A Generative Adversarial Network For Seismic Imaging Applications

Francesco Picetti, Vincenzo Lipari, Paolo Bestagini, Stefano Tubaro, Politecnico di Milano, Italy*

SUMMARY

The new challenges of geophysical imaging applications ask for new methodologies going beyond the standard and well established techniques. In this work we propose a novel tool for seismic imaging applications based on recent advances in deep neural networks. Specifically, we use a generative adversarial network (GAN) to process seismic migrated images in order to potentially obtain different kinds of outputs depending on the application target at training stage.
We demonstrate the promising features of this tool through a couple of synthetic examples. In the first example, the GAN is trained to turn a low-quality migrated image into a high-quality one, as if the acquisition geometry were much more dense than in the input. In the second example, the GAN is trained to turn a migrated image into the respective deconvolved reflectivity image.

INTRODUCTION

Applications in hydrocarbons exploration, reservoir characterization and civil engineering, require a subsurface mapping at increasingly higher resolution and higher fidelity. Moreover, the amount of collected data that needs to be analyzed is constantly increasing and, as for hydrocarbon exploration, the areas of interest are more and more complex to analyze. This determined a high demand for advanced seismic imaging methodologies that prove computationally efficient.

To face some of these issues, in the last few years, there has been an increasing renewed interest in machine learning techniques. In particular, supervised classification methods have been increasingly explored by the geophysical community, especially as a helping tool for the interpretation step (Hall, 2016; Bestagini et al., 2017).

However, most of the components of seismic imaging workflows are large scale ill-posed inverse problems, rather than classification ones. From the analytical point of view, the study of regularization, iterative methods and appropriate cost functions has historically played a key role both in seismic imaging and in other areas (e.g., biomedical imaging). However, for some inverse imaging applications, deep learning techniques have also proved particularly interesting recently (Lucas et al., 2018).

As a matter of fact, the recent advancements brought by convolutional neural networks (CNNs) have greatly impacted the whole signal and image processing community. In particular, among the different architectures, generative adversarial networks (GANs) emerged as a promising approach for problems that need some form of regularization that is not easy to express through simple modeling (Goodfellow et al., 2014).

While leading to state-of-the-art results in computer vision, image processing and various related field, deep learning has barely started to be studied for inverse imaging problems (Al-Regib et al., 2018). McCann et al. (2017) provide a review of recent applications of convolutional neural networks for biomedical imaging problems. Recently, Araya-Polo et al. (2018) proposed a deep learning strategy for seismic velocity model building.

In this work we introduce a possible way of using deep learning for seismic imaging applications. In particular, we propose to use a GAN as a tool for processing seismic images obtained via Reverse Time Migration (RTM). We formulate the problem as the estimation of a post processing operator that can be learned through a training phase to tackle different problems. Specifically, the GAN is fed with pairs of images composed of input images and desired output images depending on the target application (e.g., deconvolved images, Least squares-RTM images, deghosted images, etc.).

The used CNN architecture builds upon the recently proposed pix2pix GAN (Isola et al., 2017). Its potential is shown by tailoring the proposed architecture to two different applications on synthetic migrated data.

The first application is data interpolation in the image space: we aim at recovering an image obtained with a dense source-receiver acquisition geometry from an image migrated with a very coarse acquisition geometry. The example we show is obtained on several 2D migrated sections of the SEG/EAGE Overthrust velocity model.

The second application is deconvolution. In particular we train our CNN on a portion of the well known SMAART JV Sigsbee velocity model, to transform the migrated image on the corresponding reflectivity section (obtained from the stratigraphic velocity model). Then we predict the reflectivity from the remaining part of the migrated image.

Preliminary results confirm the positive impact that deep learning can have in seismic image processing in the future.

GAN FOR SEISMIC IMAGE PROCESSING

In this section we introduce the way we cast seismic image processing problem in the GAN framework, then we provide all details about the proposed solution.

Problem formulation

The goal of the proposed method is to build a machine that takes a migrated image $\mathbf{I}$ as input, and produces an image $\hat{\mathbf{I}}$ as output, as depicted in Figure 1. During training stage, the target application is chosen in order to obtain a desired output $\hat{\mathbf{I}}$ depending on the problem to solve.

In other words during training, the machine is fed with a set of K pairs $\{\mathbf{I}^{(k)}, \mathbf{I_{ref}}^{(k)}\}$,$k \in [1,K]$, where $\mathbf{I}^{(k)}$ are migrated images, and $\mathbf{I_{ref}}^{(k)}$ are the corresponding desired outputs. In this

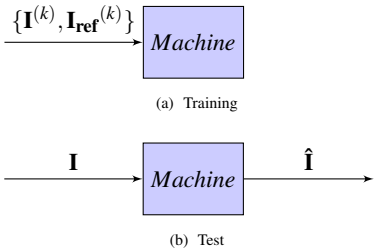


Figure 1: Training and test pipeline.

phase, the machine learns how to transform a migrated image $\mathbf{I}$ into the corresponding desired image $\hat{\mathbf{I}}$, by minimizing an appropriate cost function. Depending on the application we can decide to train the machine over very different desired images, for instance: deconvolved images, least-square migrated images, deghosted images, etc.

Proposed solution

To solve the aforementioned problem, we propose to use a generative adversarial network (GAN). Specifically, a GAN is a composition of two neural networks trained in a joint fashion: a *generator* G that takes care of the input-output image mapping (i.e., $\hat{\mathbf{I}} = G(\mathbf{I})$); and a *discriminator* D that aims at distinguishing between generated images $\hat{\mathbf{I}}$ (i.e., $D(\hat{\mathbf{I}}) = 0$) and the reference ones $\mathbf{I_{ref}}$ (i.e., $D(\mathbf{I_{ref}}) = 1$).

As architectures for G and D we followed the guidelines provided in (Isola et al., 2017). Specifically, the generator is a U-net (Ronneberger et al., 2015), which is a fully convolutional neural network composed by a series of more than 10 convolutional layers with skipped connections, for a total amount of more than 42 million parameters. As the output of the network has the same size of the input, this U-net already proved to be particularly well suited to transform images from one domain to another one (Ronneberger et al., 2015). The discriminator D is a simpler and shallower fully connected network composed by a series of convolutional, pooling and rectified linear unit layers. Its goal is to predict a label (i.e., true or false) when fed with an original or synthetically generated image.

Both G and D can be seen as series of parametric operators. The parameters are chosen through training. This means that a cost function (typically referred to as loss) is chosen, and an iterative procedure is applied to find network parameters that minimize a cost function over the used training image pairs $\{\mathbf{I}^{(k)}, \mathbf{I_{ref}}^{(k)}\}$.

The rationale behind GAN training is that the discriminator is trained to understand whether an image under analysis is a real image, or an image obtained through the generator. At the same time, the generator is trained to obtained the desired output from a given input, and fool the discriminator. In other words, the discriminator can be seen as a regularizer of the generator. It enforces the generator to output images visually similar to real ones.

From a more formal point of view, the used cost function depends on several terms. One term is the generator loss defined as

$$\mathscr{L}_{\mathrm{G}}(\mathbf{I}, \mathbf{I_{ref}}) = \|\mathbf{I_{ref}} - G(\mathbf{I})\|_1, \tag{1}$$

which represent the ℓ_1-norm of the error introduced by the generator. This term controls that the generated image is coherent with the desired one. As additional term, we define the GAN loss as

$$\mathscr{L}_{\mathrm{GAN}}(\mathbf{I}) = \log D(\mathbf{I}) + \log(1 - D(G(\mathbf{I})), \tag{2}$$

which measures how likely the generator is able to fool the discriminator in terms of binary cross-entropy. Finally, we propose an additional normalization term defined as

$$\mathscr{L}_{\ell_1}(\mathbf{I}) = \|G(\mathbf{I})\|_1, \tag{3}$$

which enforces the generated image to have small ℓ_1-norm, as typically required in some seismic imaging applications. The overall cost function to minimize is then

$$\mathscr{L}(\mathbf{I}, \mathbf{I_{ref}}) = \mathscr{L}_{\mathrm{G}}(\mathbf{I}, \mathbf{I_{ref}}) + \lambda_1 \mathscr{L}_{\mathrm{GAN}}(\mathbf{I}) + \lambda_2 \mathscr{L}_{\ell_1}(\mathbf{I}), \tag{4}$$

where λ_1 and λ_2 are used as weights for the different loss terms. In our work, minimization is achieved relying on the well-known Adam technique customary used in many deep learning applications (Kingma and Ba, 2014).

In order to adapt the proposed method to images of any size, the network is built to work on image patches. Specifically, every time we analyze an image (for either training or testing), we split it into smaller patches of 128×128 samples each, which are processed separately.

When the whole GAN has been trained, we can process new images $\mathbf{I}$. Specifically, we feed $\mathbf{I}$ (or patches of it) to the generator G, and obtain the estimated $\hat{\mathbf{I}}$ (or its patches that can be simply spliced together).

APPLICATIONS

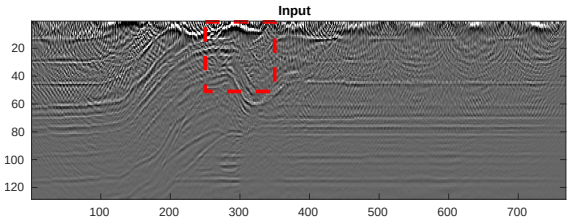
In this section we discuss the investigated applications, providing for each one of them all details about the used dataset and achieved results.

High quality images from coarse data

The scenario proposed in this example is, for instance, that of fast track projects, when we desire to obtain high quality migrated images but we have no time/resources to perform RTM over the entire data. In this case, the input image $\mathbf{I}$ is a depth migrated image obtained from a very coarse acquisition geometry. The output $\hat{\mathbf{I}}$ that we want is the corresponding depth migrated image as obtained with a dense acquisition geometry.

Here we show the results obtained on a modified version of the SEG/EAGE Overthrust model. In order to build the training set $\{\mathbf{I}^{(k)}, \mathbf{I_{ref}}^{(k)}\}$ we extracted 1 392 patches (of 128×128 samples) from 58 pairs of 2D migrated in-lines and x-lines. Each 2D section was made of 768×128 samples, with a sampling step of 30m.

The images $\mathbf{I}^{(k)}$ have been generated by migrating a coarse acquisition geometry, designed with 10 equispaced sources and 80 equispaced receivers covering all the acquisition surface.



(a)

$\mathrm{img}_{output\_over.pdf}\ \mathrm{img}_{output\_over.png}\ \mathrm{img}_{output\_over.eps}$

(b)

Figure 2: Input image $\mathbf{I}$ (a) and desired output $\mathbf{I_{ref}}$ (b) for our first application.

The images $\mathbf{I_{ref}}^{(k)}$ have been generated by migrating a dense acquisition geometry, designed with 200 equispaced sources and 800 equispaced receivers covering all the acquisition surface.

After the training, the GAN has been tested on different inlines and x-lines never used for training. This is done to assess the generalization capability of the trained GAN. In Figure **??** we show an input migrated section (a) and the corresponding output (b). The red square highlights the portion of the image zoomed in Figure 3.

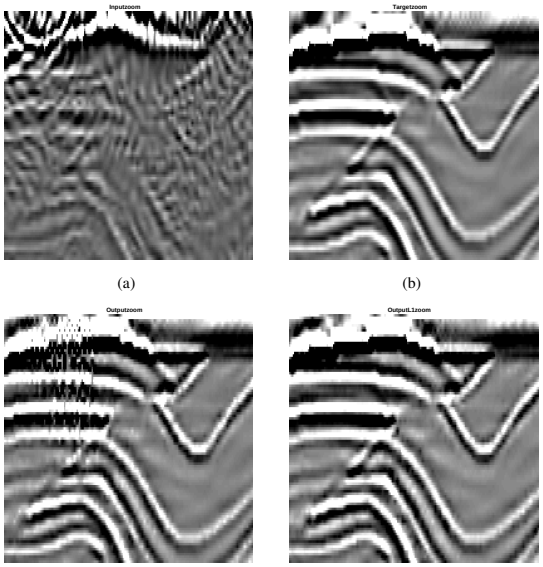
In particular, Figure 3(a) is a close-up view of the input image $\mathbf{I}$ and Figure 3(b) is the actual corresponding desired target image $\mathbf{I_{ref}}$ (i.e., the zoomed version of the image obtained by migrating the dense acquisition geometry). The output of the trained network upon convergence is shown in Figures 3(c) and 3(d) when $\lambda_2 = 0$ and $\lambda_2 = 10$, respectively (i.e., we consider or not $\mathscr{L}_{\ell_1}$). It is possible to see that both solutions are satisfactory: the low-resolution artifacts have been almost completely eliminated and the visual quality of the image is remarkably like that of the desired target. However, by adding a constraint on the ℓ_1 norm (Figure 3(d)) of the output we have been able to further improve the result and to accelerate the convergence of the training phase.

In order to have a numerical evaluation of the achieved results we choose as a metric the Signal-to-Noise Ratio (SNR) in dB, defined as

$$\mathrm{SNR} = 10\log_{10}\frac{Var(\mathbf{I_{ref}})}{VAr(\hat{\mathbf{I}}-\mathbf{I_{ref}})}. \tag{5}$$

Testing 79 images, we obtained an average SNR of 17.15 dB and 16.6 dB for the outputs obtained with and without ℓ_1 constraint, respectively. This confirms the improvement given by the proposed $\mathscr{L}_{\ell_1}$ term added to the overall loss function.

Once the training of the network is completed, the computational time needed to obtain an image of comparable quality with respect to that obtained with a dense acquisition geometry dramatically reduces. For instance, in order to build the example shown in this section, the time needed to generate a single migrated section with the dense migration geometry was around 40 minutes. Instead, the time needed to generate an output image of the network was about 2 minutes, almost entirely dedicated to migration with the coarse geometry, only



(a) (b) (c) (d)

Figure 3: Input (a), desired output (b), results achieved without (c) and with (d) $\mathscr{L}_{\ell_1}$ loss term.

few seconds were needed to generate the output $\hat{\mathbf{I}}$ from the input $\mathbf{I}$ using a single Nvidia Titan X GPU.

Reflectivity from migrated images

As we believe that the analyzed CNN represents a promising approach for deconvolution-like problems (e.g., deghosting, LS-RTM etc.), we propose a synthetic example in order to illustrate its potential. Here, the input $\mathbf{I}$ is a standard depth migrated image and the desired output $\hat{\mathbf{I}}$ is an image of subsurface's reflectivity.

In a synthetic experiment we are able to build a training set $\{\mathbf{I}^{(k)}, \mathbf{I_{ref}}^{(k)}\}$ where the elements $\mathbf{I}^{(k)}$ are patches of depth migrated images and the elements $\mathbf{I_{ref}}^{(k)}$ are the corresponding patches of reflectivity images $r(x,z)$, computed from the stratigraphic velocity models $v(x,z)$ as

$$r(x,z) = \frac{v(x,z+\Delta z) - v(x,z)}{v(x,z)}. \tag{6}$$

We trained the network on 194 pairs of patches from the Sigsbee model and then we tested it on a different validation set of 193 pairs.

As evaluation metric we compared the output and the target through the structural similarity index (SSIM) defined as:

$$\mathrm{SSIM} = \frac{(2E[\mathbf{I_{ref}}]E[\hat{\mathbf{I}}]+c_1)(2Cov(\mathbf{I_{ref}},\hat{\mathbf{I}})+c_2)}{(E[\mathbf{I_{ref}}]+E[\hat{\mathbf{I}}]+c_1)(Var(\mathbf{I_{ref}})+Var(\hat{\mathbf{I}})+c_2)}, \tag{7}$$

where c_1 and c_2 are appropriate damping factors, E computes the mean, and Cov the covariance.

We show a couple of results computed on the validation patches with different structural features: an area with the presence of a salt body, and the water bottom interface (Figure **??**) and an area of sediments only (Figure 5).

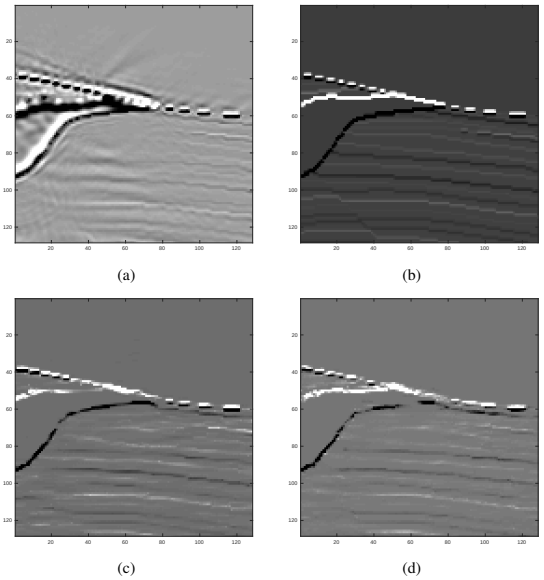


Figure 4: Input (a), desired output (b), results achieved without (c) and with (d) $\mathscr{L}_{\ell_1}$ loss term.

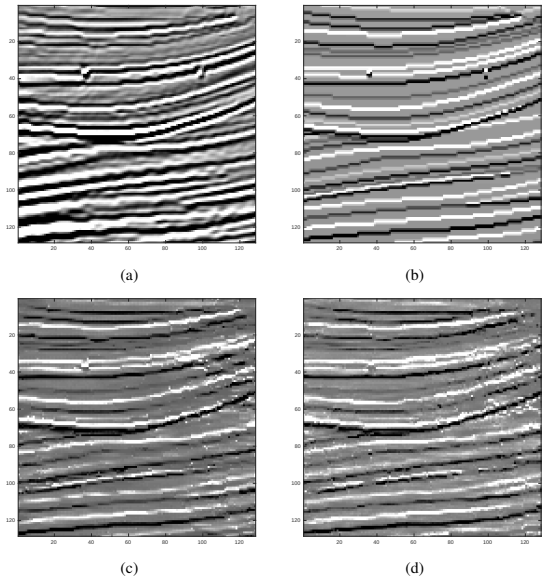


Figure 5: Input (a), desired output (b), and results achieved without (c) and with (d) $\mathscr{L}_{\ell_1}$ loss term.

For each example we show: (a) the input migrated image $\mathbf{I}$; (b) the corresponding reflectivity, which is the reference desired output $\mathbf{I}$; (c) the output $\hat{\mathbf{I}}$ of the GAN trained on the training dataset for 1000 iterations; (d) the output $\hat{\mathbf{I}}$ of the GAN with the additional ℓ_1 constraint on the loss function and with a training of 200 iterations. In the first example, the average SSIM is around 0.66, as the salt makes the problem more challenging. In the second example, the average SSIM is 0.90, much closer to the optimal value of 1. Despite the use of ℓ_1 constraint in loss function does not improve the quality of the reconstructed image, it is worth noting that it makes training convergence much faster.

Figure 6 shows a 1D vertical profile extracted at the central horizontal location from the example of Figure 5. The blue dotted lines represent the ideal reflectivity profile, the black lines are extracted from the migrated section (left image) and from the output of the GAN with regularization (right image). The effect of deconvolution is quite evident, and also the amplitudes are well recovered.

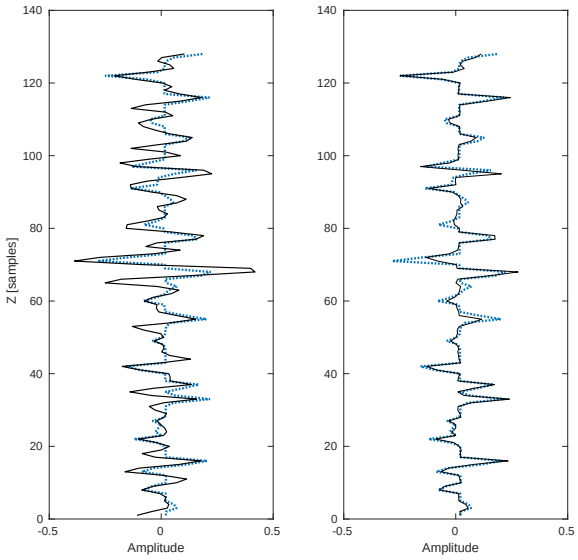


Figure 6: 1D vertical profiles. The reference profile (blue line) is compared with the migrated (a) and deconvolved (b) profiles represented with black lines.

CONCLUSIONS

In this work we proposed an alternative use of a GAN as seismic image processing operator. Specifically, we propose a GAN that builds upon (Isola et al., 2017) with a modified loss function tailored to seismic image processing. Through a preliminary experimental campaign, we show that it is actually possible to leverage recent findings in deep learning for different geophysical imaging applications. Future work will be devoted to study the generalization capabilities of GANs applied to seismic image processing.

ACKNOWLEDGMENTS

The authors would like to thank Nicola Bienati for his encouragement in focusing on this research topic.

A Generative Adversarial Network For Seismic Imaging Applications

APPENDIX A

THE SOURCE OF THE BIBLIOGRAPHY


```
@article{araya2018deep,
  title={Deep-learning tomography},
  author={Araya-Polo, Mauricio and Jennings, Joseph and Adler, Amir and Dahlke, Taylor},
  journal={The Leading Edge},
  volume={37},
  number={1},
  pages={58--66},
  year={2018},
  publisher={Society of Exploration Geophysicists}
}

@article{mccann2017convolutional,
  title={Convolutional Neural Networks for Inverse Problems in Imaging: A Review},
  author={McCann, Michael T and Jin, Kyong Hwan and Unser, Michael},
  journal={IEEE Signal Processing Magazine},
  volume={34},
  number={6},
  pages={85--95},
  year={2017},
  publisher={IEEE}
}

@inproceedings{isola2017image,
  title={Image-To-Image Translation With Conditional Adversarial Networks},
  author={Isola, Phillip and Zhu, Jun-Yan and Zhou, Tinghui and Efros, Alexei A},
  booktitle={Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition},
  pages={1125--1134},
  year={2017}
}

@incollection{goodfellow2014,
  title = {Generative Adversarial Nets},
  author = {Goodfellow, Ian and Pouget-Abadie, Jean and Mirza, Mehdi and Xu, Bing and Warde-Farley, David and Ozair
  booktitle = {Advances in Neural Information Processing Systems 27},
  editor = {Z. Ghahramani and M. Welling and C. Cortes and N. D. Lawrence and K. Q. Weinberger},
  pages = {2672--2680},
year = {2014},
publisher = {Curran Associates, Inc.},
}

@article{hall2016,
  author = {Brendon Hall},
  title = {Facies classification using machine learning},
  journal = {The Leading Edge},
  volume = {35},
  number = {10},
  pages = {906-909},
  year = {2016},
  doi = {10.1190/tle35100906.1},
}

@inbook{bestagini2017,
  author = {Paolo Bestagini and Vincenzo Lipari and Stefano Tubaro},
  title = {A machine learning approach to facies classification using well logs},
  booktitle = {SEG Technical Program Expanded Abstracts 2017},
```

```
  chapter = {},
  pages = {2137-2142},
  year = {2017},
  doi = {10.1190/segam2017-17729805.1},
}

@INPROCEEDINGS{pix2pix,
  author = {P. Isola and J. Zhu and T. Zhou and A. A. Efros},
  booktitle = {2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)},
  title = {Image-to-Image Translation with Conditional Adversarial Networks},
  year = {2017},
  volume = {00},
  pages = {5967-5976},
  month={July}
}

@InProceedings{unet,
  author="Ronneberger, Olaf
  and Fischer, Philipp
  and Brox, Thomas",
  editor="Navab, Nassir
  and Hornegger, Joachim
  and Wells, William M.
  and Frangi, Alejandro F.",
  title="U-Net: Convolutional Networks for Biomedical Image Segmentation",
  booktitle="Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015",
  year="2015",
  publisher="Springer International Publishing",
  pages="234--241",
}

@ARTICLE{spm2018,
  author={A. Lucas and M. Iliadis and R. Molina and A. K. Katsaggelos},
  journal={IEEE Signal Processing Magazine},
  title={Using Deep Neural Networks for Inverse Problems in Imaging: Beyond Analytical Methods},
  year={2018},
  volume={35},
  number={1},
  pages={20-36},
  month={Jan},
}

@ARTICLE{subsurfaceCNN,
  author={G. AlRegib and M. Deriche and Z. Long and H. Di and Z. Wang and Y. Alaudah and M. A. Shafiq and M. Alfarr
  journal={IEEE Signal Processing Magazine},
  title={Subsurface Structure Analysis Using Computational Interpretation and Learning: A Visual Signal Processing
  year={2018},
  volume={35},
  number={2},
  pages={82-98},
  month={March},
 }

@article{adam,
  author    = {Diederik P. Kingma and
               Jimmy Ba},
  title     = {Adam: {A} Method for Stochastic Optimization},
  journal   = {CoRR},
  volume    = {abs/1412.6980},
```

```
  year      = {2014},
  url       = {http://arxiv.org/abs/1412.6980},
  archivePrefix = {arXiv},
  eprint    = {1412.6980},
}
```

REFERENCES

AlRegib, G., M. Deriche, Z. Long, H. Di, Z. Wang, Y. Alaudah, M. A. Shafiq, and M. Alfarraj, 2018, Subsurface structure analysis using computational interpretation and learning: A visual signal processing perspective: IEEE Signal Processing Magazine, **35**, 82–98.

Araya-Polo, M., J. Jennings, A. Adler, and T. Dahlke, 2018, Deep-learning tomography: The Leading Edge, **37**, 58–66.

Bestagini, P., V. Lipari, and S. Tubaro, 2017, *in* A machine learning approach to facies classification using well logs: 2137–2142.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, 2014, Generative adversarial nets, *in* Advances in Neural Information Processing Systems 27: Curran Associates, Inc., 2672–2680.

Hall, B., 2016, Facies classification using machine learning: The Leading Edge, **35**, 906–909.

Isola, P., J. Zhu, T. Zhou, and A. A. Efros, 2017, Image-to-image translation with conditional adversarial networks: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 5967–5976.

Kingma, D. P., and J. Ba, 2014, Adam: A method for stochastic optimization: CoRR, **abs/1412.6980**.

Lucas, A., M. Iliadis, R. Molina, and A. K. Katsaggelos, 2018, Using deep neural networks for inverse problems in imaging: Beyond analytical methods: IEEE Signal Processing Magazine, **35**, 20–36.

McCann, M. T., K. H. Jin, and M. Unser, 2017, Convolutional neural networks for inverse problems in imaging: A review: IEEE Signal Processing Magazine, **34**, 85–95.

Ronneberger, O., P. Fischer, and T. Brox, 2015, U-net: Convolutional networks for biomedical image segmentation: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, 234–241.