BUILDING AND UNDERSTANDING DEEP NEURAL NETWORKS COMPONENTS FOR SEISMIC PROCESSING: LESSONS LEARNED

M. Chambefort[1,2], J. Messud[1]

[1] CGG; [2] MinesParisTech - PSL

Summary

Learning how to best mimic seismic processing algorithms or workflows with deep learning (DL) has become a very active field of research. However, seismic processing own particularities may necessitate adaptations of current DL methods. In this paper, we explain and illustrate how the different DL components can affect the outcome of a given seismic processing task. Among others, we show that the Unet neural network architecture (Ronneberger et al., 2015) is naturally suited to learn how to “separate” the events into kinematics and their amplitudes, and how to use both information efficiently to perform the common image gathers preconditioning, skeletonization (or picks probability computation) and muting task. We also show how the convolution kernel shapes, the number of layers, the training cost function and the batch size can be adapted to specific data and seismic processing tasks.

Introduction

Since its recent successes in natural image classification and analysis (LeCun et al., 2015), deep learning (DL) has gained significant attention in many industrial fields, including in seismic processing. There, most of the current investigations consist in learning to mimic seismic processing algorithms or workflows, i.e. to predict the processed image from the "initial" image using deep neural networks (DNN); see e.g. Sun et al. (2018), Mandelli et al. (2019), Richardson and Feller (2019), Ovcharenko et al. (2019), Sen et al. (2019), Yuan et al. (2019). However, seismic processing has its own particularities, which may necessitate adaptations of current DL methods.

Firstly, seismic data or « images » are very different from natural images by their content:

- They are made of "laterally" coherent events that can be decomposed into different physical contributions such as kinematic, amplitude and wavelet.
- They are signed and oscillatory, with a frequency bandwidth (typically 2.5-150 Hz).
- The frequency bandwidth and the kinematics, as well as the amplitudes in many cases, are very important to be preserved for further processing tasks.

Secondly, many state-of-the-art algorithms and workflows are available for seismic processing. They are physical, use regularizations and can be tuned with geophysicist's expertise through user-defined parameters. The corresponding processes (algorithms together with user's tuning) are highly non-linear, and being able to benchmark them represents a challenge for DL. In addition, a comparison with those processes highlights that operations performed in DNN are not straightforward to interpret physically. This occasionally leads to consider DNNs as "black boxes", and the effort is sometimes put more on: defining good training data (1) and DNN training or optimization techniques (2) rather than on understanding the optimum DNN components according to the task (3); see e.g. Mandelli et al. (2019), Richardson and Feller (2019), Ovcharenko et al. (2019), Sen et al. (2019), Yuan et al. (2019). A step towards point (3) is sometimes made for instance by fine-tuning of DNN components (Sun et al., 2018).

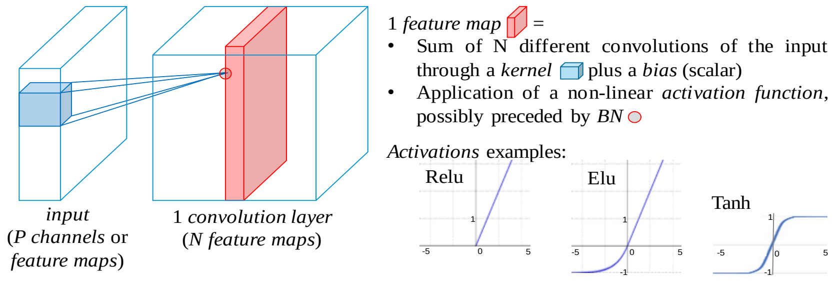
Without neglecting the importance of point (1), we focus here on point (3) and somewhere on point (2), and study how DNN components and training can affect the outcome of a given seismic processing task. Experimenting with the Unet convolutional DNN (Ronneberger et al., 2015), we show how:

- Convolutional layers learn to extract and combine meaningful seismic features in an efficient way.
- Data and task particularities can guide the choices for the optimum number of feature maps, convolution kernel shapes, as well as training parameters like cost function and batch size.

We start with some reminders and illustrate our claims with the example of CIG (common image gathers) preconditioning, "skeletonization" and muting. This processing task is necessary for the picking of RMO (residual move out) curves, the "skeleton" representing picks probability, and can be used in tomography velocity model building (Lambaré et al., 2014) or RMO corrections (Siliqi et al., 2003).

Convolutional layers and signal processing (reminder)

Convolutional layers-based neural networks are especially well suited for image processing. Indeed, convolutions are a natural building block of many standard signal processing algorithms. Convolutions themselves being a linear process, non-linearity in a convolutional layer (necessary to represent seismic non-linear processes) is brought by application of a non-linear activation function (e.g. Relu, Elu, Tanh…). The latter is preceded by the application of a bias term and often of a batch-normalization (BN) procedure, which allows overcoming some training pathologies related to activations functions "saturation" or null-derivatives. The trained parameters are the convolution kernels and biases. Each convolutional layer is composed of many feature maps that learn to "extract" data features (they are related to different convolution kernels). Goodfellow et al. (2016) give details, and the Fig. 1 together with the following equation, illustrate the key elements:


1 *feature map* =

- Sum of N different convolutions of the input through a *kernel* plus a *bias* (scalar)
- Application of a non-linear *activation function*, possibly preceded by *BN*

Activations examples:

Figure 1: *A convolutional layer's components.*

$$\forall i \in [1, N]: \quad featuremap_i = activation\left\{BN\left(bias_i + \sum_{j=1}^{P} input_j * kernel_{ij}\right)\right\},$$

where $*$ represents a convolution, which can be (Goodfellow et al., 2016):

- The standard convolution, leading to feature maps of the same size as the input images.
- A contracting (or strided) convolution, leading to smaller feature map size than the input.
- An expansive (or transposed) convolution, leading to bigger feature map size than the input.

Unet customizations and skeletonization

Unet (Ronneberger et al., 2015) is a DNN architecture commonly used in natural image analysis, combining convolutional layers in a contracting and expansive path with skip connections. It has proven to be very successful for segmentation and contour/edge detection. As the latter task shares similarities with skeletonization, we naturally studied how Unet performs to achieve the following processing:

- Learn CIG skeletonization (or picks probability), i.e. separation of kinematics from amplitudes/wavelet for each seismic event. This part is highly non-linear, classical picking algorithms involving global optimizations (Siliqi et al., 2003).
- Learn simultaneously to mute where events are largely stretched, or where events amplitudes dim drastically. Fig. 2 gives an example of training data, where the green line schematizes the total mute rough delineation.

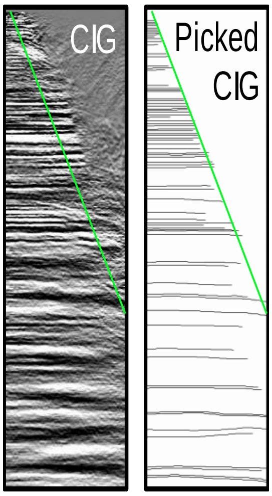


Figure 2: *A data example, with the mute roughly schematized in green. Data size is 1800×70 samples (depth×offset).*

Those tasks are difficult for DL as they involve the learning of a highly non-linear workflow (not just one algorithm), and the production of a very sparse and multi-event output that preserves the kinematics. That represents an additional challenge compared to the already explored first break picking application by DL (Yuan et al., 2019), which involves picking a single event for each trace.

After some customizations described thereafter, we found Unet particularly efficient for the CIG skeletonization and muting task. We started by studying the compromise between capacity (or parameters space size, driven by the chosen number of feature maps) and generalization (or ability to predict data "unseen" by the training). With different surveys, we found that the compromise is good enough when the architecture of Fig. 3 is used. The main difference with the original Unet of Ronneberger et al. (2015) lies into the number of feature maps in the convolutional layers, much smaller and decreasing with coarser levels; we analyze why in the next section, from the task and data point of view. This "smaller" or "good compromise" Unet has also the advantage that it eases our next analyses.

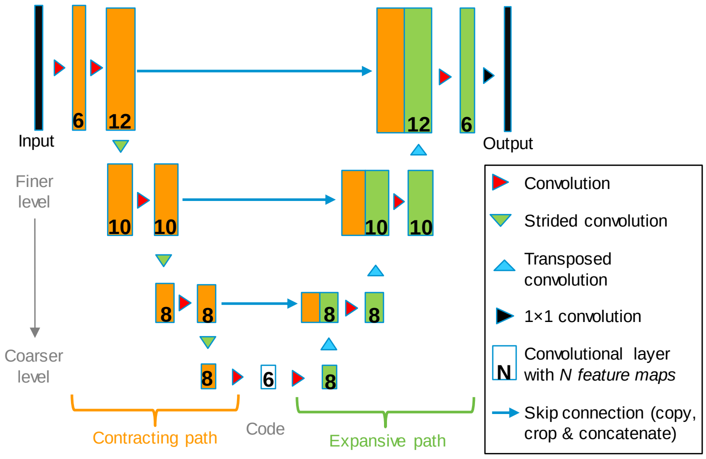


Figure 3: *Customized Unet architecture used in this article.*

The adjustments of other hyper-parameters (Goodfellow et al., 2016) of our customized Unet are:

- Larger convolution kernels (6×7) perform better than original Unet kernels (3×3), see further below.
- (3×2) strides for contracting and expansive convolutions (e.g. contracting more in the depth direction), allow to increase the efficiency without quality loss (the CIGs having much more samples in this the depth direction).
- Elu or Tanh internal activations tend to perform slightly better than Relu within our implementation (where seismic is rescaled in [-1,1], oscillating around 0; thus, pre-bias pre-activations also tend to oscillate and Relu, null for negative values, has a bit more difficulty to capitalize on this behavior).
- Sigmoid function, as an output (or last convolution) activation allows to interpret the predicted skeleton as a pick probability for each sample.
- For the training, we use the binary cross-entropy cost function. It is well suited for a probability to promote sparsity, as illustrated further below. We used the adaptive learning rate optimization (Adam) method (Goodfellow et al., 2016), batch size of 12 and trained the model for 200 epochs.

Convolutional layers learn to extract and combine meaningful seismic features in an efficient way

Let us analyze and understand the trained Unet feature maps. Fig.4 shows some of them:

- Fine level feature maps learn to highlight the "peaks" or kinematic information (wavelet removal), i.e. to sharpen the coherent events in various directions; but they do not learn the mute.
- Coarser level feature maps highlight more and more the amplitude (or the "texture") changes that occur naturally thanks to the loss of resolution due to contraction. The coarsest "code" layer feature maps are related to gross amplitude changes only, allowing to more easily learn the mute properties.
- The expansive path together with skip connections allow to combine the kinematic and pertinent amplitude information extracted in the contracting path, and so to learn skeletonization and muting. (Visible in "last layer" of Fig. 4, transposed convolutions create a checkerboard pattern that easily cancels out after last 1×1 convolution as this pattern is at the same position in each feature map).

Unet architecture thus naturally can learn to "separate" events kinematics from their amplitudes/wavelet properties, and to use both information in an efficient way for a task. This "separation" is physically meaningful for skeletonization and muting, and also for seismic processing in general (see introduction).

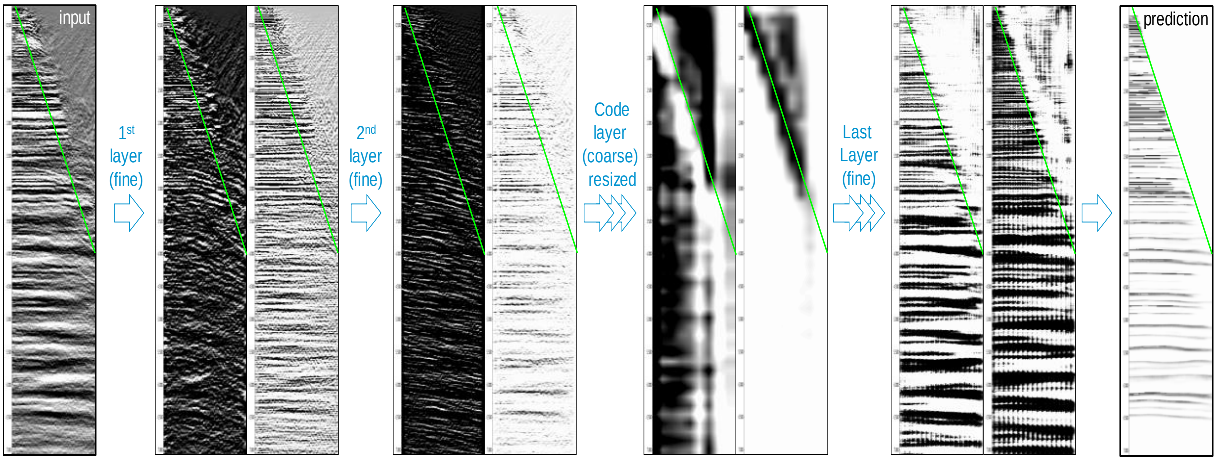
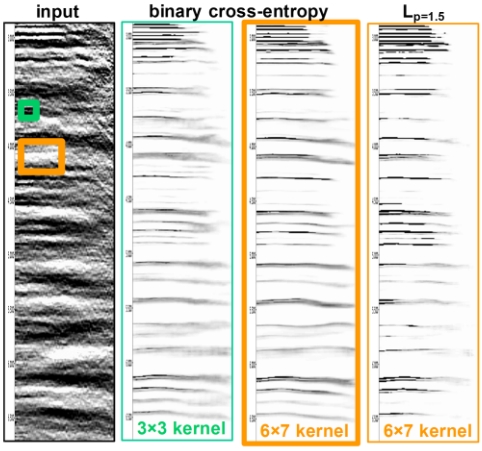


Figure 4: *For our Unet skeletonization and muting task: Left image is the input; Then 2 feature maps per convolutional layer are shown, going deeper and deeper in the DNN (for 1st and 2nd layers of contracting path, code layer and last layer of expansive path, see Fig. 3); Right image is the prediction.*



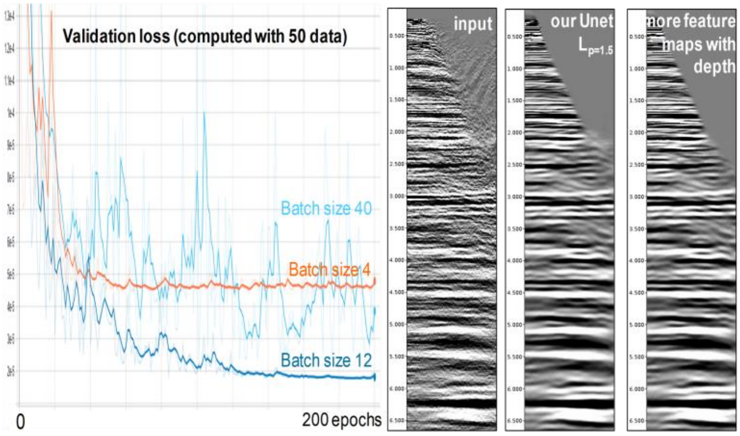


Figure 5: *Skeletonization and muting: Effect of kernel size and cost function.*

Figure 6: *Preconditioning and muting. Left: Effect of the batch size. Right: Effect of the feature maps number.*

Optimum kernel shapes, number of layers, cost function, batch size and data particularities

Correlations along a seismic event must be taken into account for the continuity and sparsity of the skeleton. Consequently, kernels, which are more elongated in the offset-direction and enough elongated to frame the wavelet in the depth-direction, tend to provide better results (see center images of Fig. 5).

Events continuity and sparsity can be promoted during the training through the cost function. For a probability output (in [0,1]), binary cross-entropy is a good choice to promote sparsity (Goodfellow et al., 2016) while preserving continuity, see Fig. 5. L_p norms ($1 \leq p < 2$) are also known to promote sparsity, but they are less efficient to preserve skeleton continuity (two right images of Fig. 5). However, for seismic outputs (whose statistical distributions tend to be symmetrical around 0), the use of L_p norms make sense in the case of CIG preconditioning and muting task (two left CIGs of Fig. 6). Interestingly, we observed that a sharper mute and better denoising is learnt using p in [1,1.5] rather than p=2 (least-squares). Also, the DNN that represents the "good compromise" for this task is the same as for the skeletonization task (Fig. 3); only the output activation has been changed (Tanh in [-1,1]).

A property of seismic data that is also worth exploiting during training is their relatively low diversity within a survey. Consequently, a gradient (for DNN model update) computed with very few data, i.e. small batches, can still be representative. It can even be worthwhile: small batches give a noisy version of the gradient that can help escaping the local minima. This explains our observation that small batches in general provides better trainings for seismic applications. However, a compromise must sometimes be found as in the case studied here. In the Fig. 6, the validation curves shown on the left indicates that the optimum batch size is 12 here, and that smaller batches lead to a less good result (too noisy gradients may have difficulties to converge towards a good local minimum).

We have seen that coarser layers are more related to seismic gross amplitude/texture variations. As those variations are usually quite limited in CIGs and seismic data in general, describing them through few coarser level feature maps should lead to reasonable results. This is illustrated in the two right images of Fig. 6 (for the preconditioning and muting task), where increasing the number of feature maps with coarser level (here going from 12 for finer level to 14, 16, 18 for coarser level and 20 for code) brings only a very slight improvement.

Conclusion and challenges

We explained and illustrated how the different components of a DNN can affect the outcome of a given seismic processing task, and how they are related to data particularities. Among others, we showed that the Unet architecture can naturally learn to "separate" the kinematics of events (finer levels feature maps) from their amplitude variations (coarser levels) and to use both information efficiently for a given processing task. Our analysis underlined that some key DNN components can be understood depending on the task and data specificities, and are therefore worth being tested and tuned.

Many DL for seismic processing challenges remain, relating to (1) defining intelligently good training data, (2) ensuring the best training and (3) understanding the optimum DNN components according to the physics. In this paper, we focused on some aspects of points (3) and (2), and on some specific tasks. Systematic study of all those three points would certainly be very beneficial to the seismic community.

Acknowledgements

We are grateful to Lundin and CGG for the permission to publish this work, and to N. Salaun, A. Henry-Baudot, G. Pignot, G. Gigou, P. Poggi, N. Deladerriere, G. Lambaré, B. Chaintreuil, H. Peng, C. Lacombe, G. Viguier, N. Chazalnoel and S. Hou for their involvement and impactful discussions.

References

Goodfellow, I., Bengio, Y., Courville, A. [2016] Deep Learning. *MIT Press*.

Lambaré G., Guillaume, P., Montel, J.-P. [2014] Recent advances on ray based tomography. *76th EAGE Conference & Exhibition*, Amsterdam.

LeCun, Y., Bengio, Y., Hinton, H. [2015] Deep learning. *Nature*, **521**, 436-444.

Mandelli, S., Lipari, V., Bestagini, P. [2019] Interpolation and Denoising of Seismic Data using Convolutional Neural Networks. *arXiv:1901.07927v4*.

Ovcharenko, O., Kazei, V., Peter, D., Alkhalifah, T. [2019] Transfer Learning For Low Frequency Extrapolation From Shot Gathers For FWI Applications. *81th EAGE Conference & Exhibition*, London.

Richardson, A., Feller, C. [2019] Seismic data denoising and deblending using DL, *arXiv:1907.01497*.

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

Sen, S., Kainkaryam, S., Ong, C., Sharma, A. [2019] Augmented Adversarial Training: Improving Robustness of NN Based Geologic Interpretation. *81th EAGE Conference & Exhibition*, London.

Siliqi, R., Le Meur, D., Gamar, F., Smith, L., Touré, J.-P. [2003] High-density moveout parameter fields V&η, Part1:Simultaneous automatic picking. *SEG Technical Program Expanded Abstracts*, 2088-2091.

Sun, H., Demanet, L. [2018] Low frequency extrapolation with deep learning. *SEG Technical Program Expanded Abstracts*, 2011-2015.

Yuan, P., Hu, W., Wu, X., Chen, J., Nguyen, H. [2019] First arrival picking using U-net with Lovasz loss and nearest point picking method. *SEG Technical Program Expanded Abstracts*, 2624-2628.