A DEEP LEARNING INVERSE HESSIAN FOR LEAST-SQUARES MIGRATION

A. Kumar[1], G. Hampson[1], T. Rayment[1], T. Burgess[1]

[1] DUG Technology

Summary

A new deep-learning (DL) approach for approximating the inverse of the Hessian as part of least-squares migration is proposed. This approach aims to compute complex non-linear non-stationary matching filters in the form of a convolutional neural network. This choice is motivated by the similarities between the conventional least-squares filtering method and DL methods. The aim is to train the network using a random selection of overlapping patches extracted from two migrated images that theoretically embed the effects of the Hessian.

A deep learning inverse Hessian for least-squares migration

Introduction

Least-squares migration is a well-established method for improving the quality of seismic images. The least-squares formulation involves the inversion of the Hessian operator which, even in an iterative approach, is expensive. Therefore, the inverse Hessian is often approximated using less expensive methods. For example, Rickett et al., (2003) proposed a simple scaling operator to approximate the inverse Hessian with a diagonal matrix. Guitton (2004) extended this approach, approximating the inverse Hessian with a bank of non-stationary matching filters, while Kaur et al. (2020) adopted a deep learning technique using generative adversarial networks.

In this paper, we propose a new deep-learning (DL) approach for approximating the action of the inverse Hessian. This approach aims to compute complex non-linear non-stationary matching filters in the form of a convolutional neural network. This choice is motivated by the similarities between the conventional least-squares match filtering method and DL methods. The aim is to train the network using a random selection of overlapping patches extracted from two migrated images that theoretically embed the effects of the Hessian. We recently developed a *supervised convolutional autoencoder* (SCAE) network for the adaptive subtraction of predicted multiples from the input recorded data (Kumar et al., 2021). Here, we exploit a similar concept to estimate the non-stationary Hessian as part of image-domain least-squares migration. We show the uplift in results using a 3D network over a 2D network and a 3D conventional least-squares filtering method.

Method

Given seismic data $\mathbf{d}$ and a linear modeling operator $\mathbf{L}$, the migration operator is described as the adjoint of the modeling operator such that,

$$\mathbf{m}_1 = \mathbf{L}^H \mathbf{d}\,, \tag{1}$$

where, ^{H} denotes Hermitian transpose. This is a conventional migration image which is relatively inexpensive to compute, but limited in resolution, amplitude fidelity and frequency content. These limitations can be substantially overcome using least-squares inversion. The least-squares estimate $\hat{\mathbf{m}}$ of the model in (1) is given by,

$$\hat{\mathbf{m}} = \left(\mathbf{L}^H \mathbf{L}\right)^{-1} \mathbf{L}^H \mathbf{d}\,, \tag{2}$$

where $\mathbf{L}^H \mathbf{L}$ is the Hessian operator. The inverse of the Hessian is regarded as a deconvolution operator (Hu et al., 2001) that corrects the amplitudes and frequency content of the final image. Guitton (2004) proposed an idea to approximate the inverse of the Hessian $\left(\mathbf{L}^H \mathbf{L}\right)^{-1}$ through a bank of non-stationary matching filters. He rewrote equation (2) by replacing $\mathbf{L}^H \mathbf{d}$ term with $\mathbf{m}_1$ to obtain,

$$\hat{\mathbf{m}} = \left(\mathbf{L}^H \mathbf{L}\right)^{-1} \mathbf{m}_1 \tag{3}$$

where, $\hat{\mathbf{m}}$ and $\left(\mathbf{L}^H \mathbf{L}\right)^{-1}$ are unknown. However, to find an approximation of the inverse Hessian with non-stationary matching filters, two known images are needed that are related by the Hessian. This can be achieved by remodeling the data from $\mathbf{m}_1$ with $\mathbf{L}$ and then remigrating with $\mathbf{L}^H$ to get second image, $\mathbf{m}_2 = \mathbf{L}^H \mathbf{L} \mathbf{m}_1$, from which we see the inverse Hessian satisfies,

$$\mathbf{m}_1 = \left(\mathbf{L}^H \mathbf{L}\right)^{-1} \mathbf{m}_2\,. \tag{4}$$

This is very similar to equation (3) except $\mathbf{m}_1$ and $\mathbf{m}_2$ are known images. Now the inverse of the Hessian can be approximated by matching filters that will map $\mathbf{m}_2$ to $\mathbf{m}_1$ in equation (4). These filters can then be applied to $\mathbf{m}_1$ to approximate $\hat{\mathbf{m}}$ in equation (2). This approach has now become widespread in order to improve the migrated image at much lower cost than an iterative least-squares migration. Several authors have developed methods to achieve improved approximations of the inverse

Hessian either to enhance the image quality or to use as a preconditioning filter at each least-squares iteration (Aoki, 2009).

Recently, we have worked on a DL method for the adaptive subtraction of predicted multiples from input recorded data (Kumar et al., 2021). We developed a SCAE network architecture that involved training using a random selection of patches from shot/channel gathers followed by the application of the trained network to the rest of the dataset. The use of deep-learning networks to perform adaptive subtraction provides significant uplift compared to conventional approaches thanks to complex non-linear matching filters. The same idea can be applied to match $\mathbf{m}_2$ to $\mathbf{m}_1$ in order to find a better approximation of the inverse Hessian than conventional least-squares match filtering. We may write the new equation for the SCAE filters by replacing the conventional match filtering with the SCAE acting on the second image, $\mathbf{m}_2$, in equation (4) as,

$$\mathbf{m}_1 = \mathrm{SCAE}(\mathbf{m}_2)\,. \tag{5}$$

Once the network has been trained using (5), it can be applied to the first image, $\mathbf{m}_1$, to get the final estimate of the least squares migrated image $\hat{\mathbf{m}}$ in the manner of equation (3) using,

$$\hat{\mathbf{m}} = \mathrm{SCAE}(\mathbf{m}_1)\,. \tag{6}$$

Network architecture

Our design proceeded by experimentation and recognising that the non-linear nature of the network architecture meant that the training would, at best, end up in a useful local, rather than global, *loss function* minimum (Kumar et al., 2021). Our final SCAE network is shown in Figure 1. It has a total of 9 layers (including the *bottleneck*) with the number of *feature maps* shown in each layer. The input and output layers reflect the *patch* sizes (64×64 samples) and the *latent representation* in the *bottleneck* has 14×14 samples per *feature map*. We found that the best results were obtained using a *convolutional filter* size of 4×4. The output *channels* of each layer were passed through a non-linear *activation function* called an *exponential linear unit* (*ELU*). We used the same network architecture with one extra dimension in all layers for the 3D SCAE.

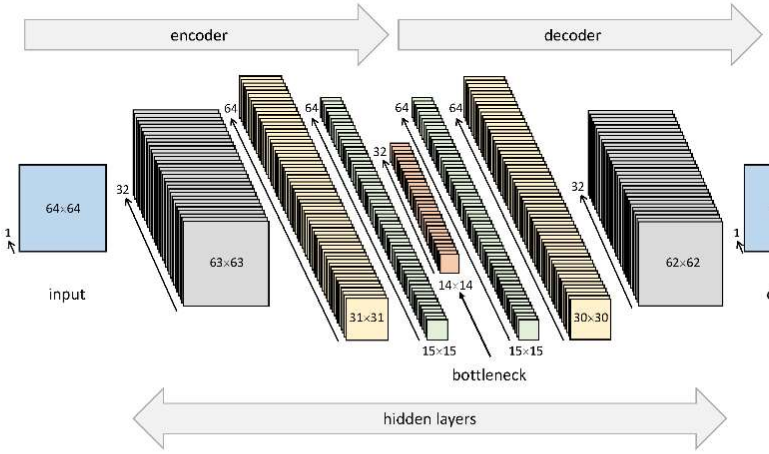

***Figure 1** The network architecture of our supervised convolutional autoencoder* (SCAE).

Results

We demonstrate the application of the proposed approach to a field data example from offshore Gabon. The data were migrated using reverse time migration ($f_{\max} = 33\mathrm{Hz}$) to compute the first image $\mathbf{m}_1$ (Figure 2a), then modelled and remigrated to get the second migrated image, $\mathbf{m}_2$ (Figure 2b). Adjoint operators were used to perform the modeling and migration steps. However, although the operators are kinematically correct, the amplitude fidelity and frequency content are affected by the acquisition geometry and migration, as anticipated. For example, the second migrated image $\mathbf{m}_2$ has lower amplitudes in the shallow section.

Using these two images $\mathbf{m}_1$ and $\mathbf{m}_2$, we derived three different approximations of the inverse Hessian: 1) 3D windowed least-squares match filters ("3D match"), 2) 2D SCAE filters, and 3) 3D SCAE filters. We first applied the resulting filters to $\mathbf{m}_2$ (shown in Figure 3). To demonstrate how well each method performed we show the residual error $\mathbf{r} = \mathbf{f} * \mathbf{m}_2 - \mathbf{m}_1$ in Figure 4. A quantitative measure of the residual, $\| \mathbf{r} \| / \| \mathbf{m}_1 \|$ is annotated on each residual panel. Although, all three methods have performed well to approximate the inverse Hessian, both of the DL approaches were slightly better than the 3D match, with the 3D SCAE being marginally better than the 2D SCAE. In this instance, 50 percent of the total data were used for the training, although even better results are possible with greater exposure to more data.

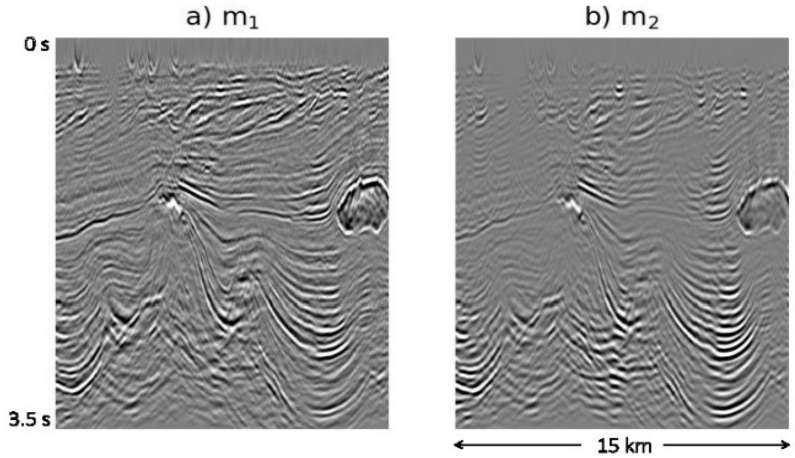

***Figure 2** a) First migrated image obtained using RTM; b) second migrated image obtained using remodeling and remigration exercise.*

Finally, we applied the same sets of filters to the first image, $\mathbf{m}_1$, to produce the estimated final image $\hat{\mathbf{m}}$ for each of the 3 approaches. The results are shown in Figure 5. All versions show an improved resolution and amplitude consistency compared to the initial migration, $\mathbf{m}_1$ (Figure 2a). The amplitudes appear slightly more consistent in the DL results than in the 3D match result, especially in the area indicated by the yellow arrows and the yellow circle.

Conclusions

We have presented a new method of approximating the inverse Hessian operator as part of least-squares migration. The method is based on a deep learning approach that is trained to find a transfer function between a migrated image that has had the Hessian applied and the initial migrated image. It is trained using a subset of the image volumes and then applied to the whole migrated image. The field data example demonstrates the effectiveness of the proposed approach. Results show that the 3D network performs slightly better than the 2D network and they both outperform a 3D least-squares match filtering method.

Acknowledgements

We would like to thank DUG Technology (DUG) for allowing us permission to present this work.

References

Aoki, N. and Schuster, G.T. [2009]. Fast least-squares migration with a deblurring filter. *Geophysics*, **74**(6), WCA83–WCA93.

Guitton, A. [2004]. Amplitude and kinematic corrections of migrated images for nonunitary imaging operators. *Geophysics*, **69**(4), 1017–1024.

Hu, J., Schuster, G. T. and Valasek, P. [2001]. Poststack migration deconvolution: *Geophysics*, **66**, 939–952.

Kaur, H., Pham, N. and Fomel, S. [2020]. Improving resolution of migrated images by approximating the inverse Hessian using deep learning. *Geophysics*, **85**(4), 1–62.

Kumar, A., Hampson, G. and Rayment, T. [2021]. Adaptive subtraction using a convolutional neural network. *First Break*, **39**(9), 35-45.
Rickett, J.E. [2003]. Illumination-based normalization for wave-equation depth migration. *Geophysics*, **68**(4), 1371–1379.

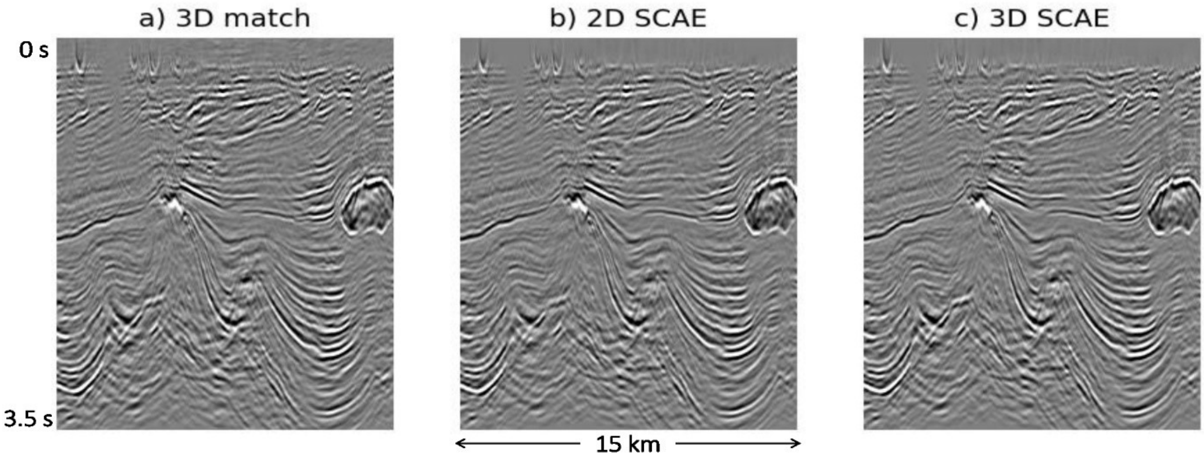

Figure 3 *Results of applying matching filters to the second migrated image. a) using 3D match; b) using a 2D SCAE network; c) using a 3D SCAE network.*

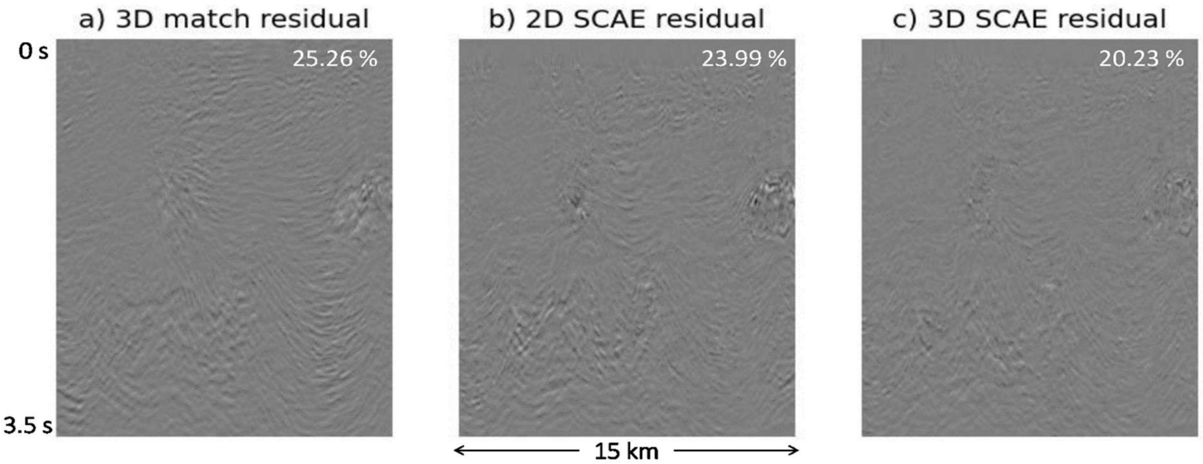

Figure 4 *Residual error between the first migrated image (Figure 2a) and the second migrated image after the application of matching filters (Figure 3).*

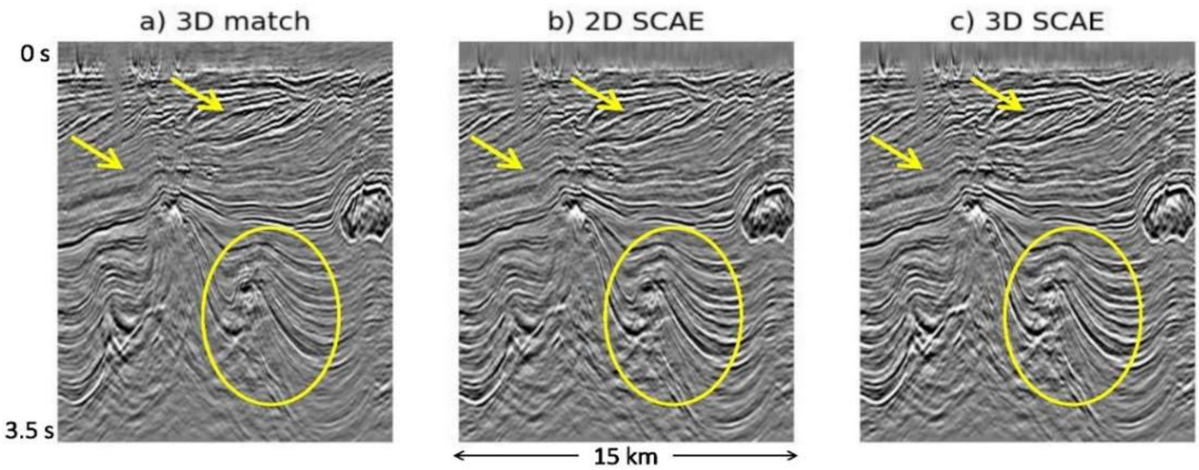

Figure 5 *Final migrated images after the application of matching filters and SCAE filters on the first migrated image (Figure 2a).*