the principles of seismic data acquisition. Section 3 explains the DL approach and the semblance geophysical feature. Section 4 presents experimental results with 2D synthetic seismic data. Section 5 compares our DL results against the state-of-the-art results obtained with industry's tool of choice. Section 6 introduces our preliminary results without extracting features from the data. Finally, conclusions and future research are provided in Sect. 7.

2 The Seismic Tomography Problem

To provide a complete context of the earth model building problem, before delving into our proposed DL-based solutions, this section explains the data to be used through the chapter and, in a succinct manner, reviews the scientific problem at hand.

2.1 Seismic Data

Seismic data are acquired, for the onshore case, via sources positioned on the earth's surface and arrays of receivers (geophones). In the offshore case, the sources and receivers (hydrophones) are towed by a ship, as illustrated in Fig. 2.

Seismic changes

The search for oil nearly always involves seismic imaging – determining what lies underground or beneath the seafloor by evaluating echoes from blasts of sound.

In recent years, offshore seismic operators have put a few twists on the process.

The basics

Conventional offshore seismic surveying uses a single vessel towing arrays of sensors called streamers in a series of parallel lines.

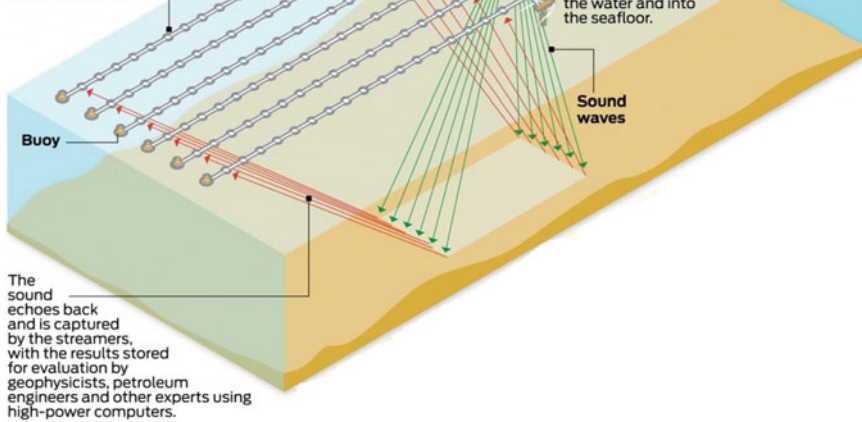


Fig. 2 Offshore seismic data acquisition (Source Houston Chronicle, BP, Schlumberger, Fairfield Nodal)

When energy is emitted from the source, it propagates through a highly heterogeneous medium (i.e., subsurface) which in turn creates reflections, refractions and diffraction effects that are recorded at the receiver (sensors) locations. As these recorded events are created due to changes in subsurface rock properties, inherently they contain information about the subsurface from whence they originated. The goal of seismic tomography and seismic imaging in general is to reconstruct the subsurface (earth model) that created the recorded seismic data.

With only one source firing and a finite number of receivers, only a limited portion of the subsurface target of interest can be sampled. Therefore, in order to adequately illuminate the subsurface, it is required that the source and array of receivers be positioned at multiple spatial locations. In reflection seismic terminology, the data obtained from the source firing at a single position \mathbf{x}_{s_i} into an array of receivers \mathbf{x}_{r_i} , $i = 1, \dots, N_r$ where N_r is the total number of receivers recording during a source firing is known as a “shot gather”. Modern reflection seismic data acquired for industrial purposes are composed of hundreds of thousands of shot gathers. Figure 3 depicts the ray paths (discrete approximation of a wavefront) associated with a shot gather for a single layer subsurface model and synthetic data recorded as a result of finite-difference modeling with a point source located at the position \mathbf{x}_s . Note that as the source moves along the surface with a dense array of receivers, subsurface

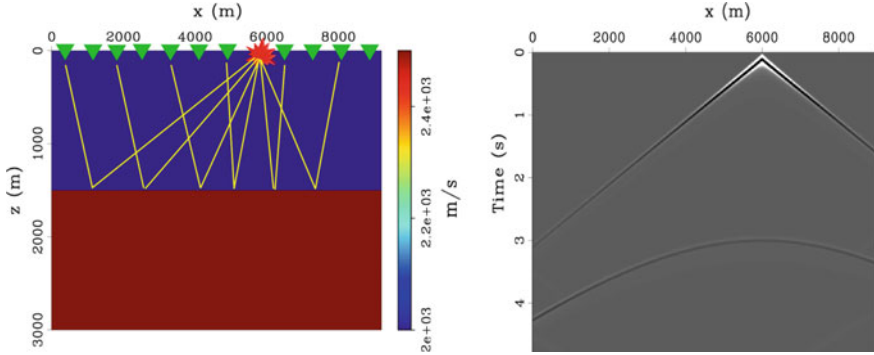


Fig. 3 (left) Raypaths for a seismic shot gather acquired over a flat layer earth. (right) Simulated data using finite-difference modeling. The linear event corresponds to the wave that travels directly from the source to the receivers along the surface. The hyperbolic event corresponds to the reflection of the wave off of the layer interface

points will be illuminated multiple times. To take advantage of this data redundancy, seismic data are typically transformed into midpoint and half-offset coordinates via the following relations

$$\mathbf{x}_{m_i} = \frac{\mathbf{x}_{s_i} + \mathbf{x}_{r_i}}{2},$$

$$\mathbf{x}_{h_i} = \frac{\mathbf{x}_{s_i} - \mathbf{x}_{r_i}}{2}.$$

where \mathbf{x}_{m_i} and \mathbf{x}_{h_i} are the midpoint and half-offset coordinates respectively. Figure 4 shows the resulting raypaths and data that arise from sorting the data in Fig. 3 into the midpoint and half-offset domain. As this collection of records is for a fixed midpoint and several offsets, this type of data is known as a common-midpoint gather. The processing of seismic data for velocity model building in general is performed with the data transformed into the midpoint and half-offset coordinates.

In Fig. 5, a selected group of traces from a more complex subsurface recording is presented. The field recordings—depending on the origin—are like the above depicted ones or more complex, therefore direct interpretation of subsurface structure is ruled out and this originates the need for advanced techniques to transform this data into usable models.

2.2 Seismic Tomography

The study of seismic tomography has spanned the past several decades and continues to be part of ongoing research [9]. While there exist many ways to formulate this

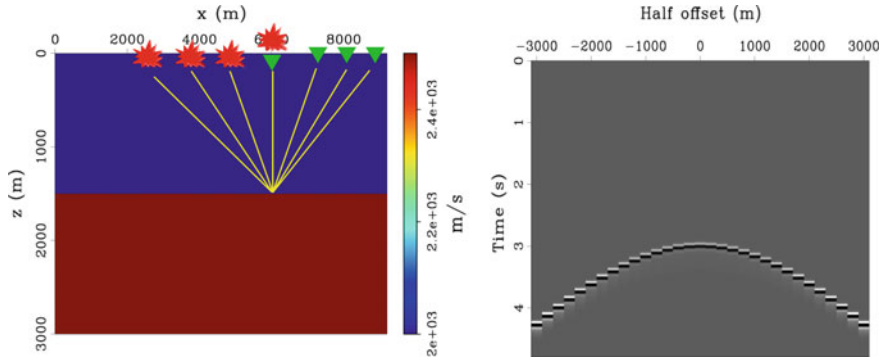


Fig. 4 (left) Raypaths for a seismic common midpoint gather acquired over a flat layer earth. (right) The synthetic data from Fig. 3 sorted into midpoint and half-offset coordinates and with a mute applied to the direct wave

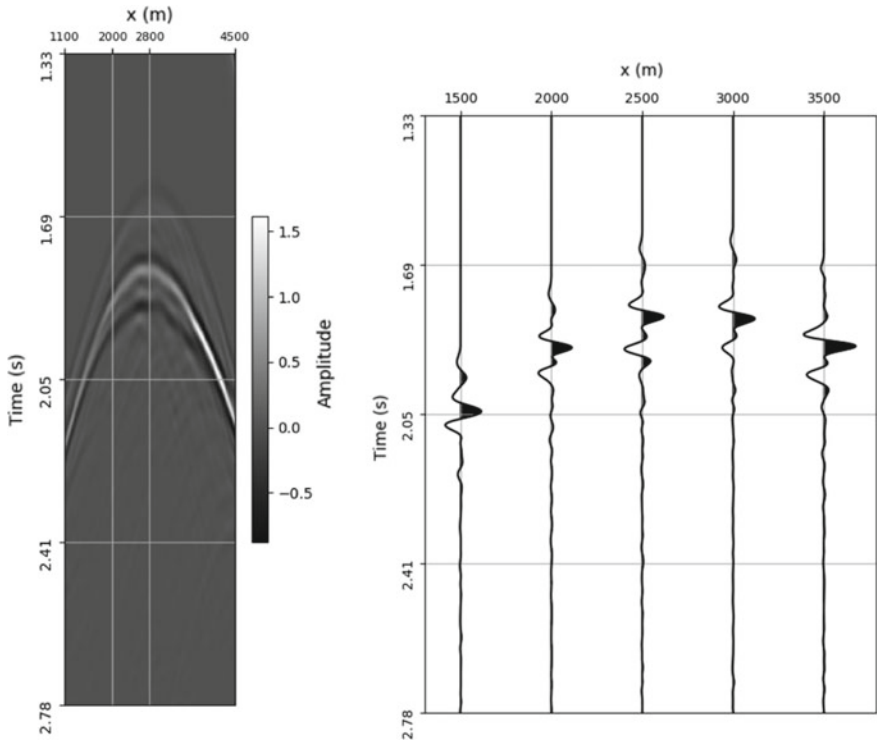


Fig. 5 (left) Windows in time and space on a shot gather from a complex subsurface model simulated with finite-difference modeling, therefore very high signal-to-noise ratio. (right) Selected traces from the shot gather of the left, traces presented as wiggles, where characteristics of the signals are shown

reconstruction problem [10–12], it can be expressed most generally as the following minimization problem:

$$\mathbf{m}^* = \arg \min_{\mathbf{m}} \{L(f(\mathbf{m}), \mathbf{d})\}, \quad (1)$$

where \mathbf{m} represents the earth model that we desire to recover, \mathbf{d} represents the recorded seismic data, $f(\mathbf{m})$ is a physics-based modeling that generates synthetic data from a prescribed earth model, L is a loss function that measures the misfit between the recorded data and the simulated data, and \mathbf{m}^* is the optimal earth model that minimizes the loss L . While a highly complex \mathbf{m} that informs us of many different earth properties (elastic moduli, density, viscoelastic parameters, etc.) is generally desired, \mathbf{m} commonly represents a three-dimensional acoustic wavespeed model. This choice of \mathbf{m} generally leads to the scalar acoustic wave equation as the choice for our physics-based simulation $f(\mathbf{m})$. Further simplifications can be made in taking the high-frequency limit of the scalar acoustic wave equation which results in the eikonal equation [13]. While the wave equation describes the propagation of waves and calculates synthetic seismograms (waveforms), the Eikonal equation is based on ray theory and calculates traveltimes. Regardless of the model parameterization and physics-based forward model used to fit the recorded geophysical data, the relationship between the data and the desired earth model is nonlinear. Therefore, a nonlinear optimization algorithm is required in order to minimize the loss-function (in Eq. (1)). Additionally, because $f(\mathbf{m})$ is in general very computationally expensive to evaluate, local/gradient-based methods for optimization must be used as opposed to global optimization methods. Using only the gradient information of the loss function can result in convergence to a local minimum and therefore unsatisfactory solutions. Additionally, because for reflection seismic surveys the data are recorded at the earth's surface, the data do not contain all of the necessary information to define a velocity model that varies arbitrarily space. This therefore implies that Eq. (1) defines a non-linear ill-posed optimization problem. In using a deep-learning approach, while we still face this issue of non linearity and ill-posedness, we do not rely on an accurate solution of the wave-equation, but rather directly learn a tomographic operator from many training examples that consist of the seismic data as feature and the velocity model as label.

3 Seismic Tomography via Deep Learning

3.1 Deep Neural Networks for Inverse Imaging Problems

Seismic tomography is an inverse imaging problem, in which the observation model can be represented as:

$$\mathbf{d} = f(\mathbf{m}) + \epsilon, \quad (2)$$

where \mathbf{d} is the observed seismic data, \mathbf{m} is the unknown earth model, $f(\cdot)$ is a mapping operator and ϵ is noise. While inverse imaging problems can be solved using analytic models, recent works [14–16] (and references within), argue that state-of-the-art results for a variety of inverse imaging problems can be obtained using deep learning methods. Following this line of work, we have proposed a novel approach [7] that implements the tomography operator using a convolutional neural Network (CNN), whose coefficients are learned in a data-driven approach [17]. The tomography process is depicted in Fig. 6, and it performs reconstruction of the velocity model from raw seismic traces, or from features computed from raw seismic traces. In a real-life application, the ground-truth model is unavailable, and the tomography operator is designed to minimize the difference between the reconstructed velocity model and the (unavailable) ground-truth one. The input to the tomography operator $\mathbf{T}(\mathbf{d}; \theta)$ is a set of seismic traces (or their features) \mathbf{d} , and it is parameterized by a coefficients vector θ . The tomography operator approximates the inverse mapping operator $f^{-1}(\cdot)$, and its output is the predicted velocity model $\hat{\mathbf{m}}$. In the statistical learning framework, the tomography operator is learned using a collection of N training example pairs $\{\mathbf{d}_i, \mathbf{m}_i\}_{i=1}^N$, where the data \mathbf{d}_i denotes the set of seismic traces (i.e. seismic gather) or their features, as generated by wave propagation simulation using the i -th velocity model \mathbf{m}_i (the i -th label). The average misfit between the ground truth models and their predicted versions, also known as the empirical risk, is defined by:

$$\mathbf{J}(\theta) = \frac{1}{N} \sum_{i=1}^N L(\mathbf{m}_i, \hat{\mathbf{m}}_i), \quad (3)$$

where $L(\mathbf{m}_i, \hat{\mathbf{m}}_i)$ is the loss function that measures the misfit between the ground truth velocity model and its prediction $\hat{\mathbf{m}}_i = \mathbf{T}(\mathbf{d}_i; \theta)$. The tomography operator is learned by minimizing the empirical risk:

$$\hat{\theta} = \arg \min_{\theta} \mathbf{J}(\theta). \quad (4)$$

The loss function employed in this work is the squared L_2 -norm of the pixel-wise difference $\hat{\mathbf{m}} - \mathbf{m}$, given by: $L(\mathbf{m}_i, \hat{\mathbf{m}}_i) = \|\mathbf{m}_i - \hat{\mathbf{m}}_i\|_2^2$, which is frequently used in regression problems, and leads to the following risk minimization problem:

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^N \|\mathbf{m}_i - \mathbf{T}(\mathbf{d}_i; \theta)\|_2^2. \quad (5)$$

In addition, regularization [18] of network parameters is optionally applied by an additional term $\mathbf{R}(\theta)$, leading to the following minimization problem:

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^N \|\mathbf{m}_i - \mathbf{T}(\mathbf{d}_i; \theta)\|_2^2 + \lambda \mathbf{R}(\theta), \quad (6)$$

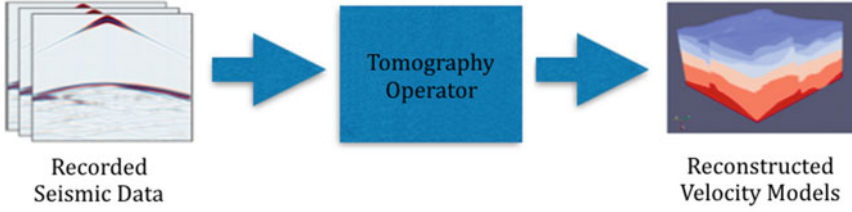


Fig. 6 Tomography reconstruction of velocity models from recorded seismic data

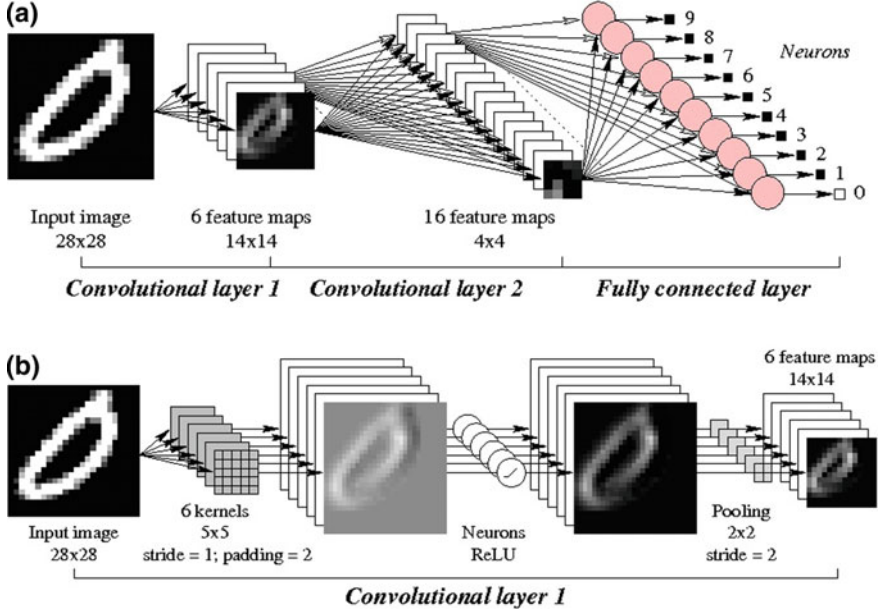


Fig. 7 Convolutional Neural Network (CNN): **a** a CNN with two convolutional layers and one fully-connected layer; and **b** zoom into the first convolutional layer (Source [20])

where $\lambda \geq 0$ controls the weight of the regularization term $\mathbf{R}(\theta)$, which is often defined as Ridge regression $\mathbf{R}(\theta) = \|\theta\|_2^2$ or Lasso regression $\mathbf{R}(\theta) = \|\theta\|_1$.

The tomography operator is implemented by a CNN, and thus can be represented as a hierarchical composition of k non-linear functions, each representing one of the k layers of the network:

$$\mathbf{T}(\mathbf{d}; \theta) = g_k(g_{k-1}(\cdots g_2(g_1(\mathbf{d}; \theta_1); \theta_2); \theta_{k-1}); \theta_k), \quad (7)$$

where $\theta = [\theta_1, \theta_2, \dots, \theta_{k-1}, \theta_k]^T$, and each function represents either a fully-connected (FC) or a convolutional layer [17, 19], as illustrated in Fig. 7.

3.2 Velocity Semblance as Input Feature for Deep Networks

Feature extraction is an optional step in our workflow as it can accelerate the training of the CNN by providing it with the most relevant data for learning. Our ML platform is capable of handling diverse network architectures and data, but given the focus on learning a tomographic operator from the data, we perform what is known as velocity (main subsurface model attribute) analysis and use its output as the input feature space.

To perform velocity analysis, we first transform the data into the midpoint half-offset coordinates as discussed previously. Then, we perform a time shift to each offset h of the common-midpoint gather in order to flatten the reflection (which has a hyperbolic shape) along the offset direction. This time-shift is a function of the half-offset h and the velocity in the medium V and can be calculated via the following relationship

$$t^2(h, V) = t_0^2 + \frac{h^2}{V^2}, \quad (8)$$

where t is the travel time of the hyperbolic event and t_0 is the time at which the data were recorded. Note that Eq. 8 describes the shape of a hyperbola which is exactly the shape of the recorded reflection shown in Fig. 4. Performing this time shift requires that the medium velocity be known a priori (which in the case of VMB is not). Therefore, trial velocities are prescribed in order to flatten the reflection event and then the following coherency measure is used in order to measure the flatness of the time-shifted event

$$s[i] = \frac{\sum_{j=i-M}^{i+M} \left(\sum_{k=0}^{N-1} q[j, k] \right)^2}{N \sum_{j=i-M}^{i+M} \sum_{k=0}^{N-1} q[j, k]^2}, \quad (9)$$

where $q[j, k]$ is the time-shifted common-midpoint gather for a particular velocity V and j and k are the time and offset indices respectively. The inner sum over all N offsets sums the time-shifted event along the offset direction. Therefore, the flatter the event (or the closer the prescribed velocity is to the true velocity), the greater the output of the sum. The outer sum is an average in time over a window of $2M + 1$ samples. The output $s[i]$ coherency measure is known as semblance [8] and is often the first step towards building a velocity model in reflection seismology. Performing this calculation for multiple midpoints, a semblance cube which has axes of time, velocity and midpoint can be created. The right half of Fig. 8 shows an example of a semblance cube for the velocity model shown in the left half of Fig. 8. Note that while the semblance cube does not offer very high resolution information about the velocity model. Rather, it gives an overall trend of how the velocity increases with depth from midpoint to midpoint.

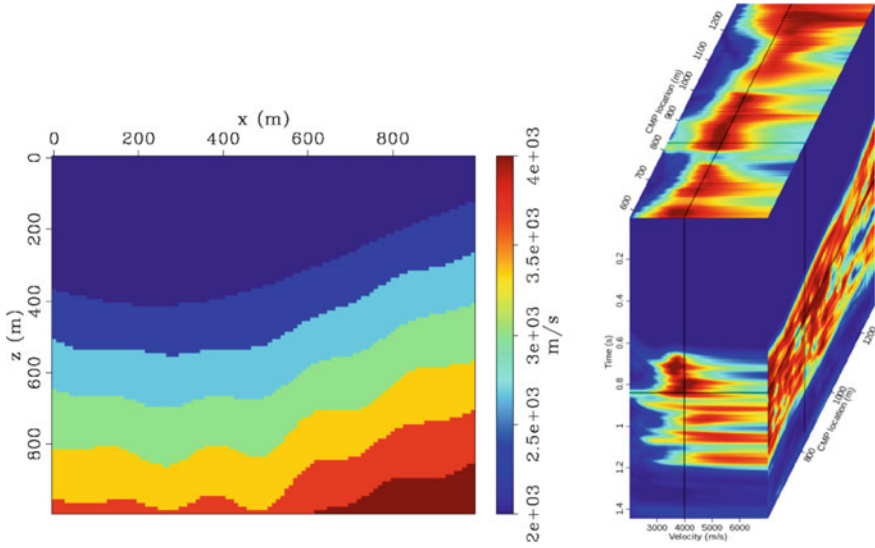


Fig. 8 (left) 2D Synthetic earth model, layers of sediments in a simple depositional system, velocity ranges between 2000 and 4500 m/s. Horizontal coordinate is and vertical represents depth. (right) Example of a calculated semblance cube for the model in left. In our case, during training, models like left are the labels and semblance cubes the input data

4 Semblance-Based CNN Results

In this section we first describe the experimental setup, including network architecture, datasets, hardware and software and metrics used for quantitative analysis. Second, we present the qualitative and quantitative results and discussion.

4.1 Experimental Setup

The network architecture is composed of four 3D convolutional layers (64 filters with kernel size of $6 \times 6 \times 6$) and two fully connected layers. Each layer employs a ReLU activation function. In addition, max-pooling, batch normalization and dropout with probability of 0.25 are deployed after each convolutional layer, as depicted in Fig. 9. The loss function is mean squared error and Nesterov ADAM [21, 22] optimizer is used. The network is implemented in python using TensorFlow [23] and Keras [24] as DL supporting frameworks.

The training reaches early stopping on around 250 epochs, in about 6 hours running on one high performance computing (HPC) node sporting four general purpose graphical processing units (GPGPUs) NVIDIA K80 [25] in data parallelism fashion. In this parallel execution mode, the model is copied to all computing units and

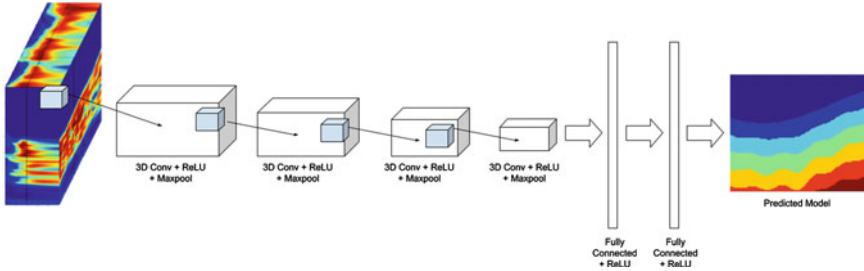


Fig. 9 Semblance-based 3D CNN architecture: the semblance cube is the input feature to the network, which includes four 3D convolutional layers and two fully connected layers. Each convolutional layer is composed of 3D kernels, ReLU activation per kernel, Maxpool, Batch Normalization and Dropout layer

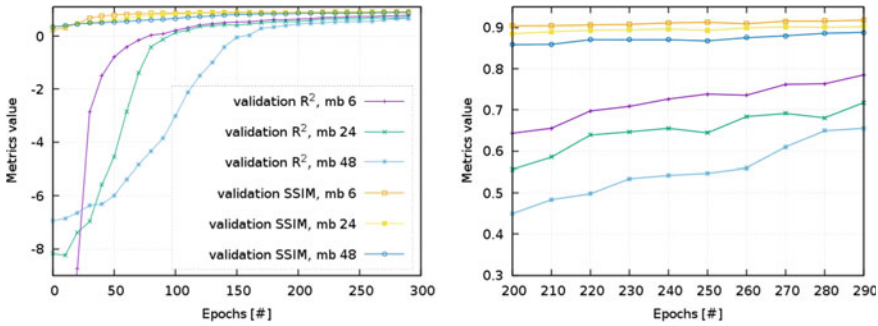


Fig. 10 (left) The plot shows the metrics value across training time. The vertical axis represent the metric value and the horizontal axis represents time in epoch units, where one epoch is a complete sweep through the training dataset. (right) A detailed view of the left plot around an area of interest. Plots share color codes and mb stands for minibatch

then the training data are evenly split and distributed among the computing units to be solved. Inference per model is a matter of seconds, which is extremely appealing when large amount of data is predicted or multiple velocity scenarios are under investigation.

Two datasets are prepared for both training and testing our model. In the first dataset, the velocity models only contain layers with velocities that increase with depth. Additionally, the layers exhibit both undulation and dip (Fig. 11). The second dataset consists of similar velocity models as the first dataset, but now a portion have been augmented with salt bodies. To add a factor of realism to these models, the shape of the salt bodies were extracted from earth models that were the end result of real life exploration projects in the Gulf of Mexico. Moreover, this dataset also contains velocity models without salt bodies. Each dataset consists of 6400 semblance cubes and the corresponding velocity model labels of size 100×100 grid points (the size of the output layer). For validation and testing purposes, we separated 1600 data/label pairings.

4.2 Quantitative Metrics

In terms of quantitative metric for model quality comparison, we decided to recourse to the widely accepted standard metric in image-dominated fields, the structural similitude index metric (SSIM) [26] and peak signal-to-noise ratio (PSNR). SSIM differs from traditional objective metric since it is based on structural degradation rather than error or general distortion of the images.

From statistical perspective the robustness of the model is measure with R^2 score (coefficient of determination).

4.3 Results and Analysis

The prediction accuracy metrics on the testing set for the first dataset (earth models only containing layers) are 0.812 for the R^2 score and 0.919 for the SSIM. R^2 score for

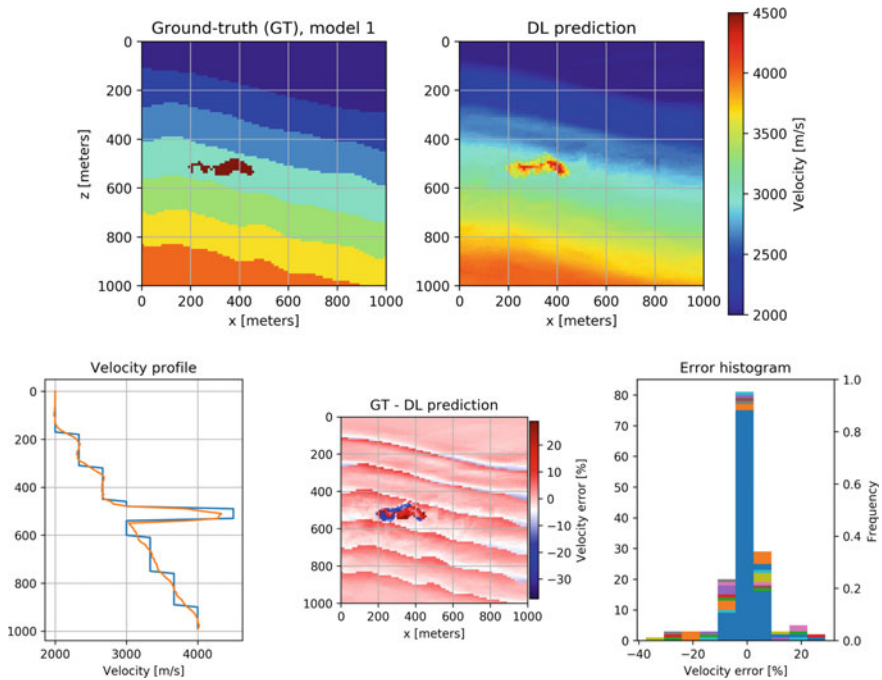


Fig. 11 (top, left) model 1 of the test dataset, includes salt bodies, which have high velocity and tend to distort classical modeling. Salt bodies are key in offshore hydrocarbon exploration. (top, right) prediction, where vertical axis represents depth and horizontal axis represents lateral offset. (bottom, left) comparison of the velocity profile for $x = 400$, where the vertical axis represents depth in meters. (bottom, center) absolute error between the ground-truth and prediction and (bottom, right) the corresponding error distribution

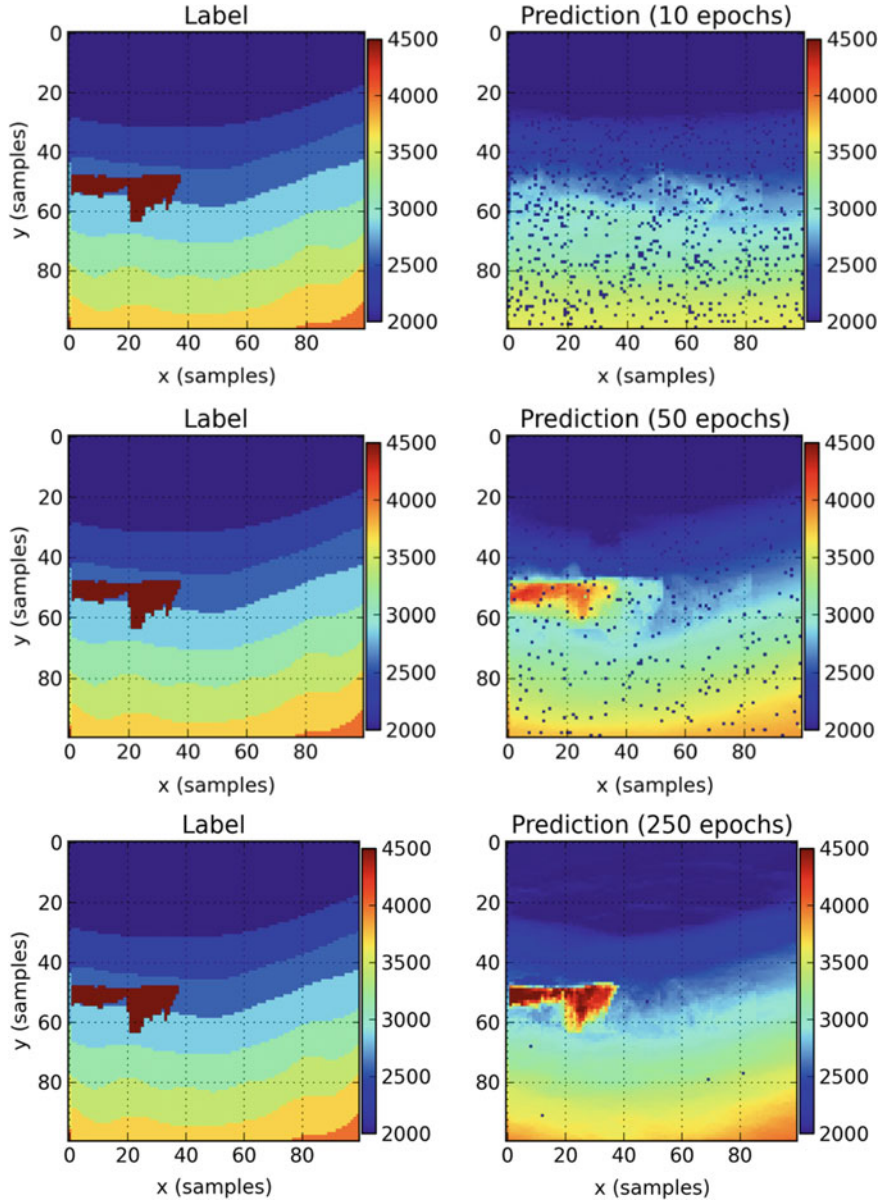


Fig. 12 Improving model reconstruction (for one model in the testing dataset) as the learning process sweeps through the training dataset

the test set with the second dataset is 0.741 and the SSIM is 0.892. The convergence of these metrics with respect to epoch can be observed in Fig. 10. The convergence curves shows the influence of different batch sizes on the performance for both metrics, although the effect is most noticeable on the R^2 metric, which converges later than the SSIM metric. As expected, the task of predicting a model with salt bodies is more difficult and therefore the performance is lower for dataset two than dataset one. It is difficult to learn the size, shape, and location of these salt bodies from the input data space. Furthermore, the datasets are relatively small for the task at hand. The main impediment to obtaining more training data is the computationally expensive step of generating features via finite difference wave propagation and calculating the semblance feature.

Qualitatively, the overall performance trend is positive, the salt bodies are mostly located properly and the surrounding formation resembles the labels in structure and velocity value (see Fig. 11), thus making the predicting model valid.

The main structural elements of the predicted model matches the ground-truth. The predicted expression and location of the salt body in Fig. 11 is remarkable. In particular, the velocity profile shows that the velocity trend is perfectly recovered, only missing the sharp interfaces between layers. In Fig. 12 we observe how a model from the validation set is learned as the training of the network progresses (by epochs). The first prediction (Fig. 12, top row) shows a model with low velocity in a gradient-based background, with many unresolved samples (blue dots). After few epochs (Fig. 12, center row) the predicted model corrects the deeper sections towards higher velocity and the salt body is clearly reconstructed. Finally, the model prediction is complete (Fig. 12) and even fine grained details of the salt body are satisfactory resolved.

5 Industry Baseline: Full Waveform Inversion

5.1 Industry VMB Methods

Many seismological techniques exist to estimate material properties in the shallow mantle of the earth. Ray tracing methods rely on high frequency ray theory approximation and picked arrival times of body waves to invert for optimal shear and pressure velocity models [27, 28]. These techniques are restricted to smooth medium predictions and fall short when surface wave amplitudes dominate body wave arrivals [29]. Full-waveform inversion (FWI) attempts to achieve this by iteratively simulating the seismic experiment and updating the earth model until the simulated seismic data matches the recorded seismic data in a least squares sense [30]. By fully modeling how energy propagates through the subsurface, FWI is more likely than other methods to find accurate representations of the earth's material properties [31].

5.2 Full Waveform Inversion

FWI uses the entire seismic waveform recording, that being all recorded frequencies and locations, to invert for earth model parameters beneath the surface. The goal of FWI is to find some earth model that minimizes the distance between modeled seismic data, which is a function of the earth model, and recorded seismic data, which was gathered in the field. When we have changed the earth model in such a way that the modeled data very closely resembles the recorded data, we assume we have found an earth model that is representative of the true earth model. Albert Tarantola [30] was the first to propose solving for earth parameters with such an inverse solution. In exploration geophysics, FWI is a topic of intense study and is at the forefront of earth model building from seismic data [32]. That being said, it is plagued with numerous limitations including high computational cost, extreme sensitivity to the choice of starting model, and unwanted convergence to incorrect earth model solutions. Moreover, when these limitations are properly accounted for and addressed, FWI is regarded as an area of development that may rectify the gap between low and high wavenumber earth model building and represent an all-inclusive solution to seismic exploration. For this reason we have chosen it as a baseline method to compare the velocity model prediction results of the ML approach defined previously. If ML can compete with the current cutting edge industry techniques, it will surely make waves in the exploration community.

More verbosely, consider the i th shot of a seismic survey \mathbf{d}_i^{obs} where $i = 1, 2, \dots, M$. Further, consider some modeled data, \mathbf{d}_i^{mod} , which is the synthetic recreation of the i th observed experiment. We can define the distance between the observed and modeled data as the L_2 norm of the two vectors,

$$L_2(\mathbf{d}_i^{mod}, \mathbf{d}_i^{obs}) = \|\mathbf{d}_i^{mod} - \mathbf{d}_i^{obs}\|_2. \quad (10)$$

To create the modeled data we use some wave equation operator, f_i , which represents a single seismic experiment. f_i is a function of the earth model, \mathbf{m} , and maps from the earth model space into the data space, $f_i(\mathbf{m}) = \mathbf{d}_i^{mod}$. In our case \mathbf{m} represents $1/v^2$, the inverse of the squared pressure wave velocity, at each point in the subsurface. Many wave equation formulations can model how seismic energy propagates through the earth. Generally speaking, the more complex and accurate the wave equation, the more computationally expensive the wave modeling becomes. For our purposes we use the acoustic, constant density, isotropic wave equation [33].

$$(\mathbf{A} - \mathbf{M}\mathbf{D}_2) \mathbf{p} = \mathbf{f}, \quad (11)$$

where \mathbf{A} is a laplacian operator, \mathbf{D}_2 is a second time derivative operator, \mathbf{M} is composed of nt copies of the flattened \mathbf{m} , where nt is the total number of time samples in the seismic recording, \mathbf{p} is the pressure wavefield, and \mathbf{f} is the injected seismic source. This wave equation assumes the earth can be represented by a single elastic parameter, pressure wave velocity, is isotropic, has constant density, and has a zero

shear modulus. We can represent this wave equation operator with a matrix, $H(\mathbf{m})$, and solve for the wavefield \mathbf{p}_i :

$$H(\mathbf{m})\mathbf{p}_i = \mathbf{f}_i \quad (12)$$

$$\mathbf{p}_i = H^{-1}(\mathbf{m})\mathbf{f}_i, \quad (13)$$

where \mathbf{p}_i represents the wavefield resulting from the i th seismic experiment in the entire domain. We can use an operator, K , to extract the wavefield at the point receiver locations to arrive at the modeled data, \mathbf{d}_i^{mod} :

$$\mathbf{d}_i^{mod} = K\mathbf{p}_i = KH^{-1}(\mathbf{m})\mathbf{f}_i = f_i(\mathbf{m}). \quad (14)$$

Using this wave equation operator we can define a scalar function, $J(\mathbf{m})$, which sums the L_2 difference between modeled and observed seismic data over all experiments:

$$J(\mathbf{m}) = \sum_{i=1}^N ||f_i(\mathbf{m}) - \mathbf{d}_i^{obs}||_2^2. \quad (15)$$

Here we have arrived at what is referred to as the FWI objective function. The model that reaches the minimum of this objective function is the solution to the FWI problem and the model that is our best estimate of the velocity profile beneath the surface.

Solving this inverse problem, that is finding the model that minimizes $J(\mathbf{m})$, is notoriously difficult for a variety of reasons. Primarily, the objective function is nonlinear with respect to \mathbf{m} , which means a perturbation in the earth model is not linearly mapped into the modeled data. It follows that the numerous, well studied strategies to solve linear least squares inverse problems are useless to us. Instead we must resort to nonlinear regression techniques for which there is no general theory for finding the optimal model parameters [34]. Iterative methods are a popular choice for solving nonlinear inverse problems and rely on the gradient of the objective function at the current model iteration, \mathbf{m}_j , to update the model parameters to find the next model iteration, \mathbf{m}_{j+1} .

$$\mathbf{m}_{j+1} = \mathbf{m}_j + \alpha_j \mathbf{s}_j. \quad (16)$$

The next model, \mathbf{m}_{j+1} , is found by summing the current model, \mathbf{m}_j , to some search direction, \mathbf{s}_j , scaled by a step length, α_j . There are many ways to compute the search direction, \mathbf{s}_j . We use the nonlinear conjugate gradient method in which:

$$\mathbf{s}_j = \mathbf{s}_{j-1} + \beta \nabla J(\mathbf{m}_j), \quad (17)$$

where \mathbf{s}_{j-1} is the previous search direction and β is the conjugate direction coefficient, and $\nabla J(\mathbf{m}_j)$ is the gradient of the objective function at the current model.

Furthermore, $B(m_j)^*$ is the adjoint of the wave equation operator linearized around the current model iteration applied to the difference between the modeled and observed data:

$$\nabla J(\mathbf{m}_j) = - \sum_{i=1}^N B(\mathbf{m}_j)^* (f_i(\mathbf{m}) - \mathbf{d}_i^{obs}) \quad (18)$$

To put it concisely, at each iteration of FWI we use the gradient of the objective function to update the earth model in order to reduce the value of the objective function. We stop iterating when the objective function reaches zero or, more realistically, once it stops reducing.

However, the nonlinearity of the FWI objective function means it is not convex with respect to the earth model. Gradient descent methods, like the one described above, will fall into a local minimum, that is find an earth model at which the objective function stops reducing but does not represent the global minimum of the objective function. In order to avoid local minima, the initial model used in the inversion scheme, \mathbf{m}_0 must be fairly close to the true model. Herein lies one of the largest restrictions of FWI, that being we must start from an earth model that is fairly close to the true model in order for the gradient based optimization algorithm to converge to the true solution.

Many methods exist and extensive research continues to find ways to avoid these convergence issues. A highly effective and widely accepted method is that of [35] which is referred to as multiscale FWI. This technique decomposes the FWI problem by scale and performs conventional FWI with progressively higher bandpasses of the source wavelet and observed data.

5.3 Baseline Comparison Setup

To compare the velocity model predictions of the proposed CNN-based approach and FWI, four synthetic seismic surveys are created and used as inputs. The intent is to keep the input data consistent in order to create a fair comparison between each method as in [36]. Below describes the data generation, the parameters of the three experiments, and the quantitative methods used to compare model results.

The synthetic models used to generate the seismic data are 1.8 km in the x direction and 1.4 km in the z direction with grid cell discretization of 10 m. The models parameter used is pressure wave velocity that increases with depth and contains salt bodies of varying shape and size. The velocities range from 2.0 to 4.5 km/s. All four of the models have layer cake backgrounds which are representative of the upper crust of the earth. Three of the four compared models contain high velocity zones, around 4500 m/s, which are characteristic of larger salt diapirs that often trap migrating oil and gas. It follows that the industry places large interest on finding and resolving these salt bodies.

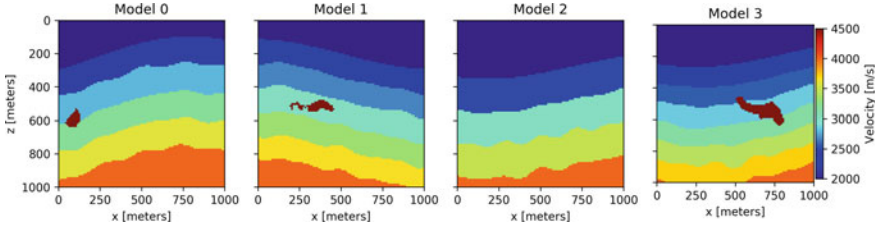


Fig. 13 Ground-truth velocity models

The data itself is generated from 19 shots at the surface with 40 m spacing in the x direction beginning at 520 m. The shot wavelet is a 15 Hz peak Ricker. 144 receivers located at the surface record pressure data. They begin at 180 m in x with 10 m spacing. The wave propagation modeling assumes an acoustic, constant density earth and uses second order approximation in time and eight order in space. Figure 13 illustrates the four models used to compare each method. Note, the data was generated on 1.8×1.4 km model but the velocity predictions were made on a 1.0×1.0 km subset of the original models.

5.4 Experiments

The first experiment is conducted with conventional FWI; 1000 iterations of non-linear conjugate gradient are performed using all frequencies of all modeled shots. The starting model was a linear velocity gradient from 2.0 to 4.5 km/s. A variation of this experiment is also conducted in which 200 conjugate gradient iterations are performed using the predicted model from the CNN as the starting model for FWI. The second experiment is multiscale FWI which performed 150 conjugate gradient inversions over 5 bandpasses of the all modeled shots. The first 4 bandpasses of the data were smoothly tapered at 4, 8, 16, and 32 Hz. The fifth inversion used all frequencies. The starting model for the 4 Hz inversion was a linear velocity gradient from 2.0 to 4.5 km/s. Each progressively higher bandpass inversion uses the final model from the previous bandpass inversion. The third experiment results are obtained by exposing the trained neural network to unseen data, in our case, to unseen semblance cubes from velocity models created by our pseudo-random velocity model generator.

5.5 Results

We perform the comparative analysis on four seismic datasets generated from the velocity models in Fig. 13. The comparison is limited to four datasets because of the high computational cost of FWI. In fact, retrieving one multiscale FWI result

takes more time than training the CNN used for the ML approach. After the upfront cost of creating the trained CNN, a single model prediction can be made almost instantaneously. This speaks to the computational cost of ML compared to FWI.

Figures 14, 15, and 16 depict the results of the three VMB methods on the four models both visually and numerically. In Figs. 14 and 15, rows correspond to various models and columns to VMB methods. Since salt diapirs are of large interest in the oil and gas community, comparisons are also made over windowed portions of the earth models that contain such bodies. Figure 16 gives a more in-depth look into the results on model 0 by computing difference plots between the true models and each of the VMB method results. This gives intuition on where each method is over or under-performing relative to the others. It also shows the error histograms to illustrate the distribution of velocity errors.

5.6 Discussion

A large impact would be made in the exploration seismic community if a method emerged that could construct earth models more effectively than FWI. We claim to have found such a method that leverages ML to show promising result on synthetic experiments. Our approach succeeds where FWI fails, in that ML is more robust, void of human bias, and computationally cheaper.

To backup this claim, we must analyze the experimental results visually and numerically in Figs. 14, 15, and 16. When comparing the results of the three approaches, we observe that both the CNN prediction and multiscale FWI were able to recover the original velocity model with good accuracy while the conventional FWI approach fell into a local minimum and was not able to recover a reasonable velocity solution. For example, examine the full view and the zoom view results of model 3, plotted in rows two and three of Fig. 15. The CNN and multiscale FWI methods both resolve the correct salt body location while conventional FWI completely misjudges the depth and shape of the body. Furthermore, the CNN better predicts the complex outline of the salt, including the bottom side which is notoriously difficult in real world applications. In general we find that the output of the DNN is smoother than the velocity estimated via multiscale FWI. This is highlighted in the difference plots of Fig. 16. One can see that the multiscale FWI approach performs better at resolving the interfaces between the layers. This likely due to the fact that when calculating the input semblance cubes, a smoothing occurs which limits the maximum frequency in the semblance cube. Multiscale FWI, however attempts to match modeled and to predicted data that may have a broader range of frequencies. Figures 14 and 15 do show that, generally, multiscale FWI, according to the SSIM metric, outperforms our ML results. But we find visually that the resolution of the salt bodies from the ML approach is more impressive and appealing from an oil and gas interpretation point of view.

Even though the results of DL are preliminary, they are already competitive with FWI. Consider the biases present in each approach. In order for FWI to succeed,

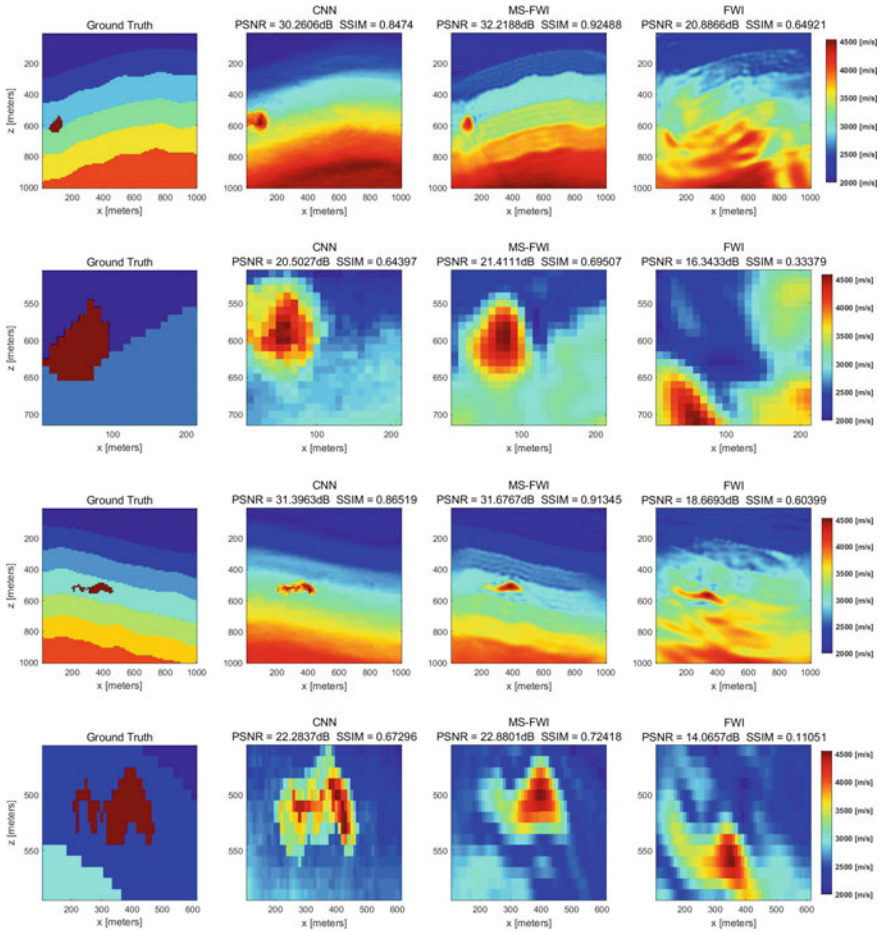


Fig. 14 Velocity Model Building results comparison: (1st row) Models 0, (2nd row) zoom into the salt body in Model 0, (3rd row) Models 1; and (4th row) zoom into the salt body in Model 1

we needed to use the multiscale scheme. There are dozens of other regularization methods that may or may not work depending on the specific experiment. It is left up to the geophysicist to decide. Furthermore, the sensitivity to the starting model means a priori information on the structure of the earth must be known. In real world scenarios, the starting model used in these experiments, the linearly increasing velocity profile with depth, will not suffice. A fairly detailed starting model must be constructed by the geophysicist beforehand. Alternatively, the ML approach did not need any handpicked regularization of the input data and it requires no starting model. ML retrieved competitive results without any human bias. Furthermore, FWI has been in development for 20 years and our ML method is also in its infancy. If we can recover sharper velocity model results with ML, and thus beat the FWI

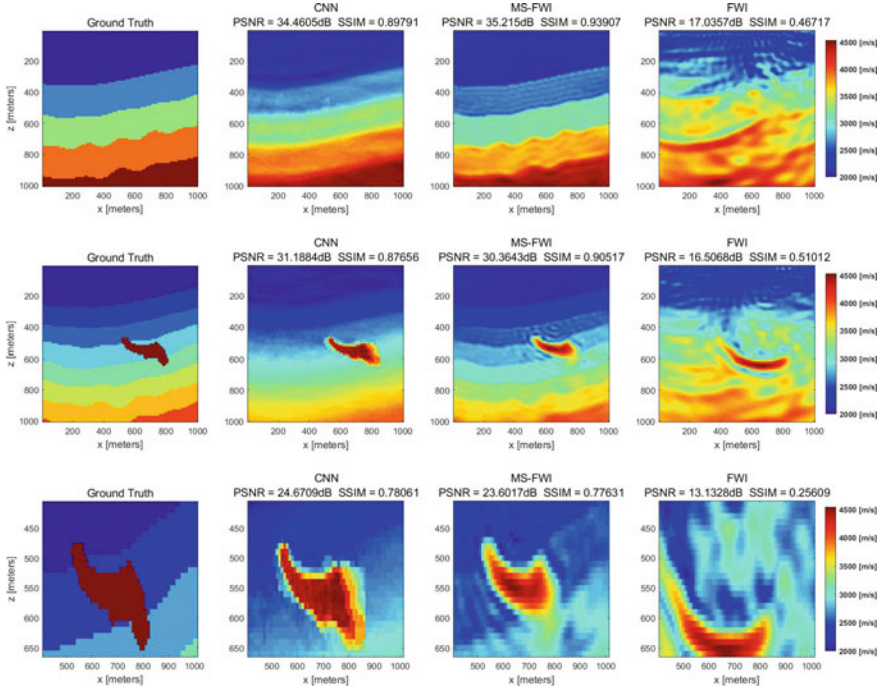


Fig. 15 Velocity Model Building results comparison: (1st row) Models 2; and (2nd row) Models 3, (3rd row) zoom into the salt body in model 0

results, there is nothing stopping ML from replacing FWI. Beyond comparing the velocity model results, we must address an equally important aspect, computational efficiency. The ML and FWI results were computed at different high performance computer cluster facilities, making a direct computational comparison difficult. But, we will find that examining precise clock cycles is not necessary because, by rough estimation, ML is orders of magnitude more efficient. Consider that to perform 1000 iterations of nonlinear conjugate gradient to recover the multiscale FWI results took about two days on a busy Stanford University computer cluster. Now, of course, the modeling and inversion codes used were for academic purposes and were therefore not fully optimized. But, the earth models used are also fairly small, $1 \text{ km} \times 1 \text{ km}$, by industry standards. If more efficient code was used on larger models, the compute time would most likely remain on the same order of magnitude; days. Now, consider that training the CNN model to map from semblance cubes to velocity models took about a day to finish. If larger models are used this may increase. Regardless, one may conclude that both methods are about equally efficient as both take on the order of days to finish. But, herein lies a critical difference between the two approaches; the CNN model is *reusable*. Once the training is finished, mapping from a single new dataset to a velocity model is nearly instant. Whereas, mapping a new dataset using multiscale FWI would take days to finish. The cost of the ML approach is all

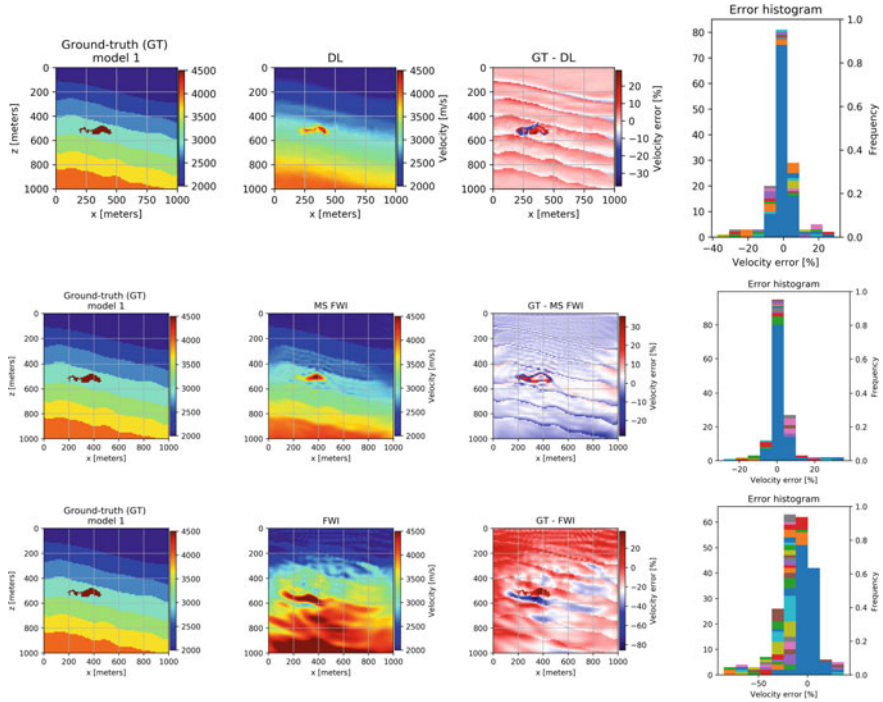


Fig. 16 Comparison of tomography results from the DL and FWI for model 0. Leftmost column shows ground-truth (label), second from left shows the prediction from the DL (top), the multiscale (MS) FWI result (middle) and the standard FWI result (bottom). Third column from left shows the difference between the ground truth and the prediction as a percentage of the velocity error. The last column shows the percentage of velocity errors for each sample binned and plotted in a histogram form. When comparing the prediction of the DNN to the MS FWI result, we observe that the DNN has difficulty in resolving sharp interfaces. Also note that a MS FWI approach was necessary to avoid cycle skipping that is apparent with the conventional FWI result

upfront and can be reused an infinite amount of times to make instant predictions. Nothing about FWI is reused and each application is equally expensive.

We show a new way of doing tomography with ML that leaves human biases and reoccurring high computational cost behind. While the ML results are competitive, they are still beat by a regularized FWI method. But, our ML method is also in it's infancy and FWI has been in development for over 20 years. Further progress may reap a ML method that can outperform FWI on all fronts, including model quality. A synergistic approach that utilizes both techniques is also an interesting, and a more realistic proposition. Using the unbiased results of ML as a starting model, FWI could fill in the remaining, sharp contrasts with fewer required iterations. This could quickly produce high quality models completely void of human bias. The broader case we make here is for the revolution of workflows in industry exploration. We see potential for many intermediate steps to be absorbed by ML-driven approaches, and seismic tomography is a stepping stone towards that.

6 Feature Extraction-Free Results

Human biased feature extraction is not desired when truly following the deep learning paradigm, which encourages an end-to-end learning process which maps from the relevant elements of the raw data to the ground-truth. After the initial success of the semblance-based approach, experiments were conducted with a modified version of the network (as depicted in Fig. 9) that accepts seismic gathers without manipulation as inputs (Fig. 17, right). The label dataset (Fig. 17, left) is composed by the velocity models as described in previous sections. The main change in the network, compared to the one presented in Sect. 4, is in the input layer. Now the input is the raw seismic shot gathers which are of the dimension (number of shots \times number of receivers \times time samples). Each data/label pairing is composed by the later described 3D seismic gather and the corresponding velocity model as label. Furthermore, this network's training used the Nesterov optimizer ADAM with a learning rate of $1e-03$, batch size of 20 (per GPGPU) and the experiments were executed for 250 epochs using the MSE loss function. Training takes less than two hours and inference takes only seconds.

It can be observed in Fig. 18 that the velocity model used as label is larger than the ones used in Sect. 4 and much more rich in features (velocity variation), this is because these velocity models belong to datasets used in exploration in the Gulf of Mexico (due to confidentiality reasons actual geographical locations can not be shared). Consequentially, the generated seismic data is much closer to what actual field records look like. The decision of using this data is not random, the final purpose is to expose the ML approach to real field data and therefore cross the threshold from research

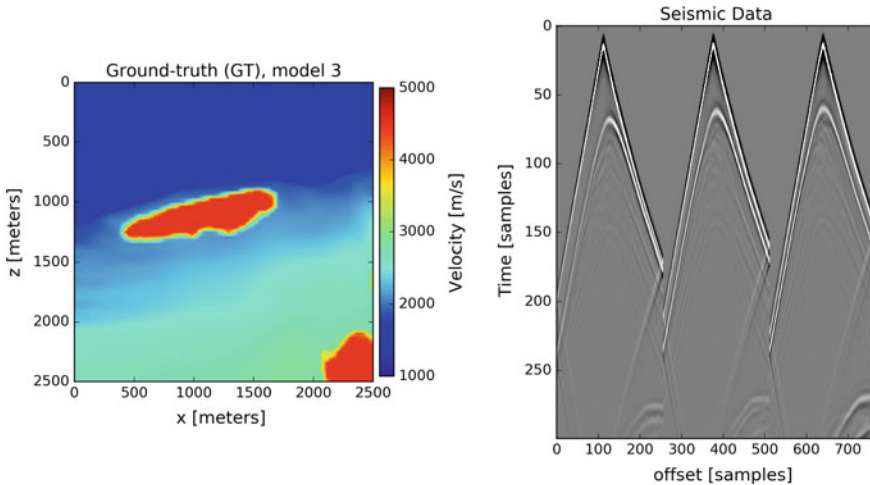


Fig. 17 (left) Example velocity model from the training dataset, (right) examples of corresponding seismic traces (only three selected sources), obtained by wave propagation and without first arrival removal, where vertical axis represents time and horizontal is offset from source location

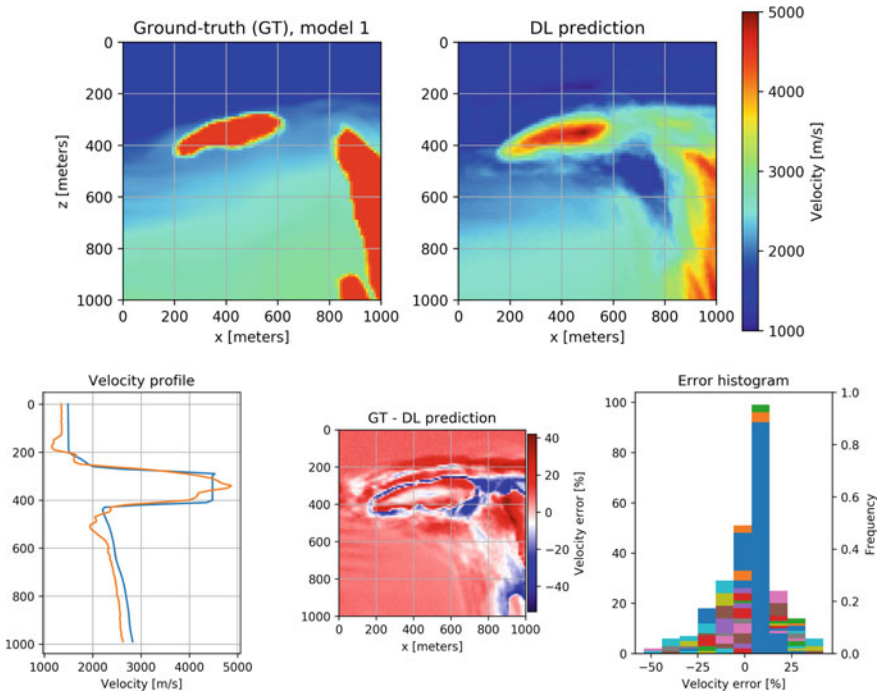


Fig. 18 Results examples for a trained model without pre-computed features. (top, left) shows the ground-truth and (top, right) the corresponding prediction. (bottom, left) comparison of the velocity profile for $x = 500$, vertical axis represents depth in meters

into industrial-tested tool. One step in data preparation is to downsample the label and data, in particular the data was downsampled to fit in GPGPU memory, which for these experiments are three NVIDIA V100s with 16 GB of internal memory each. The input data dimension used is $31 \times 256 \times 300$ (where dimension were described in previous paragraph). The label and predicted model dimension is 100×100 samples, where the first dimension represents the horizontal axis and the second dimension represents depth or vertical dimension. The total size of the training dataset is 960 samples, where 80 sample were separated for validation and another 80 samples for testing.

The results training no pre-computed features (in Fig. 18) are at least comparable if not superior to the ones presented in Sect. 4. Comparable in the sense that all major features of the expected reconstructed models are present and the error ranges are similar. In quantitative terms, the test dataset SSIM metric is 0.8181 and the R^2 metric is 0.8272. This two figures are slightly less impressive than the ones reported in Sect. 4, three main factors are the culprit: complex velocity models, smaller dataset and forced downsampled data. In qualitative terms, the largest errors appear around the fine-grained contours of salt bodies, which is also the case for the traditional

techniques as is shown in Sect. 5. Nonetheless, the results are superior in the sense that the expected reconstructed models are more detailed and therefore harder to predict, also these velocity models are essentially what is used in exploration geophysics within seismic imaging workflows.

7 Conclusions

This chapter presents a novel DL approach to a key geoscience problem [5, 37]. By utilizing DL, it is possible to predict earth models directly from the recorded seismic data. Essentially, we are replacing an nonlinear inverse problem with a data-driven learning process. Results with synthetic data achieve high visual accuracy, both with structural similarity image metric (SSIM) and PSNR. This solution enables fast turnaround of exploration workflows that nowadays take weeks to complete, therefore empowering domain experts allowing them to focus on the most complex prospects within the data. The proposed approach can be extended to other relevant geoscience problems where accurate earth model are also required. Future work is twofold: extension to 3D tomography, namely reconstruction of three dimensional subsurface models, and validation with field recorded seismic data.

Acknowledgements The authors would like to thank Shell International Exploration and Productions for supporting and allowing us to share this material.

References

1. Alpak, F.O., Araya-Polo, M.: Big loop in the machine learning era. In: 81st EAGE Conference and Exhibition (2019)
2. Farmer, P., Miller, D., Pieprzak, A., Rutledge, J., Woods, R.: Exploring the subsalt. *Oilfield Rev.* 50–64 (1996)
3. AlRegib, G., Deriche, M., Long, Z., Di, H., Wang, Z., Alaudah, Y., Shafiq, M.A., Alfarraj, M.: Subsurface structure analysis using computational interpretation and learning: a visual signal processing perspective. *IEEE Signal Process. Mag.* **35**(2), 82–98 (2018)
4. Di, H., Wang, Z., AlRegib, G.: Why using CNN for seismic interpretation? An investigation. In: SEG, pp. 2216–2220 (2018)
5. Rawlinson, N., Pozgay, S., Fishwick, S.: Seismic tomography: a window into deep earth. *Phys. Earth Plan. Inter.* **178**(3), 101–135 (2010)
6. Araya-Polo, M., Dahlke, T., Frogner, C., Zhang, C., Poggio, T., Hohl, D.: Automated fault detection without seismic processing. *Lead. Edge* **36**(3), 208–214 (2017)
7. Araya-Polo, M., Jennings, J., Adler, A., Dahlke, T.: Deep-learning tomography. *Lead. Edge* **37**(1), 58–66 (2018)
8. Claerbout, J.F.: *Straightedge Determination of Interval Velocity* (1978)
9. Virieux, J., Brossier, R., Mtivier, L., Etienne, V., Operto, S.: Challenges in the full waveform inversion regarding data, model and optimisation. In: 74th EAGE Conference and Exhibition-Workshops (2012)

10. Stork, C., Clayton, R.W.: Linear aspects of tomographic velocity analysis. *Geophysics* **56**(4), 483–495 (1991)
11. Sava, P., Biondi, B.: Wave-equation migration velocity analysis. I. theory. *Geophys. Prospect.* **52**(6), 593–606 (2004)
12. Tromp, J., Tape, C., Liu, Q.: Seismic tomography, adjoint methods, time reversal and banana-doughnut kernels. *Geophys. J. Int.* **160**(1), 195–216 (2005)
13. Fomel, S.: Traveltimes computation with the linearized Eikonal equation. *Stanf. Explor. Project Rep.* **94**, 123–131 (1997)
14. Jin, K.H., McCann, M.T., Froustey, E., Unser, M.: Deep convolutional neural network for inverse problems in imaging. *IEEE Trans. Image Process.* **26**(9), 4509–4522 (2017)
15. Lucas, A., Iliadis, M., Molina, R., Katsaggelos, A.K.: Using deep neural networks for inverse problems in imaging: beyond analytical methods. *IEEE Signal Process. Mag.* **35**(1), 20–36 (2018)
16. McCann, M.T., Jin, K.H., Unser, M.: Convolutional neural networks for inverse problems in imaging: a review. *IEEE Signal Process. Mag.* **34**(6), 85–95 (2017)
17. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press. <http://www.deeplearningbook.org> (2016)
18. Hastie, T., Tibshirani, R., Friedman, J.H.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, New York (2009)
19. LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. In: *The Handbook Of Brain Theory and Neural Networks*, vol. 3361, no. 10 (1995)
20. Couchot, J.F., Couturier, R., Guyeux, C., Salomon, M.: Steganalysis via a convolutional neural network using large convolution filters (2016). CoRR, [arXiv:1605.07946](https://arxiv.org/abs/1605.07946)
21. Kinga, D., Ba, J.: Adam: a method for stochastic optimization. In: *International Conference on Learning Representations (ICLR)*, vol. 5 (2015)
22. Dozat, T.: Incorporating nesterov momentum into adam. In: *Proceedings of ICLR Workshop* (2016)
23. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). Software available from www.tensorflow.org
24. Chollet, F., et al.: Keras. <https://keras.io> (2015)
25. nVIDIA, Tesla K40 and K80 GPU accelerators for servers. <http://www.nvidia.com/object/tesla-servers.html> (2014)
26. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
27. Červeny, V.: *Seismic Ray Method*. Cambridge University Press (2001)
28. Vidale, J.: Finite-difference calculations of travel times. *Bull. Seismol. Soc. Am.* **78**(6), 2062–2076 (1988)
29. Ellefsen, K.J.: A comparison of phase inversion and traveltimes tomography for processing near-surface refraction traveltimes. *Geophysics* **74**(6), WCB11–WCB24 (2009)
30. Tarantola, A.: Inversion of seismic reflection data in the acoustic approximation. *Geophysics* **49**(8), 1259–1266 (1984)
31. Romdhane, A., Grandjean, G., Brossier, R., Réjiba, F., Operto, S., Virieux, J.: Shallow-structure characterization by 2D elastic full-waveform inversion. *Geophysics* **76**(3), R81–R93 (2011)
32. Virieux, J., Operto, S.: An overview of full-waveform inversion in exploration geophysics. *Geophysics* **74**(6), WCC1–WCC26 (2009)
33. Aki, K., Richards, P.G.: *Quantitative seismology*
34. Aster, R.C., Borchers, B., Thurber, C.H.: *Parameter Estimation and Inverse Problems*. Elsevier (2005)

35. Bunks, C., Saleck, F.M., Zaleski, S., Chavent, G.: Multiscale seismic waveform inversion. *Geophysics* **60**(5), 1457–1473 (1995)
36. Farris, S.: Tomography: a deep learning vs full-waveform inversion comparison. In: First EAGE Workshop on High Performance Computing for Upstream in Latin America (2018)
37. Taner, M.T., Koehler, F.: Velocity spectra—digital computer derivation applications of velocity functions. *Geophysics* **34**(6), 859–881 (1969)