Deep Convolutional Neural Network and Sparse Least Squares Migration

<table>
<thead>
<tr>
<th>Item Type</th>
<th>Article</th>
</tr>
</thead>
<tbody>
<tr>
<td>Authors</td>
<td>Liu, Zhaolun; Chen, Yuqing; Schuster, Gerard T.</td>
</tr>
<tr>
<td>Eprint version</td>
<td>Publisher’s Version/PDF</td>
</tr>
<tr>
<td>DOI</td>
<td>10.1190/geo2019-0412.1</td>
</tr>
<tr>
<td>Publisher</td>
<td>Society of Exploration Geophysicists</td>
</tr>
<tr>
<td>Journal</td>
<td>GEOPHYSICS</td>
</tr>
<tr>
<td>Rights</td>
<td>Archived with thanks to GEOPHYSICS. © 2020 The Authors. Published by the Society of Exploration Geophysicists. All article content, except where otherwise noted (including republished material), is licensed under a Creative Commons Attribution 4.0 Unported License (CC BY). See <a href="http://creativecommons.org/licenses/by/4.0/">http://creativecommons.org/licenses/by/4.0/</a>. Distribution or reproduction of this work in whole or in part commercially or noncommercially requires full attribution of the original publication, including its digital object identifier (DOI).</td>
</tr>
<tr>
<td>Download date</td>
<td>2023-10-31 22:02:03</td>
</tr>
<tr>
<td>Item License</td>
<td><a href="http://creativecommons.org/licenses/by/4.0/">http://creativecommons.org/licenses/by/4.0/</a></td>
</tr>
<tr>
<td>Link to Item</td>
<td><a href="http://hdl.handle.net/10754/660193">http://hdl.handle.net/10754/660193</a></td>
</tr>
</tbody>
</table>
Deep convolutional neural network and sparse least-squares migration

Zhaolun Liu¹, Yuqing Chen², and Gerard Schuster³

ABSTRACT

We have recast the forward pass of a multilayered convolutional neural network (CNN) as the solution to the problem of sparse least-squares migration (LSM). The CNN filters and feature maps are shown to be analogous, but not equivalent, to the migration Green’s functions and the quasi-reflectivity distribution, respectively. This provides a physical interpretation of the filters and feature maps in deep CNN in terms of the operators for seismic imaging. Motivated by the connection between sparse LSM and CNN, we adopt the neural network version of sparse LSM. Unlike the standard LSM method that finds the optimal reflectivity image, neural network LSM (NNLSM) finds the optimal quasi-reflectivity image and the quasi-migration Green’s functions. These quasi-migration Green’s functions are also denoted as the convolutional filters in a CNN and are similar to migration Green’s functions. The advantage of NNLSM over standard LSM is that its computational cost is significantly less and it can be used for denoising coherent and incoherent noise in migration images. Its disadvantage is that the NNLSM quasi-reflectivity image is only an approximation to the actual reflectivity distribution. However, the quasi-reflectivity image can be used as an attribute image for high-resolution delineation of geologic bodies.

INTRODUCTION

Deep convolutional neural networks (CNNs) have been recently used for solving geophysical problems, such as seismic first-arrival picking (Lu and Feng, 2018; Yuan et al., 2018; Hu et al., 2019), seismic interpretation (Shi et al., 2019; Wu et al., 2019a, 2019b), and seismic imaging and inversion (Xu et al., 2019; Kaur et al., 2020; Sun et al., 2020). Interpretation examples include salt classification (Waldeland et al., 2018; Shi et al., 2019), fault detection (Huang et al., 2017; Xiong et al., 2018; Wu et al., 2019b; Zheng et al., 2019), reservoir characterization (Karimpouli et al., 2010; Cao and Roy, 2017; Zhu et al., 2017), and seismic lithofacies classification (Ross and Cole, 2017; Liu et al., 2019) with semisupervised learning (Di et al., 2020). Other applications include the use of neural networks for well-log interpolation (Saggaf and Nebrija, 2003; Salehi et al., 2017; Pham et al., 2020), seismic-data interpolation (Mandelli et al., 2018, 2019; Wang et al., 2020), velocity-model building (Araya-Polo et al., 2018; Richardson, 2018), well-log ties (Bader et al., 2019), synthetic well-data generation (Rolon et al., 2009), autoencoders for unsupervised facies analysis (Qian et al., 2018), and supervised horizon tracking (Peters et al., 2019a, 2019b). Recently, an unsupervised autoencoder method with regularization was developed by Shi et al. (2020) to track target horizons.

There are many types of neural network or machine-learning methods, including generative adversarial networks for seismic interpolation (Siahkoohi et al., 2018), residual networks for traces missing at regular intervals (Wang et al., 2019), synthetic well-data generation (Achim et al., 2009), autoencoders for unsupervised facies analysis (Qian et al., 2018), and supervised horizon tracking (Peters et al., 2019a, 2019b). Recently, an unsupervised autoencoder method with regularization was developed by Shi et al. (2020) to track target horizons.
tracking, and recurrent neural networks for well-log interpolation (Pham et al., 2020). The design of geophysical CNN architectures is largely based on empirical evidence from computer vision research, insights from the principles of artificial intelligence, and heuristic experimentation. Heuristic experimentation is most often used to decide the parameter design (the design of a CNN architecture) that selects the number of CNN layers, the number of filters/layer, the size of the filters, the type of activation functions, the number of skip layers, and whether a layer acts as a decoding or encoding operation) for the CNN architecture, which has merits and liabilities. The merit is that trial-and-error with different architecture parameters is likely to give excellent results for a particular data set, but it might not be the best one for a different data set. This shortcoming in using empirical tests for parameter selection largely results from the absence of a rigorous mathematical foundation (Papyan et al., 2016) for neural networks in general, and CNN in particular.

To partly mitigate this problem for CNN-based imaging algorithms, we now present a physical interpretation of CNN filters and feature maps in terms of physics-based operators for seismic imaging. With such an understanding, we can use a physics-based rationale for the better design of CNN architectures in seismic migration and inversion.

Donoho (2019) points out that “machine learning has a troubled relationship with understanding the foundation of its achievements well and its literature is admittedly corrupted by anti-intellectual and antischolarly tendencies.” Progress in advancing the capabilities of deep neural networks will be severely stymied unless its mathematical foundations are established. As a first step in this direction, Papyan et al. (2016) propose that the forward modeling operation of CNN could be recast as finding the sparsest model \( \mathbf{m} \) under the \( \ell_1 \) norm subject to honoring the data-misfit constraint

\[
\| \mathbf{Gm} - \mathbf{d} \|_2 \leq \beta,
\]

where \( \mathbf{m}^* \) is the optimal solution, \( \mathbf{G} \) is the dictionary matrix, \( \mathbf{d} \) represents the signal, and the scalar \( \beta \) is the specified noise tolerance. The iterative solution to this problem is a series of forward modeling operations of a neural network, in which the mathematical operations of each layer consist of two steps: a weighted summation of input values to give the vector \( \mathbf{z} \) followed by a two-sided soft thresholding operation denoted as \( \sigma(\mathbf{z}) \) (Papyan et al., 2016).

The sparse constraint problem defined in equation 1 is commonly seen in geophysics, e.g., least-squares migration (LSM) with a sparsity constraint. LSM is an important seismic-imaging technique that produces images with better balanced amplitudes, fewer artifacts, and better resolution than standard migration (Lailly, 1983; Tarantola, 1987; Schuster, 1993; Chavent and Plessix, 1999; Nemeth et al., 1999; Duquet et al., 2000; Feng and Schuster, 2017; Schuster and Liu, 2019). The sparsity constraint is one of the important regularization terms used in solving the ill-conditioned least-squares problem (Sacchi and Ulrych, 1995; De Roeck, 2002; Kühn and Sacchi, 2003; Wang and Sacchi, 2005), and sparse LSM (SLSM) has been demonstrated to be effective for mitigation of aliasing artifacts and crosstalk in LSM (Wang and Sacchi, 2007; Herrmann et al., 2009; Dutta, 2017; Witte et al., 2017; Li and Gao, 2018). Image-domain SLSM finds the optimal reflectivity \( \mathbf{m}^* \) with sparse constraints to minimize the objective function \( \| \mathbf{Gm} - \mathbf{m}^{\text{img}} \|_2^2 \), where \( \mathbf{G} \) is the Hessian matrix and \( \mathbf{m}^{\text{img}} \) is the migration image. Here, we can see that SLSM shares the same problem as that defined in equation 1. Following the work of Papyan et al. (2016), we show that the sparse solution to the LSM problem reduces to the forward modeling operations of a multilayered neural network. The CNN filters and feature maps are shown to be analogous, but not equivalent, to the migration Green’s functions (Hessian) and the reflectivity distribution.

The standard SLSM algorithm needs to solve the wave equation, which is time-consuming. Motivated by the connection between SLSM and CNN, we propose the neural network version of sparse LSM, which does not need to solve the wave equation for the back-propagation of residuals and is faster than standard SLSM. Instead of just finding the optimal reflectivity \( \mathbf{m}^* \), we optimize for the quasi-reflectivity \( \mathbf{m} \) and the quasi-migration Green’s functions \( \mathbf{G} \). These quasi-migration Green’s functions approximate the role of migration Green’s function (Schuster and Hu, 2000) and are denoted as the convolutional filters in a CNN. As discussed in Appendix A, the migration Green’s function is the point scatterer response of the migration operator. The final image is denoted as the neural network least-squares migration (NNLSM) estimate of the quasi-reflectivity distribution that honors the \( \ell_1 \) sparsity condition.

The next section shows the connection between the multilayer neural network and the solution to the multilayer NNLSM problem. This is followed by numerical examples with synthetic models and field data from the North Sea.

THEORY OF NEURAL NETWORK LSM

The theory of standard image-domain LSM is first presented to establish the benchmark solution in which the optimal reflectivity function minimizes the image misfit under the \( \ell_2 \) norm. This is then followed by the derivation of the SLSM solution for a single-layer network. The final two subsections derive the NNLSM solution for single-layer and multilayer networks, respectively.

Least-squares migration

The LSM problem can be defined (Schuster and Hu, 2000; Schuster, 2017) as finding the reflectivity coefficients \( m_i \) in the \( N \times 1 \) vector \( \mathbf{m} \) that minimize the \( L_2 \) objective function

\[
\epsilon = 1/2 \| \mathbf{Gm} - \mathbf{m}^{\text{img}} \|_2^2,
\]

where \( \mathbf{G} = \mathbf{L}^T \mathbf{L} \) is the symmetric \( N \times N \) Hessian matrix, \( \mathbf{L} \) is the forward modeling operator, and \( \mathbf{L}^T \) is the migration operator. Here, \( \mathbf{m}^{\text{img}} = \mathbf{L}^T \mathbf{d} \) is the migration image computed by migrating the recorded data \( \mathbf{d} \) with the migration operator \( \mathbf{L}^T \). Alternatively, the image-domain LSM problem can also be defined as finding \( \mathbf{m} \) that minimizes \( \epsilon = 1/2 \| \mathbf{m}^T \mathbf{Gm} - \mathbf{m}^T \mathbf{m}^{\text{img}} \|_2^2 \), which has a more well-conditioned solution than the one in equation 2 (Schuster, 2017). However, we will use equation 2 as the definition of the LSM problem to be consistent with the notation from Papyan et al. (2017a). The kernel associated with the Hessian matrix \( \mathbf{L}^T \mathbf{L} \) is also known as the point scatterer response of the migration operator or the migration Green’s function (Schuster and Hu, 2000). It is a square matrix that is assumed to be invertible; otherwise, a regularization term is incorporated into the objective function.
A formal solution to equation 2 is
\[ \mathbf{m}^* = \Gamma^{-1} \mathbf{m}^{\text{mig}}, \] (3)
where it is too expensive to directly compute the inverse Hessian \( \Gamma^{-1} \). Instead, a gradient method gives the iterative solution
\[ \mathbf{m}^{(k+1)} = \mathbf{m}^{(k)} - \alpha \Gamma^T (\Gamma \mathbf{m}^{(k)} - \mathbf{m}^{\text{mig}}), \] (4)
where \( \alpha \) is the step length, \( \Gamma \) is symmetric, and \( \mathbf{m}^{(k)} \) is the solution at the \( k \)th iteration. Typically, a regularization term is used to stabilize the solution, e.g., the sparse constraint that will be introduced in the next subsection.

### SLSM

The SLSM in the image domain is defined as finding the reflectivity coefficients \( m_i \) in the \( N \times 1 \) vector \( \mathbf{m} \) that minimize the objective function \( e \) (Perez et al., 2013):
\[ e = \frac{1}{2} \frac{||\Gamma \mathbf{m} - \mathbf{m}^{\text{mig}}||_2^2 + \lambda S(\mathbf{m})}{C_3}, \] (5)
where \( \Gamma = L' L \) represents the migration Green’s function (Schuster and Hu, 2000), \( \lambda > 0 \) is a positive scalar, \( \mathbf{m}^{\text{mig}} = L' \mathbf{d} \) is the migration image, and \( S(\mathbf{m}) \) is a sparseness function. For example, the sparseness function might be \( S(\mathbf{m}) = ||\mathbf{m}||_1 \) or \( S(\mathbf{m}) = \log(1 + ||\mathbf{m}||_2^2) \).

The solution that minimizes equation 5 is
\[ \mathbf{m}^* = \arg \min_{m} \left[ \frac{1}{2} \frac{||\Gamma \mathbf{m} - \mathbf{m}^{\text{mig}}||_2^2 + \lambda S(\mathbf{m})}{C_3} \right], \] (6)
which can be approximated by an iterative gradient-descent method:
\[ m_i^{(k+1)} = m_i^{(k)} - \alpha \frac{\Gamma^T (\Gamma m_i - m_i^{\text{mig}}) + \lambda S(m_i)'}{C_3}, \] (7)
Here, \( S(m_i)' \) is the derivative of the sparseness function with respect to the model parameter \( m_i \) and the step length is \( \alpha \). Vectors and matrices are denoted by boldface lowercase and uppercase letters, respectively. When \( S(\mathbf{m}) = ||\mathbf{m}||_1 \), the iterative solution in equation 7 can be recast as
\[ m_i^{(k+1)} = \text{soft} \left( \frac{m_i^{(k)} - \frac{1}{\alpha} \Gamma^T (\Gamma m_i^{(k)} - m_i^{\text{mig}})}{C_3} \right), \] (8)
where \text{soft} is the two-sided soft thresholding function (Papyan et al., 2016) derived in Appendix B (see equation B-4). Here, \( \Gamma = L' L \) is computed by solving the wave equation to get the forward modeled field and backpropagating the data by a numerical solution to the adjoint wave equation.

Equation 8 is similar to the forward modeling operation associated with the first layer of the neural network in Figure 1. That is, set \( \kappa = 0 \), \( \mathbf{m}^{(0)} = 0 \), \( \alpha = 1 \), and let the input vector be the scaled residual vector \( \mathbf{r} = -(\Gamma \mathbf{m}^{(0)} - \mathbf{m}^{\text{mig}}) = \mathbf{m}^{\text{mig}} \) so that the first-iterate solution can be compactly represented by
\[ \mathbf{m}^{(1)} = \text{soft}(\Gamma^T \mathbf{m}^{\text{mig}}, \lambda). \] (9)
Here, the input vector \( \mathbf{r} = \mathbf{m}^{\text{mig}} \) is multiplied by the matrix \( \Gamma^T \) to give \( \mathbf{z} = \Gamma^T \mathbf{r} \), and the elements of \( \mathbf{z} \) are then thresholded and shrunk to give the output \( \mathbf{m} = \text{soft}(\mathbf{z}, \lambda) \). If we impose a positivity constraint for \( \mathbf{z} \) and a shrinkage constraint so that \( \lambda \) is small, then the soft thresholding function becomes that of a one-sided threshold function, also known as the rectified linear unit or ReLU function. To simplify the notation, the \( \text{soft}(\mathbf{z}, \lambda) \) function is replaced by \( \sigma_i(\mathbf{z}) \) so that equation 9 is given by
\[ \mathbf{m}^{(1)} = \sigma_i(\Gamma^T \mathbf{m}^{\text{mig}}). \] (10)
For the ReLU function, there is no shrinkage so \( \lambda = 0 \). \( \Gamma \) in equation 10 is computed by forward and backward solutions to the wave equation. Unlike a neural network, the physics of wave propagation is included with the SLSM solution in equation 8.

### NNLSM

We now propose the neural network version of SLSM that finds the \( \Gamma^* \) and \( \mathbf{m}^* \) that minimize equation 5, which is equivalent to the convolutional sparse coding problem. We denote the optimal solution \( \mathbf{m}^* \) as the NNLSM image. Here, we assume that the migration image \( \mathbf{m}^{\text{mig}} \) can be decomposed into components that have the form \( \Gamma \mathbf{m}_i \), where \( \mathbf{m}_i \) represents a sparse quasi-reflectivity structure for the first CNN layer in Figure 1 and \( \Gamma_1 \) has a convolutional structure. The solution can be found by using the alternating direction method of multipliers method either in the Fourier domain (Heide et al., 2015) or in the space domain (Papyan et al., 2017b), which alternates between finding \( \Gamma^* \) (the dictionary learning problem) and then finding \( \mathbf{m}^* \) (the sparse-pursuit problem). The key difference between least-squares sparse inversion and NNLSM is that \( \Gamma^* \) is not computed by numerical solutions to the wave equation. Instead, the coefficients of \( \Gamma^* \) are computed by the usual gradient-descent learning algorithm of a CNN. For this reason, we denote \( \mathbf{m}^* \) as the quasi-reflectivity distribution and \( \Gamma^* \) as the quasi-migration Green’s function.

Appendix C provides the general solution for NNLSM for a single-layer neural network, where the optimal \( \Gamma^* \) is computed as the convolutions of the quasi-migration Green’s functions, which are denoted as convolutional

![Figure 1](https://example.com/figure1.png)

Figure 1. The forward modeling procedure for a multilayer CNN is equivalent to the multilayer sparse solution.

DOI:10.1190/geo2019-0412.1
filters in machine learning terminology (Liu and Schuster, 2018, 2019). Each filter is used to compute a feature map that corresponds to a subimage of quasi-reflection coefficients in the context of LSM.

We now compute the NNLSM image for a 1D model, where we assume \( \mathbf{m}^{\text{mig}} \) is an \( N \)-dimensional vector which can be expressed as

\[
\mathbf{m}^{\text{mig}} = \sum_{i=1}^{k_0} \gamma_i * \mathbf{m}'_{1i}. 
\] (11)

Here, \( \gamma_i \) is the \( i \)th local filter with the length of \( n_0 \), \( \mathbf{m}'_{1i} \) is the \( i \)th feature map, * denotes the convolution operator, and \( k_0 \) is the number of the filters. Alternatively, following Figure 2a, equation 11 can be written in matrix form as

\[
\mathbf{m}^{\text{mig}} = \mathbf{\Gamma}_0 \mathbf{m}_1 = \mathbf{\Gamma}_1^{(1)} \mathbf{m}'_{11} \quad \text{(Papyan et al., 2017a),}
\]

where \( \mathbf{\Gamma}_1 \) is a convolutional matrix with its columns consisting of the \( k_0 \) filters with all of their shifts. The term \( \mathbf{\Gamma}_1^{(1)} \) is a concatenation of banded and circulant (we shall assume throughout this paper that boundaries are treated by a periodic continuation, which gives rise to the cyclic structure) matrices, which is the same as \( \mathbf{\Gamma}_1 \) except that the order of the columns is different. The term \( \mathbf{m}'_{1i} \) is a concatenation of the feature map vectors \( \mathbf{m}'_{1i} \) onto a single long vector for \( i = 1, 2, \ldots, k_0 \).

The advantage of NNLSM is that only inexpensive matrix-vector multiplications are used and no expensive solutions to the wave equation are needed for backward and forward wavefield propagation. As will be seen later, convolutional filters that appear to be coherent noise can be excluded for denoising the migration image.

**Multilayer NNLSM**

Multilayer NNLSM is a natural extension of single-layer NNLSM. For NNLSM, the migration image \( \mathbf{m}^{\text{mig}} \) can be expressed as \( \mathbf{m}^{\text{mig}} = \mathbf{\Gamma}_1 \mathbf{m}_1 \) (Figure 2a), where there are \( k_0 \) filters in \( \mathbf{\Gamma}_1 \) and \( k_0 \) sub-quasi-reflectivity images in \( \mathbf{m}_1 \). Following Sulam et al. (2018), we can cascade this model by imposing a similar assumption on the sparse representation \( \mathbf{m}_1 \), i.e., \( \mathbf{m}_1 = \mathbf{\Gamma}_2 \mathbf{m}_2 \), for a corresponding convolutional matrix \( \mathbf{\Gamma}_2 \) with \( k_1 \) local filters and a sparse sub-quasi-reflectivity image \( \mathbf{m}_2 \), as depicted in Figure 2b. In this case, the filter size is \( n_1 \times k_0 \) and there are \( k_1 \) sub-quasi-reflectivity images in \( \mathbf{m}_2 \).

Similar to the derivation by Papyan et al. (2017a) and Sulam et al. (2018), the multilayer NNLSM problem is defined as the following:

Find: \( \mathbf{m}_i, \mathbf{\Gamma}_i \) such that

\[
\mathbf{m}_1 = \arg \min_{\mathbf{m}_1, \mathbf{\Gamma}_1} \frac{1}{2} \| \mathbf{\Gamma}_1 \mathbf{m}_1 - \mathbf{m}^{\text{mig}} \|_2^2 + \lambda S(\mathbf{m}_1),
\]

\[
\mathbf{m}_2^* = \arg \min_{\mathbf{m}_2} \frac{1}{2} \| \mathbf{\Gamma}_2 \mathbf{m}_2 - \mathbf{m}_1^* \|_2^2 + \lambda S(\mathbf{m}_2),
\]

\[
\vdots
\]

\[
\mathbf{m}_k^* = \arg \min_{\mathbf{m}_k} \frac{1}{2} \| \mathbf{\Gamma}_k \mathbf{m}_k - \mathbf{m}_{k-1}^* \|_2^2 + \lambda S(\mathbf{m}_k),
\] (12)

where \( \mathbf{\Gamma}_i \) is the \( i \)th Hessian matrix in the \( i \)th layer. The first iterate solution to the above system
of equations can be cast in a form similar to equation 10, except we have
\[ \mathbf{m}_N \approx \sigma_2 (\Gamma^T \sigma_2 (\Gamma^T \mathbf{m}^{\text{mig}})(\ldots)) \],
(13)
which is a repeated concatenation of the two operations of a multi-layered neural network: matrix-vector multiplication followed by a thresholding operation. In all cases, we use a CNN in which different filters are applied to the input from the previous layer to give feature maps associated with the next layer, as shown in Figure 1.

The migration image \( \mathbf{m}^{\text{mig}} \) can be approximated as \( \mathbf{m}^{\text{mig}} = \Gamma_1 \Gamma_2 \ldots \Gamma_i \mathbf{m}_N \). We refer to \( \Gamma(i) \) as the effective filter at the \( i \)th level,
\[ \Gamma(i) = \Gamma_1 \Gamma_2 \ldots \Gamma_i, \tag{14} \]
so that
\[ \mathbf{m}^{\text{mig}} = \Gamma(i) \mathbf{m}_i. \tag{15} \]

The next section tests the effectiveness of NNLSM on synthetic data and field data.

**NUMERICAL RESULTS**

We now present numerical simulations of NNLSM. Instead of only determining the optimal reflectivity \( \mathbf{m} \) as computed by SLSM, the NNLSM method computes the quasi-reflectivity \( \mathbf{m} \) and the elements of the Hessian matrix \( \Gamma = \mathbf{L}^T \mathbf{L} \). Each block of \( \Gamma \) is interpreted as the segment spread function (SSF) of the migration operator rather than the point spread function (PSF). If the actual Green’s functions are used to construct \( \Gamma \), then each column of the Hessian matrix is the point scatterer response of the migration operator (Schuster and Hu, 2000).

In contrast, the NNLSM Hessian is composed of blocks, where each block is the segment scatterer response of the migration operator. An example will be shown later where reflections from a segment of the reflector are migrated to give the migration segment response of the migration operator. The computational cost for computing SSFs is several orders of magnitude less than that for PSFs because no solutions to the wave equation are needed. However, the penalty is that the resulting solution \( \mathbf{m} \) is not the true reflectivity, but a sparse representation of it we denote as the quasi-reflectivity distribution.

Using the terminology of neural networks, we also can denote the sparse sub-quasi-reflectivity images as feature maps. Each block in \( \Gamma \) will be denoted as a filter. Therefore, the vector output of \( \Gamma \mathbf{m} \) can be interpreted as a sum of filter vectors \( \gamma_i \) weighted by the coefficients in \( \mathbf{m} \), where \( \gamma_i \) is the \( i \)th column vector of \( \Gamma \).

**Three-layer velocity model**

The interpretation of feature maps and filters can be understood by computing them for the Figure 3a model. The grid size of the model is 101 x 101, and the grid interval is 10 m in the \( x \)- and \( z \)-directions. There are 26 shots evenly spaced at a distance of 40 m on the surface, and each shot is recorded by 101 receivers with a sampling interval of 10 m. Figure 3b shows the reverse time migration (RTM) image.

The first test is for a 1D model in which we extract the image located at \( X = 0.5 \) km, which is displayed as the blue curve in Figure 3c. The red curve in Figure 3c represents the reflectivity model. Assume that there is only one filter in \( \Gamma \), and it extends over the depth of 400 m (41 grid points). We now compute the NNLSM image by finding the optimal \( \mathbf{m} \) and \( \Gamma \) by the two-step iterative procedure denoted as the alternating-descent method (Liu and Schuster, 2018 and Liu et al., 2018). The computed filter \( \gamma_i \) is shown in Figure 4a where the phase of the filter \( \gamma_i \) is nonzero. If we use a filter with a nonzero time lag to calculate its feature map \( \mathbf{m} \), the phases of the feature map and the true reflectivity \( \mathbf{m} \) will be different. So we need to modify the time lag and polarity of the basis function \( \Gamma_1 \). The modified basis function, i.e. filter, is shown in Figure 4b, and its coefficients are displayed as the blue curve in Figure 4c. Compared with the true reflectivity \( \mathbf{m} \) (the red curve in

![Figure 4](image_url) (a) Learned filter, (b) modified filter, and (c) feature map (blue) and reflectivity model (red).

![Figure 5](image_url) (a) Learned and (b) modified filters.
Figure 6. Feature maps (a) 1, (b) 2, and (c) 3 for the filters shown in Figure 5. The stacked feature map is shown in (d). Here, the white lines show the locations of nonzero points and the yellow lines indicate the locations of the reflectivity distributions.

Figure 7. (a) The 2D SEG/EAGE salt model, (b) RTM image, (c-e) learned effective filters $\Gamma_1$, $\Gamma_2$, and $\Gamma_3$, (f-h) stacked quasi-reflectivity coefficients for $\mathbf{m}_1$, $\mathbf{m}_2$, and $\mathbf{m}_3$, and (i-k) reconstructed migration images $\Gamma_1 \mathbf{m}_1$, $\Gamma_2 \mathbf{m}_2$, and $\Gamma_3 \mathbf{m}_3$. (see Figure 5a). The modified filters are shown in Figure 5b. Appendix D describes how we align the filters by using the cross-correlation method. The feature maps of these three filters are displayed in Figure 6a–6c. Figure 6d shows the sum of these three feature maps. It is evident that the stacked feature maps can estimate the correct locations of the reflectivity spikes.

SEG/EAGE salt model

The multilayer NNLSM procedure (see equation 12) is now applied to the migration image associated with the 2D SEG/EAGE salt velocity model in Figure 7a. The grid size of the model is 101 grid points in the $z$- and $x$-directions. The grid interval is 40 m in the $x$-direction and 20 m in the $z$-direction. Figure 7b shows the RTM image. The multilayer NNLSM consists of three convolutional layers: The first one contains 15 basis functions, i.e., filters, of size $11 \times 11$ grid points; the second one consists of 15 basis functions with dimensions $11 \times 11 \times 15$; and the third one contains 15 basis functions with dimensions $11 \times 11 \times 15$. Equation 12 is solved for $\mathbf{m}_i$ and $\Gamma_i$ ($i \in 1, 2, 3$) by the iterative alternating-descent method. The multilayered structure is shown in Figure 8, where the black points in $\mathbf{m}_i$ represent the nonzero values of the quasi-reflectivity distribution. The effective basis functions computed for these layers are shown in Figure 7c–7e, where the yellow, red, and green boxes indicate the sizes of the basis functions, which can be considered as quasi-migration Green’s functions. It indicates that the basis functions of the first layer $\Gamma_1$ contain very simple small-dimensional edges, which are called “atoms” by Sulam et al. (2018). The nonzeros of the second group of basis functions $\Gamma_2$ combine a few atoms from $\Gamma_1$ to create slightly more complex edges, junctions, and corners in the effective basis functions in $\Gamma_2$. Finally, $\Gamma_3$ combines molecules from $\Gamma_3$ to reconstruct the more complex parts of the migration image. The corresponding stacked coefficient images, also known as feature maps, are shown in Figure 7f–7h, which give the quasi-reflectivity distributions. The reconstructed migration images are shown in Figure 7i–7k.

For comparison, we computed the standard LSM image using the deblurring method described in Chen et al. (2017, 2019). Here, the deblurring filter size is $17 \times 17$ grid points (the black boxes in Figure 9) computed for a $50 \times 50$ grid (the red boxes in Figure 9) of evenly spaced point scatterers with the same migration velocity model as that used for the data migration in Figure 7a. The standard LSM images for the 1st and 50th iterations are shown in Figure 10b and 10e, respectively, next to the NNLSM image in Figure 10d. It is clear that the NNLSM image is better resolved than the LSM image, although there are discontinuities in some of the NLNSM interfaces not seen in the LSM image. Some of the detailed geology is lost in the NNLSM image as seen in the wiggly interface in the red rectangular area of Figure 10. The practical application of the NLNSM image is that it might serve as a superresolved attribute image that can be combined with other attributes to delineate the geology. For example, combining the depth slice...
of the NNLSM image with a spectral decomposition image (Aarre, 2016) can help delineate the lithologic edges of meandering channels.

NNLSM can filter out random and coherent noise in the migration image by eliminating the noisy learned basis functions and their coefficients in the NNLSM image. For example, Figure 11a shows the RTM image with a sparse acquisition geometry so that the image contains a strong acquisition footprint. The reconstructed migration image in Figure 11b shows significant mitigation of this noise in Figure 11a. However, the migration swing noise is still prominent near the red arrows in Figure 11b. Such noise is reconstructed from the noisy basis function shown in Figure 12a and the coefficients in Figure 12b.

Figure 12c is the image reconstructed by correlating the basis function in Figure 12a with the coefficients in Figure 12b. After filtering out the basis functions from the noise, the reconstructed image is shown in Figure 11c, which is free from aliasing noise at the locations indicated by the red arrows.

**North Sea data**

We apply the NNLSM method to field data collected in the North Sea (Schroot and Scüttenhelm, 2003), where the time migration image is shown in Figure 13a. The time axis is gridded with 213 evenly spaced points, and there are 301 grid points along the x-axis. We compute 2113 × 5 (grid point) convolutional basis functions, i.e., filters γi for (i = 1, 2, …, 21), by the NNLSM procedure (see Figure 13b). These filters approximate the dip-filtered migration Green’s functions, and the basis functions are marked as the yellow boxes in Figure 13a and 13b. The stacked feature maps (quasi-reflectivity distribution) are displayed in Figure 13c. It is evident that the stacked feature maps can provide a high-resolution migration image. After reconstruction from the learned filters and feature maps, the migration image is shown in Figure 13d with less noise.

Finally, we apply NNLSM to a time slice of the migration image, which is shown in Figure 14a, where the image size is 301 by 301 grid points. Figure 14b shows the 2113 × 5 filters estimated by the NNLSM procedure. The stacked feature map is displayed in Figure 14c, which could be used as an attribute image for high-resolution delineation of geologic bodies. The reconstructed migration image is shown in Figure 14d, and we can see that there is less noise.

**DISCUSSION**

The forward modeling for a multilayered neural network is shown to be equivalent to a single-iterate solution of a multilayered LSM.
problem. This assumes positivity and shrinkage constraints on the soft thresholding operation, so the activation function reduces to the ReLU function. This equivalence relates the physics of seismic imaging to architectural features of the neural network.

The size of the filters in the first layer should be approximately the same size as the migration Green’s function for that model. Experiments with numerical models suggest that this size is approximately one to two wavelengths. In this case, the filter is interpreted as an approximation to the migration Green’s function, except it is that for a reflecting segment. Thus, we interpret the approximate migration Green’s function as a migration SSF rather than a migration PSF. Sulam et al. (2018) classify each feature in the first layer as an atom that takes on the role of an SSF.

The output of the first layer provides the small-scale, i.e., high-wavenumber, features associated with the input data. For an input migration image, the feature maps of the first layer resemble sub-quasi-reflectivity maps of the subsurface. Adding the sub-quasi-reflectivity maps together gives a close approximation to the actual reflectivity model as shown in Figures 6d and 7f.

The output of the second layer is a weighted sum of the first-layer features, which create sparser feature maps. Sulam et al. (2018) classifies the concatenation of the filters from the first and second layers as analogous to the creation of molecules from different atoms (see equation 14). In the migration problem, the resulting filters are SSFs for even larger segments of the original reflector boundaries. The feature maps of the third layer are a weighted sum of the second layer’s features to produce even sparser feature maps. For migration, the final feature maps are very sparse whereas the concatenated filters are associated with large-scale reflectivity structures in the migration image.

The computational cost of computing NNLSM images is significantly less than that for LSM images because no solutions of the wave equation are needed. For example, we consider the 2D FDTD forward modeling of the acoustic equation with an eighth-order scheme in space and a second-order scheme in time, and its computational complexity is $O(n_t n^2)$ for one shot, where $n$ is the number of grid points in one direction, and $n_t$ is the number of the time steps. According to Plessix (2007), the number of time steps $n_t$ is approximately equal to 30$n$ in order to satisfy the stability condition and to make sure that there is enough recording time when $v_{max}/v_{min} \approx 3$; here, $v_{max}$ and $v_{min}$ are the maximum and minimum velocities, respectively. So, the complexity of 2D FDTD forward
modeling of the acoustic equation is $O(n_3 n^3)$, where $n_3$ is the number of shots. The computational complexity of LSRTM is $O(b n^3 N_{\text{iter}} n^3)$ (Schuster, 2017), where $N_{\text{iter}}$ is the iteration number. For NNLSM, the complexity is $O(N_{\text{iter}} n^2 \log n)$ according to Heide et al. (2015) and can be reduced to $O(N_{\text{iter}} n^2)$ if a local block coordinate-descent algorithm is used (Zisselman et al., 2019).

NNLSM can have more than one filter and the filters can be 2D or 3D filters. We show the relationship among BD, SLSM, NNLSM, and CNN in Figure 15. NNLSM is an unsupervised learning method. Compared to a supervised learning method, it does not heavily depend on the huge amount of training data. However, it may need human intervention for inspecting the quality of the quasi-reflectivity images and we need to align the filters to get a more accurate estimate of the reflectivity distribution.

In Figure 14, we apply 2D NNLSM to a time slice of the 3D North Sea data. The 2D NNLSM method can be extended to a 3D implementation to learn a set of 3D filters. Incorporating the third dimension of information from 3D filters will likely lead to a better continuity of the reflectors in Figure 14c. However, the computational cost will increase by more than a multiplicative factor of $n$.

**CONCLUSIONS**

NNLSM finds the optimal quasi-reflectivity distribution and quasi-migration Green’s functions that minimize a sum of migration misfit and sparsity regularization functions. The advantages of NNLSM over standard LSM are that its computational cost is significantly less than that for LSM and that it can be used for filtering coherent and incoherent noise in migration images. A practical application of the NNLSM image is as an attribute map that provides superresolution imaging of the layer interfaces. This attribute image can be combined with other attributes to delineate the structure and lithology in depth/time slices of migration images. Its disadvantage is that the NNLSM quasi-reflectivity image is only an approximation to the actual reflectivity distribution.

A significant contribution of our work is that we show that the filters and feature maps of a multilayered CNN are analogous to migration Green’s functions and reflectivity distributions. We now have a physical interpretation of the filters and feature maps in deep CNN in terms of the operators for seismic imaging. Such an understanding has the potential to lead to better architecture design of the network and extend its application to waveform inversion. In answer to Donoho’s plea for more rigor, NNLSM represents a step forward in establishing the mathematical foundation of CNN in the context of LSM.

**ACKNOWLEDGMENTS**

The research reported in this publication was supported by the King Abdullah University of Science and Technology (KAUST) in Thuwal, Saudi Arabia. We are grateful to the sponsors of the Center for Subsurface Imaging and Modeling Consortium for their...
Data and Materials Availability

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

Appendix A

Migration Green’s Function

Schuster and Hu (2000) show that the poststack migration (Yilmaz, 2001) image \( m(x)_{\text{mig}} \) in the frequency domain is computed by weighting each reflectivity value \( m(z) \) by \( \Gamma(x|z) \) and integrating over the model-space coordinates \( z \):

\[
m(x) = \eta \int_{D_{\text{model}}} d\Gamma(z) \sum_{y \in D_{\text{data}}} dy \omega^4 G(x|y)^2 G(y|z)^2 m(z),
\]

where \( \eta \) represents terms such as the frequency variable raised to the fourth power. The migration Green’s function \( \Gamma(x|z) \) is given by

\[
\text{for } x, z \in D_{\text{model}}: \Gamma(x|z) = \eta \sum_{y \in D_{\text{data}}} dy \omega^4 G(x|y)^2 G(y|z)^2.
\]

APPENDIX B

Soft Thresholding Function

Define the sparse inversion problem as finding the optimal value \( x^* \) that minimizes the objective function

\[
e = \frac{1}{2} ||z - x||_2^2 + \lambda ||x||_1,
\]

where the \( L_1 \) norm demands sparsity in the solution \( x \). An example is where \( z \) is a noisy \( M \times N \) image such that \( z = x + \text{noise} \) and we seek the optimal vector \( x \) that satisfies equation B-1. Here, the noisy \( M \times N \) image has been flattened into the tall \( MN \times 1 \) vector \( z \). The stationary condition for equation B-1 is

\[
\frac{\partial e}{\partial x_i} = (x_i - z_i) + \lambda \frac{\partial ||x||_1}{\partial x_i} = 0,
\]

where

\[
\frac{\partial ||x||_1}{\partial x_i} = 1 \text{ for } x_i \geq 0; \frac{\partial ||x||_1}{\partial x_i} = -1 \text{ for } x_i < 0.
\]

Equations B-2–B-3 can be combined to give the optimal \( x^* \) expressed as the two-sided ReLU function

\[
x_i = \text{soft}(z_i, \lambda) = \begin{cases} 
    z_i - \lambda & \text{if } z_i > \lambda \\
    0 & \text{if } |z_i| \leq \lambda \\
    z_i + \lambda & \text{if } z_i < -\lambda 
\end{cases}.
\]

More generally, the iterative soft-thresholding algorithm (ISTA) that finds \( x^* \)

\[
x^* = \arg \min_x \left[ \frac{1}{2} ||z - Wx||_2^2 + \lambda ||x||_1 \right],
\]

is

\[
x^{(k+1)}_i = \text{soft}(x^{(k)}, \frac{1}{\alpha} W^T(Wx^{(k)} - z), \frac{\lambda}{\alpha}),
\]

There are several more recently developed algorithms that have faster convergence properties than ISTA. For example, fast-ISTA has quadratic convergence (Beck and Teboulle, 2009).

Appendix C

Neural Network Least-Squares Migration

The NNLSM algorithm in the image domain is defined as solving for the basis functions \( \tilde{\Gamma}(x_i|z_j) \) and \( \tilde{m}_j \) that minimize the objective function defined in equation 5. In contrast, SLSM finds only the
LSTM image in the image domain and uses the precomputed migration Green’s functions that solve the wave equation.

The NNLSM solution is defined as

\[
\langle \tilde{m}^*, \tilde{F}^* \rangle = \arg \min_{\tilde{m}} \left[ \frac{1}{2} \| \tilde{F} \tilde{m} - \tilde{m}^{\text{mig}} \|_2^2 + \lambda S(\tilde{m}) \right],
\]

(C-1)

where now \( \tilde{F}^* \) and \( \tilde{m}^* \) are to be found. The functions with tildes are mathematical constructs that are not necessarily identical to those based on the physics of wave propagation as in equation 5.

The explicit matrix-vector form of the objective function in equation C-1 is given by

\[
e = \frac{1}{2} \sum_i \sum_j \tilde{F}(x_i | z_j) \tilde{m}_j - m_i^{\text{mig}}_i^2 + \lambda S(\tilde{m}),
\]

(C-2)

and its derivative with respect to \( \tilde{F}(x_i | z_j) \) is given by

\[
\frac{\partial e}{\partial \tilde{F}(x_i | z_j)} = \sum_j (\tilde{F}(x_i | z_j) \tilde{m}_j - m_i^{\text{mig}}) \tilde{m}_j',
\]

(C-3)

The iterative solution of equation C-1 is given in two steps (Olshausen and Field, 1996):

1. Iteratively estimate \( \tilde{m}_i \) by the gradient-descent formula used with SLSM:

\[
\tilde{m}_{i}^{(k+1)} = \tilde{m}_i^{(k)} - \alpha [\tilde{F}^T (\tilde{F} \tilde{m} - \tilde{m}^{\text{mig}})]_i - \lambda S(\tilde{m})_j'.
\]

(C-4)

2. Update the basis functions \( \tilde{F}(x_i | z_j) \) by inserting equation C-3 into the gradient-descent formula to get

\[
\tilde{F}(x_i | z_j)^{(k+1)} = \tilde{F}(x_i | z_j)^{(k+1)} - \alpha \frac{\partial e}{\partial \tilde{F}(x_i | z_j)} = \tilde{F}(x_i | z_j)^{(k+1)}
\]

\[
- \alpha \left( \sum_j \tilde{F}(x_i | z_j) \tilde{m}_j - m_i^{\text{mig}} \right) \tilde{m}_j',
\]

(C-5)

It is tempting to think of \( \tilde{F}(x'x') \) as the migration Green’s function and \( \tilde{m}_i \) as the component of reflectivity. However, we can’t submit to this temptation and so we must consider, unlike in the SLSM algorithm, that \( \tilde{F}(x'x') \) is a sparse basis function and \( \tilde{m}_i \) is its coefficient. To get the true reflectivity, we should equate equation A-3 to \( \sum_j \tilde{F}(x_i | z_j) \tilde{m}_j \) and solve for \( m_j \).

**APPENDIX D**

**ALIGNMENT OF THE FILTERS**

To align the learned filters, we first choose a “target” filter, which is denoted as a 2D matrix \( \mathbf{A} \) with the size of \( M \times N \). Then, we try to align all of the other filters with the target filter through their cross-correlation. For example, if we choose one filter denoted as matrix \( \mathbf{B} \) with the same size as \( \mathbf{A} \), we can get the cross-correlation matrix \( \mathbf{C} \) with its elements defined as

\[
C_{i+M,j+N} = \sum_{m=1}^{M} \sum_{j=1}^{N} d_{m,n} \cdot b_{m+i,n+j},
\]

(D-1)

where \( -M < i < M \) and \( -N < j < N \). The term \( a_{i,j} \), \( b_{i,j} \), and \( C_{i,j} \) indicate the element at position \( (i,j) \) in matrices \( \mathbf{A}, \mathbf{B}, \) and \( \mathbf{C} \), respectively. The location of the maximum absolute value of the elements in matrix \( \mathbf{C} \) indicates how much we should shift filter \( \mathbf{B} \) to filter \( \mathbf{A} \) in each direction. Figure 16 shows the calculation of the cross-correlation matrix \( \mathbf{C} \) for two filters \( \mathbf{A} \) and \( \mathbf{B} \). The term \( c_{i,0} \) (or \( c_{0,j} \)) is the maximum absolute value of the elements in matrix \( \mathbf{C} \), which indicates that filter \( \mathbf{B} \) should be shifted one position along the first direction. Here, we need to pad zeros along all the dimensions of filter \( \mathbf{B} \) before shifting it, which is displayed in Figure 17.

Figure 18a shows the learned filters with a size of \( 17 \times 9 \) from the migration image of the SEG/EAGE salt model. Filter No. 7 (the yellow box) is chosen as the target filter. The aligned filters are...
shown in Figure 18b, and the stacked feature maps from the original and aligned filters are displayed in Figure 18c and 18d, respectively. It is evident that the reflector interfaces from the aligned filters are more continuous especially in the red box compared with those from the original filters.

REFERENCES


Schuster, G. T., 2017, Seismic inversion: SEG.


Yilmaz, O., 2001, Seismic data analysis: Processing, inversion, and interpretation of seismic data: SEG.


