ABSTRACT

We recast the multilayered sparse inversion problem as a multilayered neural network problem. Unlike standard least squares migration (LSM) which finds the optimal reflectivity image, neural network least squares migration (NNLSM) finds both the optimal reflectivity image and the quasi-migration-Green’s functions. These quasi-migration-Green’s functions are also denoted as the convolutional filters in a convolutional neural network (CNN) and are similar to migration Green’s functions. We show that the CNN filters and feature maps are directly related to the migration Green’s functions and reflectivity distributions. Thus, we provide for the first time a physical interpretation of the filters and feature maps in deep CNN in terms of the operators for seismic imaging.

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 reflectivity image is only an approximation to the actual reflectivity distribution. However, the approximate reflectivity image can be used as a superresolution attribute image for high-resolution delineation of geologic bodies.
INTRODUCTION

The biological processing of information in a cat’s brain can be mathematically approximated as a weighted summation of electro-chemical input values into a vertical layer of neurons, followed by a thresholding operation [Hubel and Wiesel, 1962]. Thresholding simplifies the amount of information to be processed by eliminating unimportant input values that fall below a certain threshold. This pair of operations, weighted summation and thresholding of the input values into the first layer of neurons, leads to new inputs inserted into a second layer of neurons. These new inputs are reweighted, summed, and thresholded to be injected into the second column of neurons. This process is repeated again and again to form a multilayered neural network. Such networks are now used in many areas to automatically classify and make decisions about large data sets [LeCun et al., 2015].

Trial-and-error experimentation with neural networks over several decades eventually generated the architecture known as deep convolutional neural networks (CNNs). The success of the current CNN architectures is evidenced by their practical applications with self-driving cars [Hadsell et al., 2009; Tian et al., 2018], image classification [Krizhevsky et al., 2012; He et al., 2016], data retrieval from digital archives [Shen et al., 2014], and medical diagnosis from a combination of body images generated by MRIs, CAT scans, and PET scans [Bakator and Radosav, 2018].

Until recently, the design of effective CNN architectures was largely based on heuristic experimentation. This shortcoming largely results from the absence of a rigorous mathematical foundation for neural networks in general, and CNN in particular. In 2016, Elad and his coauthors proposed that the CNN problem could be recast as finding the sparsest model $m$ under the $L_1$ norm subject to honoring the data misfit constraint $||\Gamma m - m^{\text{mig}}||_2^2 \leq \beta$ [Papyan et al., 2016; Elad, 2018]. For least squares migration with a neural network [Liu and Schuster, 2018], this problem is defined in the following way.

\[
\begin{align*}
\text{Given :} & \quad \Gamma, m^{\text{mig}} \quad \text{and} \quad \Gamma m = m^{\text{mig}} + \text{noise}, \\
\text{Find :} & \quad m^* = \arg \min_{m} ||m||_1 \\
& \quad \text{subject to} \quad ||\Gamma m - m^{\text{mig}}||_2^2 \leq \beta
\end{align*}
\]

where $m$ is an $N \times 1$ input real-valued vector, $m^{\text{mig}}$ is the migration image computed from the seismic data, and $\Gamma$ is an $N \times N$ matrix of real-valued weights that can be related to the values of the migration Green’s function [Schuster and Hu, 2000]. 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, where each layer consists of a concatenation of a weighted summation of input values to give the vector $z$ followed by a two-sided soft thresholding operation denoted as $\sigma(z)$ (Elad, 2010).

We now show that the sparse solution to the least squares migration problem reduces to the forward modeling operations of a multilayered neural network. Instead of just finding the optimal reflectivity $m^*$, we optimize for both the reflectivity $m$ and the quasi-migration-Green’s functions $\Gamma$. 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 convolutional neural network. As discussed in Appendix 1, the migration Green’s function is the point scatterer response of the migration operator. The final image is denoted as the NNLSM estimate of the reflectivity distribution that honors the $L_1$
sparsity condition. The next section shows the connection between the multilayer neural network and the solution to the multilayer LSM problem. This is followed by the numerical examples with the synthetic models and field data from the North Sea.

**THEORY OF NEURAL NETWORK LEAST SQUARES MIGRATION**

The theory of standard least squares migration is first presented to establish the benchmark solution where the optimal reflectivity function minimizes the image misfit under the $L_2$ norm. This is then followed by the derivation of the sparse least squares migration (SLSM) solution for a single-layer network. The final subsection derives the NNLSM solution for a multilayer network.

**Least Squares Migration**

The least squares migration (LSM) problem can be defined (Schuster and Hu, 2000; Schuster, 2017) as finding the reflectivity coefficients $m_i$ in the $N \times 1$ vector $m$ that minimize the $L_2$ objective function $\epsilon = 1/2 ||\Gamma m - m^{mig}||_2^2$,

$$m^* = \arg\min_m \left\{ \frac{1}{2} ||\Gamma m - m^{mig}||_2^2 \right\},$$

(2)

where $\Gamma = L^T L$ is the symmetric $N \times N$ Hessian matrix, $L$ is the forward modeling operator, and $L^T$ is the migration operator. Here, $m^{mig} = L^T d$ is the migration image computed by migrating the recorded data $d$ with the migration operator $L^T$. Alternatively, the LSM problem in the image domain can also be defined as finding $m$ that minimizes $\epsilon = 1/2(m^T \Gamma m - m^T m^{mig})$, 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 in order to be consistent with the notation from Papan et al. (2016) and Elad (2018). The kernel associated with the Hessian matrix $L^T L$ 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

$$m^* = \Gamma^{-1} m^{mig},$$

(3)

where it is too expensive to directly compute the inverse Hessian $\Gamma^{-1}$. Instead, a gradient method gives the iterative solution

$$m^{(k+1)} = m^{(k)} - \alpha \Gamma^T (\Gamma m^{(k)} - m^{mig}),$$

(4)

where $\alpha$ is the step length, $\Gamma$ is symmetric, and $m^{(k)}$ is the solution at the $k^{th}$ iteration.

**Sparse Least Squares Migration**

The sparse least squares problem (SLSM) is defined as finding the reflectivity coefficients $m_i$ in the $N \times 1$ vector $m$ that minimize the objective function $\epsilon$ (Perez et al., 2013):

$$\epsilon = \frac{1}{2} ||\Gamma m - m^{mig}||_2^2 + \lambda S(m),$$

(5)
where \( \bm{\Gamma} = \bm{L}^T \bm{L} \) represents the migration Green’s function (Schuster and Hu, 2000), \( \lambda > 0 \) is a positive scalar, \( \bm{m}^{mig} = \bm{L}^T \bm{d} \) is the migration image, and \( S(\bm{m}) \) is a sparseness function. For example, the sparseness function might be \( S(\bm{m}) = ||\bm{m}||_1 \) or \( S(\bm{m}) = \log(1 + ||\bm{m}||_2^2) \).

**Single-Layer Sparse LSM**

The solution to equation 5 is

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

(6)

which can be approximated by an iterative gradient descent method:

\[
m_i^{(k+1)} = m_i^{(k)} - \alpha \left[ \Gamma^T \left( \Gamma \bm{m}^{(k)} - \bm{m}^{mig} \right) + \lambda S' \left( \bm{m} \right) \right],
\]

(7)

Here, \( S' \) 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, respectively, denoted by boldface lowercase and uppercase letters.

When \( S(\bm{m}) = ||\bm{m}||_1 \), the iterative solution in equation 7 can be recast as

\[
m^{(k+1)} = \text{soft} \left( \left[ \bm{m}^{(k)} - \frac{1}{\alpha} \Gamma^T \left( \Gamma \bm{m}^{(k)} - \bm{m}^{mig} \right) \right], \frac{\lambda}{\alpha} \right),
\]

(8)

where, \( \text{soft} \) is the 2-sided soft thresholding function (Elad, 2010) derived in Appendix 2 (see equation 22).

Equation 8 is similar to the forward modeling operation associated with the first layer of the neural network in Figure 1. That is, set \( k = 0, \bm{m}^{(0)} = 0 \), and let the input vector be the scaled residual vector \( \bm{r} = -(\bm{\Gamma} \bm{m}^{(0)} - \bm{m}^{mig}) = \bm{m}^{mig} \) so that the first-iterate solution can be compactly represented by

\[
\bm{m}^{(1)} = \text{soft} \left( \Gamma^T \bm{m}^{mig}, \lambda \right),
\]

(9)

where \( \alpha = 1 \). Here, the input vector \( \bm{r} = \bm{m}^{mig} \) is multiplied by the matrix \( \Gamma^T \) to give \( \bm{z} = \Gamma^T \bm{r} \), and the elements of \( \bm{z} \) are then thresholded and shrunk to give the output \( \bm{m} = \text{soft} \left( \bm{z}, \lambda \right) \). If we impose a positivity constraint for \( \bm{z} \) and a shrinkage constraint so \( \lambda \) 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 \( \text{soft} \left( \bm{z}, \lambda \right) \) function or ReLu(\( \bm{z} \)) function is replaced by \( \sigma_\lambda(\bm{z}) \) so that equation 9 is given by

\[
\bm{m}^{(1)} = \sigma_\lambda(\Gamma^T \bm{m}^{mig}),
\]

(10)

For the ReLu function there is no shrinkage so \( \lambda = 0 \).

We now propose the neural network version of sparse least squares migration that finds both \( \bm{\Gamma}^* \) and \( \bm{m}^* \) which minimize equation 5 which is equivalent to the convolutional sparse coding (CSC) problem. We denote the optimal solution \( \bm{m} \) as the neural network least squares migration (NNLSM) image. Here, we assume that the migration image \( \bm{m}^{mig} \) can be decomposed into components that have the form \( \bm{\Gamma}_1 \bm{m}_1 \), where \( \bm{m}_1 \) represents a sparse
reflectivity structure for the 1\textsuperscript{st} 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 (ADMM) method either in the Fourier domain (Heide et al., 2015) or in the space domain (Papyan et al., 2017b), which alternates between finding $\Gamma^*$ (dictionary learning problem) and then finding $m^*$ (sparse pursuit problem).

Appendix 3 shows the general solution for NNLSM for a single-layer neural network, where the optimal $\Gamma^*$ is composed of the approximate migration Green’s functions, which are denoted as convolutional filters in the machine learning terminology (Liu and Schuster, 2018). Each filter is used to compute a feature map that corresponds to a sub-image of reflection coefficients in the context of LSM.

We now compute the NNLSM image for a 1D model, where we assume $m_{\text{mig}}$ is a $N$ dimensional vector which can be expressed as,

$$m_{\text{mig}} = \sum_{i}^{\text{k}_0} \gamma_i \ast m_{1i}'.$$

Here, $\gamma_i$ is the $i$\textsuperscript{th} local filter with length of $n_0$, $m_{1i}'$ is the $i$\textsuperscript{th} feature map, “\ast” denotes the convolution operator and $\text{k}_0$ is the number of the filters. Alternatively, following Figure 2\textsuperscript{a}, equation 11 can be written in matrix form as $m_{\text{mig}} = \Gamma_1 m_1 = \Gamma_1' m_1'$ (Papyan et al., 2017a), where $\Gamma_1$ is a convolutional matrix containing in its columns the $\text{k}_0$ filters with all of their shifts. $\Gamma_1'$ is a concatenation of banded and circulant matrices, which is the same as $\Gamma_1$ except that the order of the columns is different. $m_1'$ is a concatenation of the feature map vectors $m_{1i}'$ for $i = 1, 2, \ldots, \text{k}_0$.

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

\footnote{We shall assume throughout this paper that boundaries are treated by a periodic continuation, which gives rise to the cyclic structure.}
propagation of the wavefield. As will be seen later, convolutional filters that appear to be coherent noise can be excluded for denoising the migration image.

Figure 2: (a) Single-layer NNLSM and (b) multilayer NNLSM for a one-dimensional migration image $\mathbf{m}^{\text{mig}}$.

**Multilayer Neural Network LSM**

The multilayer NNLSM is a natural extension of the single-layer NNLSM. For NNLSM, the migration image $\mathbf{m}^{\text{mig}}$ can be expressed as $\mathbf{m}^{\text{mig}} = \Gamma_1 \mathbf{m}_1$ (Figure 2a), where there are $k_0$ filters in $\Gamma_1$ and $k_0$ sub-reflectivity images in $\mathbf{m}_1$. Following Elad (2018), we can cascade this model by imposing a similar assumption to the sparse representation $\mathbf{m}_1$, i.e., $\mathbf{m}_1 = \Gamma_2 \mathbf{m}_2$, for a corresponding convolutional matrix $\Gamma_2$ with $k_1$ local filters and a sparse sub-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-reflectivity images in $\mathbf{m}_2$. 

6
Similar to the derivation by Elad (2018) and Panyan et al. (2017a), the multilayer neural network LSM problem is defined as the following.

Find: \( m_i, \Gamma_i \) such that

\[
\begin{align*}
\mathbf{m}_1^* &= \arg \min_{\mathbf{m}_1, \Gamma_1} \left[ \frac{1}{2} ||\Gamma_1 \mathbf{m}_1 - \mathbf{m}_{\text{mig}}||^2_2 + \lambda S(\mathbf{m}_1) \right], \\
\mathbf{m}_2^* &= \arg \min_{\mathbf{m}_2, \Gamma_2} \left[ \frac{1}{2} ||\Gamma_2 \mathbf{m}_2 - \mathbf{m}_1^*||^2_2 + \lambda S(\mathbf{m}_2) \right], \\
&\vdots \\
\mathbf{m}_N^* &= \arg \min_{\mathbf{m}_N, \Gamma_N} \left[ \frac{1}{2} ||\Gamma_N \mathbf{m}_N - \mathbf{m}_{N-1}^*||^2_2 + \lambda S(\mathbf{m}_N) \right],
\end{align*}
\]

(12)

where \( \Gamma_i \) is the \( i \)th Hessian matrix in the \( i \)th layer and \( \Gamma \) is the matrix that contains all of the sub-block matrices \( \Gamma_i \). The first iterate solution to the above system of equations can be cast in a form similar to equation 10, except we have

\[
\mathbf{m}_N^* \approx \sigma_\lambda \left( \Gamma_N^T \sigma_\lambda(\Gamma_N^T \cdots \sigma_\lambda(\Gamma_1^T \mathbf{m}_{\text{mig}}) \cdots) \right),
\]

(13)

which is a repeated concatenation of the two operations of a multilayered neural network: matrix-vector multiplication followed by a thresholding operation. In all cases we use a convolutional neural network where 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.

For a perfect prediction of the migration image, \( \mathbf{m}_{\text{mig}} \) can also be approximated as \( \mathbf{m}_{\text{mig}} = \Gamma_1 \Gamma_2 \ldots \Gamma_N \mathbf{m}_N \). We refer to \( \Gamma^{(i)} \) as the effective filter at the \( i \)th level,

\[
\Gamma^{(i)} = \Gamma_1 \Gamma_2 \ldots \Gamma_i,
\]

(14)

so that

\[
\mathbf{m}_{\text{mig}} = \Gamma^{(i)} \mathbf{m}_i.
\]

(15)

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 both \( \mathbf{m} \) and the elements of the Hessian matrix \( \Gamma = \mathbf{L}^T \mathbf{L} \). Each block of \( \Gamma \) is considered to be the segment response function (SSF) of the migration operator rather than the point spread function (PSF). If the actual Green’s functions is 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 a segment of the reflector is migrated to give the migration segment response of the migration operator. The computational cost for computing SSF’s is several orders of magnitude less than that for PSFs because no solutions to the wave equation are needed.

Using the terminology of neural networks, the sparse sub-reflectivity image are also denoted 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 \( \Gamma \).
Figure 3: a) Three-layer velocity model, b) RTM image, and c) migration image (blue curve) at X=0.5 km, where the red curve is the normalized reflectivity model.
Figure 4: a) Learned filter, b) modified filter, and c) feature map (blue) and reflectivity model (red).
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 $101 \times 101$, and the grid interval is 10 m in both 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 show the reverse time migration (RTM) image.

The first test is for a 1D model where 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 is 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 $m$ and $\Gamma$ by the two-step iterative procedure denoted as the alternating descent method (see 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 non-zero-phase filter to calculate its feature map $m$, the phases of the feature map and the true reflectivity $m$ will be different. So, we need to modify the phase and polarity of the basis function $\tilde{\gamma}_i$. The modified basis function is shown in Figure 4b, and its coefficients are displayed as the blue curve in Figure 4b. Compared with the true reflectivity $m$ (red curve in Figure 4), the feature map can give the correct positions but also give the wrong values of the reflectivity distribution.

Next, we perform a 2D test where the input is the 2D migration image in Figure 3b. Three 35-by-35 (grid point) filters are learned (see Figure 5a). The modified filters are shown in Figure 5b. Appendix 4 describe how we align the filters by using the cross-correlation method. The feature maps of these three filters are displayed in Figures 6a, b and c. 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-direction and 101 grid points in the x-direction. The grid interval is 40 m in the x direction and 20 m in the z direction. Figure 7b shows the reverse time migration (RTM) image. The multilayer NNLSM consists of 3 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 last one contains 15 basis function of dimensions $11 \times 11 \times 15$. Equation 12 is solved for both $m_i$ and $\Gamma_i$ ($i \in \{1, 2, 3\}$) by the two-step iterative procedure denoted as the alternating descent method. The computed effective basis functions for these layers are shown in Figures 7c-7e, where the yellow, red and green boxes indicate the sizes of the effective basis functions, which can be considered as quasi-migration Green’s functions. It indicates that the basis functions of the first layer $\Gamma_1$ contains very simple small-dimensional edges, which are called “atoms” by Elad (2018). The non-zeros of the second basis functions $\Gamma_2$ combine a few atoms from $\Gamma_1$ to create slightly more complex edges, junctions and corners in the effective basis function $\Gamma_2^{(2)}$. Lastly, $\Gamma_3$ combines atoms from $\Gamma_2^{(2)}$ in order to create more complex structure of the migration image. The corresponding stacked coefficient images, also known as feature maps, are shown in Figures 7f-7h, which give the reflectivity distributions.
reconstructed migration images are shown in Figures 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 17x17 grid points (black boxes in Figure 8) and computed for a 50x50 grid (red boxes in Figure 8) of evenly spaced point scatterers with the same migration velocity model as used for the data migration in Figure 7a. The standard LSM images for the first and 50th iterations are shown in Figures 9b and 9c, respectively, next to the NNLSM image in Figure 9d. It is clear that the NNLSM image is more resolved than the LSM image, although there are discontinuities in some of the NNLSM interfaces not seen in the LSM image. Some of the detailed geology is lost in the LSM image as seen in the wiggly interface in the red-circled area of Figure 9. The practical application of the NNLSM image is that it might serve a super-resolved attribute image that can be combined with other attributes to delineate geology. For example, combining the depth-slice of the NNLSM image with a spectral decomposition image (Aarre, 2016) can help delineate the edges of lithology of meandering channels.

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 10a. The time axis is gridded with 213 evenly-spaced points and there are 301 grid points along the x-axis. Twenty-one 13-by-5 (grid point) convolutional basis functions, i.e. filters $\gamma_i$ for
Figure 6: Feature maps for the features a) 1, b) 2, and c) 3 shown in Figure 5. The stacked feature map is shown in d). Here, the white lines show the locations of non-zero points and the yellow lines indicate the locations of the reflectivity distributions.
Figure 7: (a) 2D SEG/EAGE salt model, (b) RTM image, (c)-(e) learned effective filters $\Gamma^{(1)}$, $\Gamma^{(2)}$ and $\Gamma^{(3)}$, (f)-(h) stacked reflectivity coefficients for $m_1$, $m_2$ and $m_3$, (i)-(k) reconstructed migration images $\Gamma^{(1)}m_1$, $\Gamma^{(2)}m_2$ and $\Gamma^{(3)}m_3$. 
Figure 8: (a) Reference model and (b) its migration image for the standard deblurring LSM method.

\(i = 1, 2, \ldots, 21\), are estimated by the NNLSM procedure (see Figure 10b). These filters approximate the dip-filtered migration Green’s functions, and the basis function is marked as the yellow boxes in Figure 10a and 10b. The stacked feature maps (reflectivity distribution) are displayed in Figure 10c. 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 10d with less noise.

**SUMMARY**

Neural network least squares migration finds the optimal reflectivity distribution \(m(x)\) and quasi-migration-Green’s functions that minimizes 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 it can be used for filtering both coherent and incoherent noise in migration images. A practical application of the NNLSM image is as an attribute that provides superresolution of the layer interfaces. This attribute image can be combined with other attributes to delineate both structure and lithology in depth/time slices of migration images. Its disadvantage is that the NNLSM reflectivity image is only an approximation to the actual reflectivity distribution.

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 it reduces to the ReLu operation. This equivalence relates the physics of seismic imaging to architectural features in the neural network.

- The size of the filters in the first layer should be about the same size as the Green’s
Figure 9: (a) RTM image, (b) the first and (c) 50\textsuperscript{th} iteration results by LSM with deblurring, and (d) NNLSM image.
Figure 10: a) Migration image computed from the F3 offshore block data, b) learned filters, c) stacked feature maps and d) migration image after filtering.
function for that model. Experiments with numerical models suggest that this size is approximately 1-2 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 segment spread function (SSF) rather than a migration point spread function. Elad classifies each feature in the first layer as an atom which takes on the role of a 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-reflectivity maps of the subsurface. Adding the sub-reflectivity maps together gives the 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. Elad classifies the concatenation of the filters from the first and second layers as molecules (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 the sparsest feature maps. For migration the final feature maps are very sparse while the concatenated filters are associated with large-scale features of 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.

A significant contribution of our work is that we show that the filters and feature maps of a multilayered CNN are directly related to the migration Green’s functions and reflectivity distributions. For the first time 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 addition, the approximate reflectivity image can be used as a superresolution attribute image for high-resolution delineation of geologic bodies.
REFERENCES

Aarre, V., 2016, Understanding spectral decomposition by victor aarre: https://youtu.be/1nDyMHs8zuw. (Accessed: 2019-04-19).

Ahmad, R., H. Xue, S. Giri, Y. Ding, J. Craft, and O. P. Simonetti, 2015, Variable density incoherent spatiotemporal acquisition (vista) for highly accelerated cardiac mri: Magnetic Resonance in Medicine, 74, 1266–1278.

Bakator, M., and D. Radosav, 2018, Deep learning and medical diagnosis: A review of literature: Multimodal Technologies and Interaction, 2, 47.

Beck, A., and M. Teboulle, 2009, A fast iterative shrinkage-thresholding algorithm for linear inverse problems: SIAM journal on imaging sciences, 2, 183–202.

Chen, Y., G. Dutta, W. Dai, and G. T. Schuster, 2017, Q-least-squares reverse time migration with viscoacoustic deblurring filters: GEOPHYSICS, 82, S425–S438.

Chen, Y., B. Guo, and G. T. Schuster, 2019, Migration of viscoacoustic data using acoustic reverse time migration with hybrid deblurring filters: GEOPHYSICS, 84, S127–S136.

Elad, M., 2018, Sparse modeling in image processing and deep learning: https://youtu.be/9C-QJmaRkcA. (Accessed: 2019-01-28).

Hadsell, R., P. Sermanet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller, and Y. LeCun, 2009, Learning long-range vision for autonomous off-road driving: Journal of Field Robotics, 26, 120–144.

He, K., X. Zhang, S. Ren, and J. Sun, 2016, Deep residual learning for image recognition: Proceedings of the IEEE conference on computer vision and pattern recognition, 770–778.

Heide, F., W. Heidrich, and G. Wetzstein, 2015, Fast and flexible convolutional sparse coding: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 5135–5143.

Hubel, D. H., and T. N. Wiesel, 1962, Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex: The Journal of Physiology, 160, 106–154.

Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012, Imagenet classification with deep convolutional neural networks: Advances in neural information processing systems, 1097–1105.

LeCun, Y., Y. Bengio, and G. Hinton, 2015, Deep learning: Nature, 521, 436.

Liu, Z., K. Lu, and X. Ge, 2018, Convolutional sparse coding for noise attenuation of seismic data: Presented at the 2018 Beijing SEG Workshop: Maximizing asset value through Artificial Intelligence and Machine Learning.

Liu, Z., and G. Schuster, 2018, Neural network least squares migration: Presented at the First EAGE/SBGf Workshop on Least-Squares Migration.

Olshausen, B. A., and D. J. Field, 1996, Emergence of simple-cell receptive field properties by learning a sparse code for natural images: Nature, 381, 607.

Papyan, V., Y. Romano, and M. Elad, 2016, Convolutional neural networks analyzed via convolutional sparse coding: arXiv preprint arXiv:1607.08194.

——–, 2017a, Convolutional neural networks analyzed via convolutional sparse coding: The Journal of Machine Learning Research, 18, 2887–2938.

Papyan, V., Y. Romano, J. Sulam, and M. Elad, 2017b, Convolutional dictionary learning via local processing: Proceedings of the IEEE International Conference on Computer Vision, 5296–5304.

Perez, D. O., D. R. Velis, and M. D. Sacchi, 2013, Estimating sparse-spike attributes from ava data using a fast iterative shrinkage-thresholding algorithm and least squares, in SEG Technical Program Expanded Abstracts 2013: Society of Exploration Geophysicists,
APPENDIX 1: 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 \( z \):

\[
\text{for } x \in D_{\text{model}} : \quad m(x) = \eta \int_{D_{\text{model}}} \int_{y \in D_{\text{data}}} dy \omega^4 G(x|y)^2 G(y|z)^2 m(z), \tag{16}
\]

where \( \eta \) represents terms such as the frequency variable raised to the 4th power. The migration Green’s function \( \Gamma(x|z) \) is given by

\[
\text{for } x, z \in D_{\text{model}} : \quad \Gamma(x|z) = \eta \int_{y \in D_{\text{data}}} dy \omega^4 G(x|y)^2 G(y|z)^2.	ag{17}
\]

Here we implicitly assume a normalized source wavelet in the frequency domain, and \( D_{\text{model}} \) and \( D_{\text{data}} \) represent the sets of coordinates in, respectively, the model and data spaces. The term \( G(x'|x) = e^{i\omega \tau_{xx'}}/||x - x'|| \) is the Green’s function for a source at \( x \) and a receiver at \( x' \) in a smoothly varying medium\(^2\). The traveltime \( \tau_{xx'} \) is for a direct arrival to propagate from \( x \) to \( x' \).

The physical interpretation of the kernel \( \Gamma(x'|x) \) is that it is the migration operator’s\(^3\) response at \( x' \) to a point scatterer at \( x \), otherwise known as the MGF or the migration Green’s function (Schuster and Hu, 2000). It is analogous to the point spread function (PSF) of an optical lens for a point light source at \( x \) in front of the lens and its optical

---

\(^2\)If the source and receiver are coincident at \( x \) then the zero-offset trace is represented by the squared Green’s function \( G(x|x')^2 \).

\(^3\)This assumes that the zero-offset trace is generated with an impulsive point source with a smoothly varying background velocity model, and then migrated by a poststack migration operation. It is always assumed that the direct arrival is muted and there are no multiples.
image at \( x' \) behind the lens on the image plane. In discrete form, the modeling term \( \Gamma_m \) in equation [16] can be expressed as

\[
[\Gamma_m]_i = \sum_j \Gamma(x_i|z_j) m_j. \tag{18}
\]

with the physical interpretation that \( [\Gamma_m]_i \) is the migration Green’s function response at \( x_i \). An alternative interpretation is that \( [\Gamma_m]_i \) is the weighted sum of basis functions \( \Gamma(x_i|z_j) \) where the weights are the reflection coefficients \( m_j \) and the summation is over the \( j \) index. We will now consider this last interpretation and redefine the problem as finding both the weights \( m_i \) and the basis functions \( \Gamma(x_i|z_j) \). This will be shown to be equivalent to the problem of a fully connected (FCN) neural network.

**APPENDIX 2: SOFT THRESHOLDING FUNCTION**

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

\[
\epsilon = \frac{1}{2} \| z - x \|_2^2 + \lambda \| x \|_1, \tag{19}
\]

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 [19]. Here, the noisy \( M \times N \) image has been flattened into the tall \( MN \times 1 \) vector \( z \).

The stationary condition for equation [19] is

\[
\frac{\partial \epsilon}{\partial x_i} = (x_i - z_i) + \lambda \frac{\partial \| x \|_1}{\partial x_i}, \tag{20}
\]

where

\[
\frac{\partial \| x \|_1}{\partial x_i} = 1 \text{ for } x_i \geq 0; \quad \frac{\partial \| x \|_1}{\partial x_i} = -1 \text{ for } x_i < 0. \tag{21}
\]

Equations [20, 21] can be combined to give the optimal \( x^* \) expressed as the two-sided ReLu function

\[
x_i = \text{soft}(z_i, \lambda) = \begin{cases} 
    z_i - \lambda & \text{if } z_i \geq \lambda \\
    0 & \text{if } |z_i| < \lambda \\
    z_i + \lambda & \text{if } z_i < -\lambda 
\end{cases}. \tag{22}
\]

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

\[
\mathbf{x}^* = \arg \min_{\mathbf{x}} \left[ \frac{1}{2} \| z - \mathbf{Wx} \|_2^2 + \lambda \| \mathbf{x} \|_1 \right], \tag{23}
\]

is

\[
x_i^{(k+1)} = \text{soft}\left( x_i^{(k)} - \frac{1}{\alpha} \mathbf{W}^T(\mathbf{Wx}^{(k)} - z), \frac{\lambda}{\alpha} \right)_i. \tag{24}
\]

There are several more recently developed algorithms that have faster convergence properties than ISTA. For example, FISTA (Fast-ISTA) has quadratic convergence [Beck and Teboulle, 2009].
APPENDIX 3: NEURAL NETWORK LEAST SQUARES MIGRATION

The neural network least squares migration (NNLSM) algorithm in the image domain is defined as solving for both the basis functions \( \tilde{\Gamma}(x_i|x_j) \) and \( \tilde{m}_j \) that minimize the objective function defined in equation 5. In contrast, SLSM only finds the least squares migration image in the image domain and uses the pre-computed migration Green’s functions that solve the wave equation.

The NNLSM solution is defined as

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

where now both \( \tilde{\Gamma}^* \) and \( \tilde{m}^* \) are to be found. The functions with tilde’s 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 25 is given by

\[
\epsilon = \frac{1}{2} \sum_i \left[ \sum_j \tilde{\Gamma}(x_i|z_j) \tilde{m}_j - m^{mig}_i \right]^2 + \lambda S(\tilde{m}).
\]

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

\[
\frac{\partial \epsilon}{\partial \tilde{\Gamma}(x_i'|z_j')} = \sum_j (\tilde{\Gamma}(x_i'|z_j) \tilde{m}_j - \tilde{m}^{mig}_i) \tilde{m}_j'.
\]

The iterative solution of equation 25 is given in two steps (Olshausen and Field, 1996).

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

\[
\tilde{m}_i^{(k+1)} = \tilde{m}_i^{(k)} - \alpha \frac{1}{2} \sum_j \tilde{\Gamma}(x_i|z_j) \tilde{m}_j - m^{mig}_i \frac{\partial \epsilon}{\partial \tilde{\Gamma}(x_i'|z_{j'})},
\]

However, one migration image \( m^{mig} \) is insufficient to find so many unknowns. In this case the original migration image is broken up into many small pieces so that there are many migration images to form examples from a large training set. For prestack migration, there will be many examples of prestack migration images, one for each shot, and the compressive sensing technique denoted as VISTA (Ahmad et al., 2015) is used for the calculations.

2. Update the basis functions \( \tilde{\Gamma}(x_i|z_j) \) by inserting equation 27 into the gradient descent formula to get

\[
\tilde{\Gamma}(x_i'|z_j')^{(k+1)} = \tilde{\Gamma}(x_i'|z_j')^{(k+1)} - \alpha \frac{\partial \epsilon}{\partial \tilde{\Gamma}(x_i'|z_{j'})},
\]

\[
= \tilde{\Gamma}(x_i'|z_j')^{(k+1)} - \alpha \left\{ \sum_j \tilde{\Gamma}(x_i'|z_j) \tilde{m}_j \right\} - m^{mig}_i \tilde{m}_j'.
\]

It is tempting to think of \( \tilde{\Gamma}(x|x') \) as the migration Green’s function and \( \tilde{m}_i \) as the component of reflectivity. However, there is yet no justification to submit to this temptation and so we must consider, unlike in the SLSM algorithm, that \( \tilde{\Gamma}(x|x') \) is a sparse basis function and \( \tilde{m}_i \) is its coefficient. To get the true reflectivity then we should equate equation 18 to \( \sum_j \tilde{\Gamma}(x_i, x_j) \tilde{m}_j \) and solve for \( m_j \).
APPENDIX 4: ALIGNMENT OF THE FILTERS

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

$$C_{i+M,j+N} = \sum_{m=1}^{M} \sum_{j=1}^{N} a_{m,n} \cdot b_{m+i,n+j},$$

where $-M < i < M$ and $-N < j < N$. $a_{i,j}$, $b_{i,j}$ and $C_{i,j}$ indicate the element at position $(i,j)$ in matrices $A$, $B$ and $C$, respectively. The location of the maximum absolute value of the elements in matrix $C$ indicates how much should we shift filter $B$ to filter $A$ in each direction. Figure 11 shows the calculation of the cross-correlation matrix $C$ for two filters $A$ and $B$. $c_{1,0}$ (or $C_{4,3}$) is the maximum absolute value of the elements in matrix $C$, which indicates filter $B$ should be shifted 1 position along the first direction. Here, we need to pad zeros along all the dimensions of filter $B$ before shifting it, which is displayed in Figure 12.

Figure 13a shows the learned filters with a size of $17 \times 9$ from the migration image of the SEG/EAGE salt model. Filter No. 7 (yellow box) is chosen as the target filter. The aligned filters are shown in Figure 13b, where the filters are padded with 8 and 4 zeros along the $z$ and $x$ directions, respectively. The stacked feature maps from the original and aligned filters are displayed in Figures 13c and 13c, respectively. It is evident that the reflector interfaces from the aligned filters are more continuous especially in the red box compared with those of the original filters.
Calculation of Cross-correlation Matrix $C$

\[
C = \begin{pmatrix}
  c_{-2,-2} & c_{-2,-1} & c_{-2,0} & c_{-2,1} & c_{-2,2} \\
  c_{-1,-2} & c_{-1,-1} & c_{-1,0} & c_{-1,1} & c_{-1,2} \\
  c_{0,-2} & c_{0,-1} & c_{0,0} & c_{0,1} & c_{0,2} \\
  c_{1,-2} & c_{1,-1} & c_{1,0} & c_{1,1} & c_{1,2} \\
  c_{2,-2} & c_{2,-1} & c_{2,0} & c_{2,1} & c_{2,2}
\end{pmatrix}
\]

\[
\begin{pmatrix}
  0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 2 \\
  0 & 0 & 0 & 7 & 0 \\
  0 & 0 & 8 & 0 & 0 \\
  0 & 4 & 0 & 0 & 0
\end{pmatrix}
\]

Figure 11: Calculation of the cross-correlation matrix $C$. 
Circular Shifting of Filter B

Filter B

\[
\begin{pmatrix}
  b_{1,1} & b_{1,2} & b_{1,3} \\
  b_{2,1} & b_{2,2} & b_{2,3} \\
  b_{3,1} & b_{3,2} & b_{3,3}
\end{pmatrix}
\]

Pad zeros around B

Circularly shifts the elements in padded matrix B by 1 position along 1\(^{st}\) dimension

\[
\begin{pmatrix}
  0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & b_{1,1} & b_{1,2} & b_{1,3} & 0 \\
  0 & 0 & b_{2,1} & b_{2,2} & b_{2,3} & 0 & 0 \\
  0 & b_{3,1} & b_{3,2} & b_{3,3} & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\rightarrow
\begin{pmatrix}
  0 & 0 & 0 & 0 & 0 & 0 & 0 \\
  0 & 0 & 0 & b_{1,1} & b_{1,2} & b_{1,3} & 0 \\
  0 & 0 & b_{2,1} & b_{2,2} & b_{2,3} & 0 & 0 \\
  0 & b_{3,1} & b_{3,2} & b_{3,3} & 0 & 0 & 0 \\
  0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

Figure 12: Diagram that illustrates the circular shifting of padded filter B.
Figure 13: (a) Learned filters from the migration image of SEG/EAGE salt model; (b) the aligned filters; stacked feature maps of the (c) original and (d) aligned filters, where the yellow and green boxes show the sizes of the filters for the original and aligned filters, respectively.