Leveraging deep learning for seismic image denoising

Elena Klochikhina[1*], Sean Crawley[1], Sergey Frolov[1], Nizar Chemingui[1] and Tony Martin[1] describe a supervised machine learning approach for attenuation of noise, formed by suboptimal destructive interference within the migration process. The authors outline the training and validation approach of their deep convolutional neural network, and demonstrate its application on field data sets.

The impact of noise

Noise can affect the quality of seismic data, damaging the geological integrity of the final migrated image; therefore, we should minimize its impact. If not properly attenuated it may affect amplitude-related attributes, lead to difficulties during the quantitative interpretation (Cambois, 2001; Ball et al., 2011), and result in an inaccurate appraisal of the reservoir.

Noise can take many forms. In this paper, we consider the noise generated during the migration stage. The noise sources might be as diverse as residual impulsive noise, multiple energy, mispositioned primary energy due to errors in estimation of earth properties, or insufficient illumination caused by limitations in the data acquisition, and/or complex media-related propagation effects. In each case the migrated image may be affected by suboptimal destructive interference of the migration isochrones (Gardner and Canning, 1994), resulting in a contamination of the data by coherent noise. We concentrate on the latter case where the resulting noise is due to inadequate illumination in a complex media.

An experienced geophysicist may easily differentiate signal and noise in a seismic section. It can be challenging to remove the noise without affecting the signal, because the coherent noise often has similar seismic characteristics to the desirable components of the data. Several approaches can minimize the migration artefacts. These include data regularization prior to migration, filtering during migration, post-processing after migration, and least-squares migration methods.

Data reconstruction is often used to overcome irregularities in data coverage (Chemingui and Biondi, 2002; Schonewille et al., 2009). Depending on the method and migration algorithm, regularized data may produce less noise, but can require a significant effort in the data preparation. Moreover, regularization may impact the resolution of the final image. Aperture optimization may also reduce the impact of noise generated in the migration process (Alerini and Ursin, 2009; Klokov and Fomel, 2012), but requires the knowledge of local structural dips, and results are dependent on the accuracy of such dip information. There are pragmatic alternatives such as filters that can be designed to attenuate the noise in the image domain (Hale, 2011). As the noise is often coherent, sharing seismic characteristics with the signal, it can be challenging to design filters that only remove the unwanted components but preserve the useful energy.

The adoption of artificial intelligence

In the early days, computers learnt how to solve problems that were intellectually difficult for humans by following a sequence of strict mathematical rules. The true challenge was to create a machine that could solve problems that humans solve intuitively, problems that are hard to describe in formal rules. This knowledge somehow needs to be captured by a computer to behave in an intelligent way (Goodfellow et al., 2017). This is a challenge for creating an artificial intelligence (AI). To overcome the problem the field of machine learning (ML) was born: AI systems were given the ability to acquire knowledge by extracting patterns from the data and gather knowledge from experience. Classical ML methods are highly dependent on features that are prepared by humans. It could be time consuming and also difficult to extract and provide a right set of features without human bias in order for an ML algorithm to perform well. Deep learning (DL) overcomes this problem by extracting information from raw data: complex representations can be learnt from the input by decomposing the data into simpler intermediate representations. DL models look at the data on different scale levels layer by layer. Deep learning has an ability to perform automatic feature extraction from raw data without depending completely on human-crafted features. Together with advanced architectures and optimized training approaches, increase of training data for DL algorithms can help in reaching human performance on complex tasks by learning from a vast variety of examples.

Data is not in short supply within the seismic processing business, however the adoption of artifical intelligence has not been as extensive as in other data-rich industries. There is evidence that it is changing, as seismic companies seek to augment decision making and reduce project cycle times. The number of papers and manuscripts describing applications of artificial intelligence and data analytics has grown at geophysical conferences and in geophysical journals. The trend in paper numbers suggests meth-

[1]PGS
[*] Corresponding author, E-mail: elena.klochikhina@pgs.com
DOI: 10.3997/1365-2397.fb2020048

ods invoking artificial intelligence-enabled automation may be the future of our industry. The combination of data and computer science with geophysics may be applicable to every aspect of a seismic processing project. From unsupervised (Martin et al., 2015) to supervised (Farmani and Pedersen, 2020) classification of denoising workflows, and support vector regression for data interpolation (Jia and Ma, 2017) to parabolic dictionary learning for data reconstruction (Turquais et al., 2019), most aspects of data domain processing are being tested. Using a variety of neural network approaches, efforts are being made to compare and contrast the velocity model building with conventional inversion-based schemes (Øye and Dahl, 2019; Yang and Ma, 2019; Zheng et al., 2019). The reported results look encouraging.

Using a deep convolutional neural network for image denoising

For coherent noise attenuation, rather than explicitly formulating filters as in conventional methods, we present an AI approach that utilizes a deep convolutional neural network (CNN) to achieve the same goal. The main components of CNNs are convolutional filters that are iteratively adjusted during the training step to handle the artefacts and produce clean outputs from the noisy inputs. The trained models are then used to denoise the seismic images from field experiments.

A neural network may act as a universal function approximator to mimic the characteristics of a complex function F that maps a noisy input x to a noise-free output y:

$$F(x) = y \tag{1}$$

The goal of the training is to find a transformation $F'(x,\theta)$ that maps x into a set of corresponding y. To do this we minimize a cost function (J), that is defined as a difference between the transformed inputs y' and the desired clean outputs y, with respect to the parameters of the trained network θ in $L2$ sense:

$$F'(x,\theta) = y' \tag{2}$$

$$J = \| y - y' \| \tag{3}$$

The stability and quality of predictions depends on the network architecture, the hyperparameters and the training data set. In our example, the training data were created using noisy images as an input to the CNN and clean images (noise-free) as an output.

The architecture of a convolutional neural network contains a number of different operations; the goal of the trained network is to replicate human endeavour. In our case we are trying to identify and remove coherent noise from a seismic image. A typical CNN architecture for computer vision problems consists of a number of different components that include convolutional layers and activation functions. It may also contain other operations like downsampling (or pooling), upsampling, batch normalization, etc. All network components are connected in the form of a graph.

The essential components of CNN architectures are convolutional layers separated by non-linear activation functions. Convolutional layers use filters, which are also known as kernels. Each filter is an array consisting of a sequence of numbers or weights. The filter slides over the image, only being exposed to a small number of input pixels at any time. The operation used is a dot product of the input values with the filter weights. The output is a single number per sliding window. The results over the entire input is called an activation or a feature map. Depending on the filter, each activation map identifies distinguishing features; with each progressive convolutional layer, more complex features can be determined. Non-linearity is a crucial part that allows the neural network to approximate complex operations necessary for solving a given task. The so-called activation functions are used for this purpose.

There are many other components that could be used in CNN architectures. In our denoising architecture, we use a pooling step for downsampling. There are several types of pooling, and we used a maxpooling operation, which selects the largest number from the neighbouring cells during the downsampling process. This reduces the spatial dimensionality of the input data scale, limiting computational cost and increasing the exposure of the input for the next convolutional layer. The opposite operation is upsampling that is used to refine spatial sampling of the feature map.

The hyperparameters of a convolutional neural network are its structure, components, and training specifications. To achieve

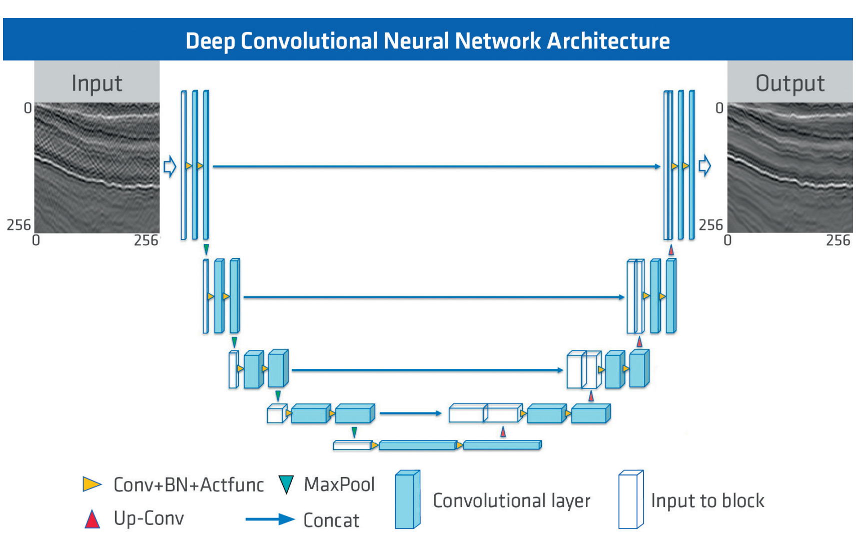

Figure 1 The architecture used for the deep learning process.

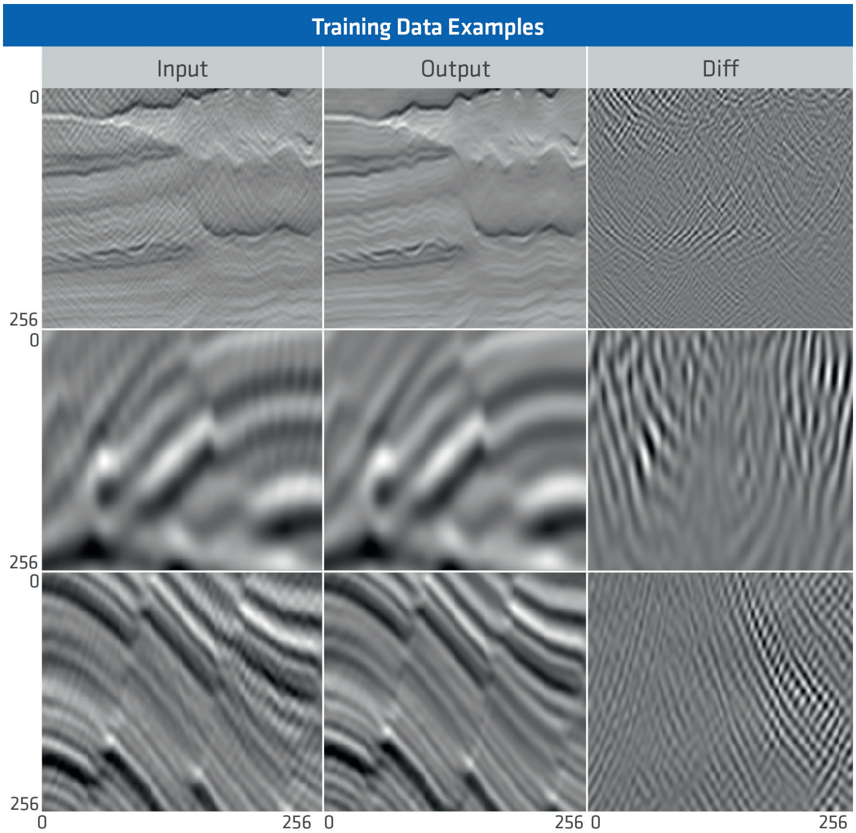

Figure 2 Examples of data used in the training of the convolutional neural network.

the best performance of the convolutional neural network, we optimize the hyperparameters through testing. In practice, the network is trained using user defined data. This is critical for the success of the neural network to achieve its goals. The data needs to be representative of the problem we are trying to solve. The training happens over the course of multiple epochs. Each epoch consists of multiple iterations. Each iteration uses a subset of the input data set, called a batch. On average, every epoch passes through all input data samples once. Data augmentation enables a modification to the pool of training data. It is one way to increase the number and variability of the data set, enabling a more robust prediction and resulting in an increase in the level of sophistication of the trained network.

Specifics of the architecture

Among the wide variety of commonly used network architectures for image denoising in computer vision, we considered the U-net architecture (Ronneberger et al., 2015) to be most suitable for this problem. During our testing phase, it showed better convergence, faster training and solves the problem naturally as it enables operations on different feature resolutions (Figure 1).

The architecture consists of three parts: the contraction (left branch), the bottleneck (bottom) and the expansion (right branch). Each convolutional layer receives an input and applies a set of 3x3 filters, followed by a nonlinear activation function.

The contracting path consists of four blocks; each block has two convolutional layers followed by a downsampling procedure (maxpooling). The number of filters in the convolutional layers doubles each time the resolution decreases, so the architecture retains the ability to explain complex features present in the input.

The bottom layer takes an input from the left branch and applies two convolutional layers.

The expansion path receives the input from the bottleneck and also consists of four blocks; each block has two convolutional layers followed by an upsampling procedure. After each upsampling step, the number of filters in the convolutional layers halves.

The corresponding blocks of the contraction and expansion paths are connected by 'skip connections'. This helps to solve the problem of a vanishing gradient during the training stage and simplifies the prediction task, as there is no need for reconstruction of the image at full resolution from its compressed representation.

To accommodate the challenge, we modified the convolutional blocks of U-net and fine-tuned the hyperparameters of the network during the training process to achieve better performance of the neural network. In order to reduce the likelihood of overfitting, we added dropout layers.

We modelled synthetic shot gathers, which were migrated to form the noisy inputs and clean outputs used in the training step. We subsampled and migrated the synthetic data to generate the coherent noise in the images. The noise-free output consisted of clean images from the migration of appropriately sampled data. The image patch size for training was 256x256 pixels (Figure 2).

We carefully selected the data set so that it included variations in the following: frequency content, structural dip, amplitude, and noise character and level. To increase the variability of the input, we used data augmentation, which included horizontal flips, random crops and sign reversal, filtering and scaling with depth, resulting in approximately 100,000 total input samples. Hyperparameters such as learning rate schedule, dropout rate and batch size were adjusted during the training phase to minimize the prediction error. We trained the network for 50 epochs on a single GPU with 32 Gb of memory.

The challenge of overfitting

A common challenge for machine learning algorithms is overfitting. This occurs when the trained network's performance shows great promise on the data used for training, but has a poor success rate when attempting to generalize on previously unseen data.

This happens when the capacity of the model is too large, compared to the diversity of a data set used for building the model. With neural networks, this occurs when there are too many parameters. The model may provide great flexibility and approximation power, but the amount and variability of the data given to it is not enough to constrain the weights within the network, at least not without regularization. As a result, the network makes unreasonable predictions for any data that differs from the training set, in our case frequency, amplitude or noise level. As a precaution, the input data set is split into two subsets, one for training and the remainder for validation. We then monitor the trained model's performance on the latter. A gradual decrease of the loss function for both training and validation data sets implies reasonable generalization, assuming fair selection of the validation data set.

To reduce the overfitting problem, we used a dropout technique. During the training step, we carefully monitored the behaviour of the objective function for both the training and validation data sets, assuring the proper behaviour whilst preserving an effective convergence (Figure 3).

Case studies

The denoising capabilities of the neural network were tested on two field data sets, one from a deep water survey offshore Brazil, the other from a shallow water example in the North Sea. In both cases, insufficient illumination and complex media cause coherent noise in the images that have similar seismic characteristics just as the signal we want to preserve. The case studies demonstrate the ability of the trained network to attenuate the noise from data sets that represent two geologically different settings. We used only synthetic data to train the neural network; therefore, the case studies demonstrate the ability of the network to generalize outside the training data set.

We compared performance of the CNN-based denoising tool with an application of a commonly used structure-oriented filter (Hale, 2011). We could parameterize the filter differently to preserve the primary energy; however here we were focused primarily on the noise attenuation aspect. More aggressive filter settings can better eliminate the noise at the cost of damaging image resolution.

Example one – offshore Brazil

In the first example from Brazil, there is strong and pervasive coherent noise. The upper yellow arrow in Figure 4 highlights where this is most evident. The middle yellow arrow in the same figure shows coherent noise above a high contrast and rugose surface. The migration noise directly overlying the unconformity distorts the seismic events, making interpretation challenging. In all cases, the noise shares seismic characteristics with the signal that we want to use and preserve, such as dipping fault planes, and the flanks of the deeper steep-sided body. Figure 5 shows the result of using the CNN on the input data. The blue arrows show the removal of the coherent noise. The reflectivity above the rugose unconformity is more continuous, no longer disrupted by the noise forms, and there is no noise contamination of the data abutting the deep steep-sided body.

The denoised section is much cleaner, and reflectivity is easier to track. Steep dipping energy, such as fault planes, are still present. The difference of the application (Figure 6) demonstrates the impact of the CNN – a large amount of noise has been removed. There are indications that the process has attenuated some steep dipping energy that correlates with the noise; however, the output section (Figure 5) shows that much of this energy remains unscathed.

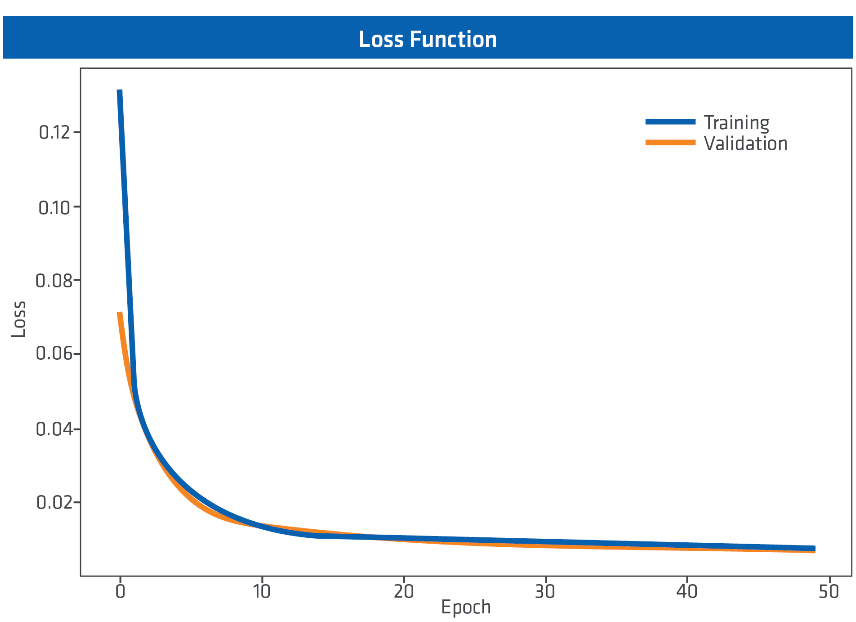

Figure 3 Objective functions for both training and validation processes. The two were monitored during the process to confirm equivalent convergence for both.

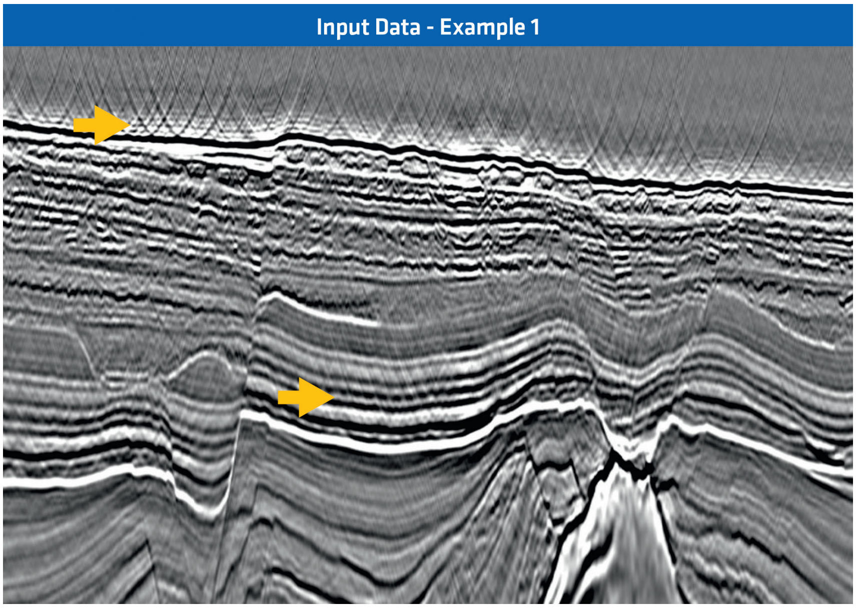

Figure 4 Input seismic data. The yellow arrows show the migration-related noise the CNN model is attempting to remove.

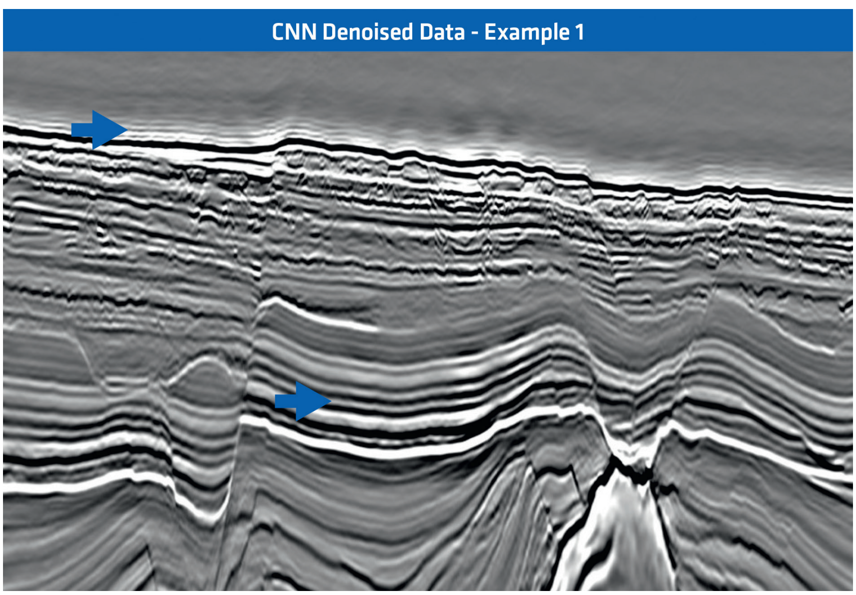

Figure 5 CNN denoised data. Blue arrows indicate that the model has removed almost all the coherent noise from the seismic section.

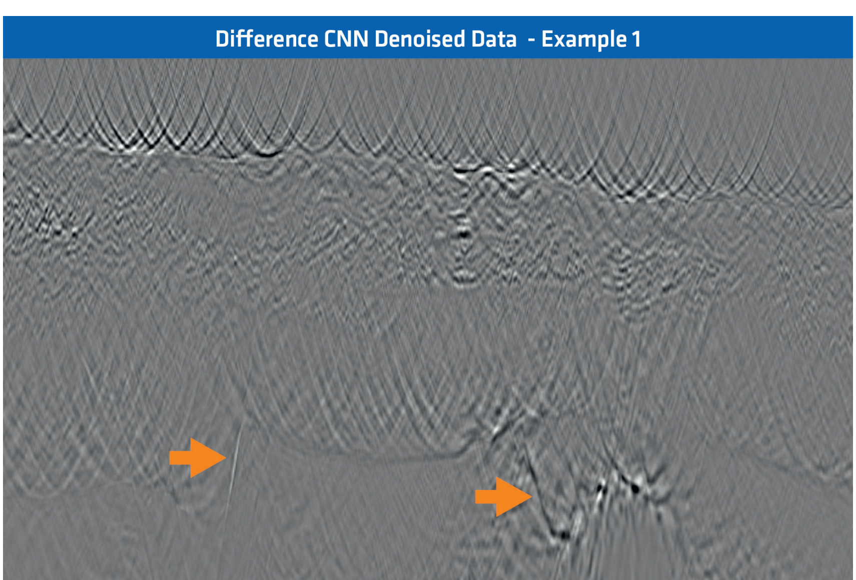

Figure 6 The application difference shows the energy removed from the seismic section. Orange arrows indicate some primary energy has been attenuated.

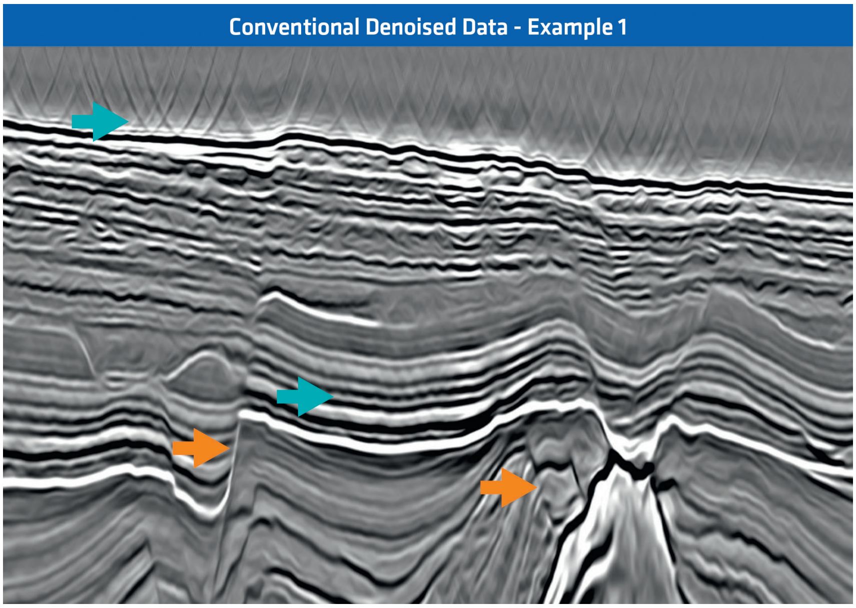

Figure 7 The application of a conventional denoise process. The turquoise arrows indicate the residual noise left in the data, whilst the orange arrows highlight the attenuation of desirable signal.

Figure 7 demonstrates an application of a structure-oriented filter. The upper turquoise arrows indicate locations where the noise is still present, whilst the lower orange arrows show where the process has removed the complementary steep dipping signal we would ideally preserve. It is also important to emphasize that the application of the structure-oriented filter affected image resolution. This conventional method has not been as effective as the CNN approach.

Example two – North Sea

In the second example, poor illumination of a single high-contrast and undulating event causes localized migration-related noise (yellow arrows – Figure 8). The noise swings upwards disrupting the events directly above the rugose event, making interpretation challenging.

The conventional denoise approach using structure-oriented filters mitigates some noise (turquoise arrows – Figure 9), but also smears some of the isolated injectites located in the shallower layer (orange arrows – Figure 9).

The injectites have localized reservoir potential. In Figure 10 (blue arrows), we see that the trained neural network has removed more noise; the events overlying the source of the noise are easier to interpret. Figure 11 shows the impact of the deep-learning approach. The noise has almost been eradicated. It is worth noting the orange arrows on Figure 11. They show that the injectites are affected by the denoise process. Their shape is very similar to the coherent noise created in the migration process, as each migration-related noise form has an apex. Consequently, the process does attenuate some energy from the injectites but no more than the conventional approach, which underperforms on the migration related noise attenuation.

Discussion

The goal of this work is to demonstrate the denoising capabilities of a convolution neural network, in particular the attenuation of coherent noise formed during the migration process. The network was trained using approximately 100,000 input samples, and

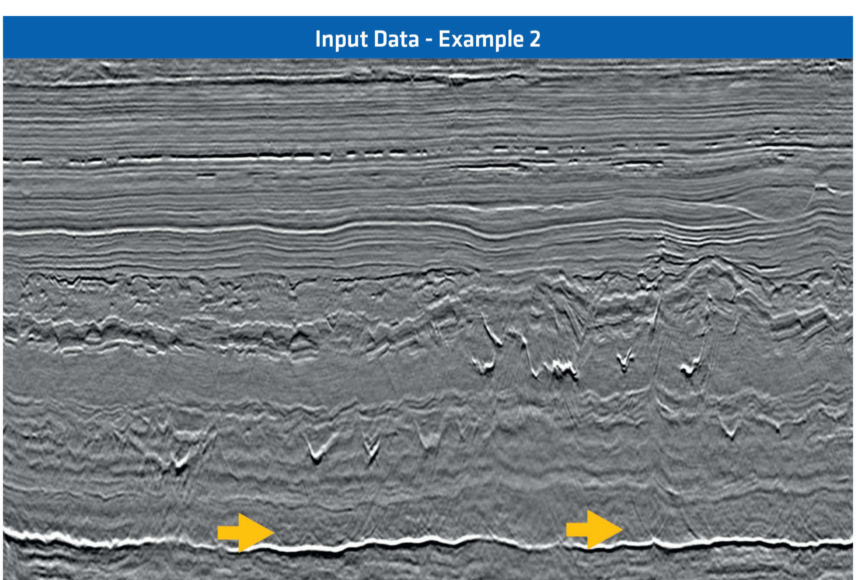

Figure 8 North Sea input seismic data. Yellow arrows highlight the offending coherent noise.

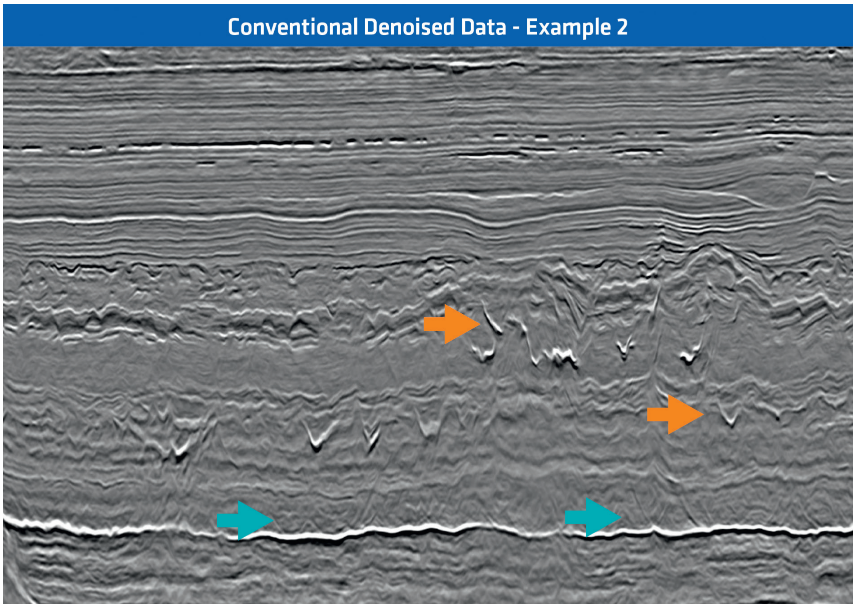

Figure 9 The structure-oriented filtering result shows residual noise (turquoise arrows), and a smearing energy of shallower injectite energy (orange arrows).

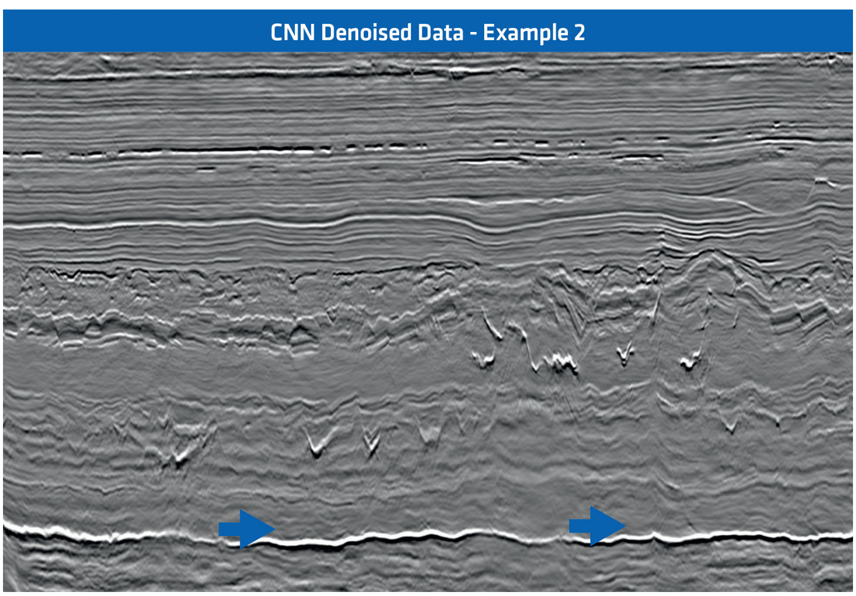

Figure 10 The output from the application of the convolutional neural network's model. The blue arrows highlight the effectiveness of the process; almost all coherent noise has been removed.

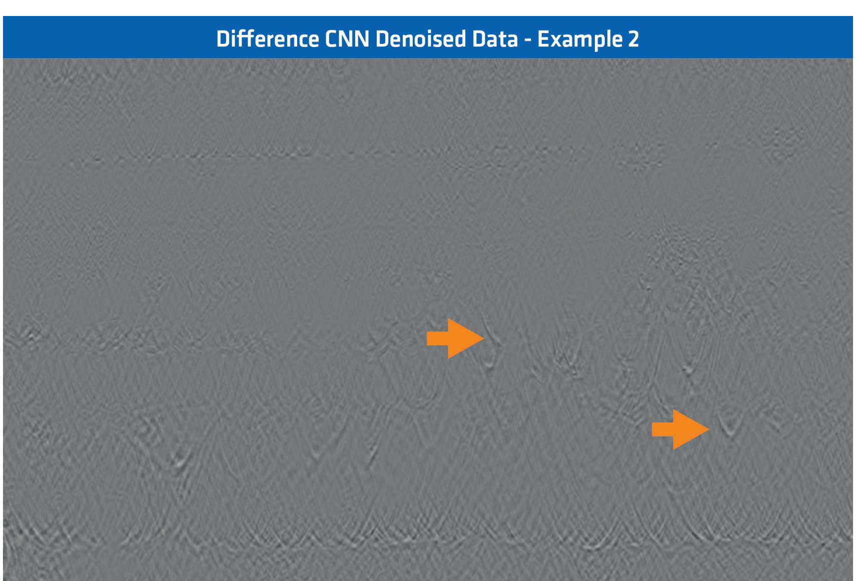

Figure 11 The difference of the deep learning approach shows the removal of the offending noise. Some injectite energy has also been attenuated, but not fully removed.

benefited from an augmentation process. The noise forms have a 3D nature. However, each input sample to the neural network training was 2D, as was the training process. Despite this, the transferability of the trained network to unseen field data shows significant potential as the noise forms we were attempting to remove had a consistent nature. How generalizable the tool to other forms of noise is still to be determined.

Improving the training process in a 3D sense would improve the results for noise that is three dimensional. In the current training process, the majority of the work was collecting an appropriate population of input samples, whereas the neural network training took an insignificant time once the optimum combination of hyperparameters was established. To extend the training to 3D would require considerably more effort. The network would become larger, and the tuning of the hyperparameters would be far more convoluted. Whilst this would be a more memory and compute intensive process, the trained network should be able to differentiate noise and signal more effectively.

It is also worth noting that we have focused on the effectiveness of the CNN to remove noise in post-stack images. The extension of the approach to the pre- or sub-stack image domain data, for attribute analysis, still needs to be confirmed. Empirical evidence from our existing tests suggests that an application to pre-stack data will also assist amplitude-related attribute generation and analysis.

Conclusions

We have demonstrated a deep learning approach for attenuating migration noise from seismic images. Historically, removing this type of noise from seismic data has been challenging, as it shares many characteristics with the signal that we would like to preserve. We have trained a convolutional neural network to differentiate the noise from geological structures.

The application of the trained model to two field data sets demonstrates the potential of the solution for successfully attenuating migration noise without compromising the resolution or structural integrity of the seismic image — we see improvements in both structure and amplitude fidelity of the seismic image.

Acknowledgements

The authors wish to thank PGS for permission to publish the paper and PGS MultiClient for providing the field data examples.

References

Alerini, M. and Ursin, B. [2009]. Adaptive focusing window for seismic angle migration. *Geophysics*, **74** (1), S1–S10.

Ball, V., Blangy, J.P., Pringle, K. and Schwark, J. [2011]. Seismic rock physics in the presence of attribute noise. *81st SEG Annual Meeting,* Extended Abstracts*,* 355-359.

Cambois, G. [2001]. AVO processing: Myths and reality. *71st SEG Annual Meeting,* Extended Abstracts.

Chemingui, N. and Biondi, B. [2002]. Seismic data reconstruction by inversion to common offset. *Geophysics*, **67**, 1575–1585.

Farmani, B. and Pedersen, M. [2020]. Extended Attributes for Machine Learning Denoise Process: First Step Towards Automation. *82nd EAGE conference and Exhibition,* Extended Abstracts.

Gardner, G. H. F. and Canning, A. [1994]. Effects of irregular sampling on 3-D prestack migration. *64th SEG Annual International Meeting.* Expanded Abstracts, 1553-1556.

Goodfellow, I., Bengio Y. and Courville, A. [2017]. *Deep learning.* MIT Press, Cambridge, MA.

Hale, D. [2011]. Structure-oriented bilateral filtering of seismic images. *81st SEG Annual Meeting,* Expanded Abstracts*,* 3596-3600.

Jia, Y. and Ma, J. [2017]. What can machine learning do for seismic processing? An interpolation application. *Geophysics*, **82** (3), 163-177.

Klokov, A. and Fomel, S. [2012]. Optimal migration aperture for conflicting dips. *82nd SEG Annual Meeting,* Expanded Abstracts*,* 1-6.

Martin, T, Saturni, C., Ashby, P. [2015]. Using machine learning to produce a global automated quantitative QC for noise attenuation. *85th SEG Annual Meeting,* Expanded Abstracts, 4790-4794.

Øye, O. K. and Dahl, E. K. [2019]. Velocity Model Building from Raw Shot Gathers Using Machine Learning. *81st EAGE conference and Exhibition,* Extended Abstracts.

Ronneberger, O., Fischer, P. and Brox, T. [2015]. U-Net: Convolutional Networks for biomedical Image Segmentation. Medical Image Computing and Computer-Assisted Intervention (MICCAI), **9351**, 234-241.

Schonewille, M., Klaedtke, A., Vigner, A., Brittan, J. and Martin, T. [2009]. Seismic data regularization with the anti-alias anti-leakage Fourier transform. *First Break*, **27**, 85-92.

Turquais, P., Söllner, W. and Pedersen, M. [2019]. Parabolic Dictionary Learning: A Method or Seismic Data Reconstruction Beyond the Linearity Assumption. *81st EAGE Conference and Exhibition,* Extended Abstracts.

Yang, F. and Ma, J. [2019]. Deep-learning inversion: A next-generation seismic velocity model building method. *Geophysics*, **84** (4), 583-599.

Zheng, Y., Zhang, Q., Yusifov, A., Shi, Y. [2019]. Applications of supervised deep learning for seismic interpretation and inversion. *The Leading Edge*, **38** (7), 526-533.