Combining Seismic Processing with Machine Learning

Daniel Trad
University of Calgary – CREWES

Summary

Machine learning (ML) is a powerful tool that has become very useful in many areas of science and seismic applications are not an exception. For example, the uses of ML in interpretation have proven to be very helpful. Also, in seismic processing, we see a significant effort for using ML techniques to help or replace traditional processing. Processing applications, however, present a more difficult challenge. In this abstract, I will discuss what some of those challenges are and examples of hybrid dataflows where ML does not replace but rather cooperates with traditional signal processing and inversion. I will present this methodology with several examples: migration, multiple attenuation, near-surface noise suppression and interpolation. To illustrate the flexibility of machine learning and the importance of data preprocessing, I will address all these problems with the same deep learning network but with different inputs and outputs.

Seismic and ML Workflows

Practical experience developing processing tools for seismic data shows that typically it is more difficult to insert new tools within an existing dataflow than to create new tools. The processing sequence is an iterative dataflow, where information is approximated and improved at each step. Different multidimensional algorithms work back-to-back in a dataflow with changing domains. Errors in the outputs of one module affect the results of the following module. Some of these modules are computationally expensive and require dividing data into multidimensional windows. Overlapping of these windows improves the certainty of the outcomes.
We are on the verge of a different and powerful technology that promises to be more flexible than conventional approaches. While physics-based methods require us to write the specific rules that nature follows, ML approaches have the potential to extract and implement those rules directly from the data (Chollet, 2021). While in the first case, we use physics to constrain the range of possible outputs, in ML we allow every possible output to occur at first, but prune them by training, eliminating in this way non-physical possibilities. The ML approach, pruning by training, is highly dependent on the existence of abundant amounts of data in quantities never required before. Therefore, more than ever, we see that the abundance of training data is the enabler and limiting factor on what can be achieved. The problem, therefore, resides in inserting this new technology into a classical dataflow. Much of the published work on deep learning (DL) tries to use an "end-to-end" machine learning dataflow, rather than an "end-to-end" hybrid dataflow which may be more appropriate for seismic.

In a seismic processing dataflow, each step is closely related to the steps before and after. Information is continuously extracted from the data, through the different algorithms, and inserted back into the dataflow to act as prior information for the following modules. Processing has to be consistent with reality, forcing us to add geologic constraints which discard unrealistic results. Different processes work on different domains, and therefore data need to be sorted or divided into subsections. Noise from acquired data permeates across modules and requires filters. All these processes are computationally expensive and typically require high-performance computing techniques. In modern times, many of them are implemented with Graphic Processes Units (GPUs), enabling processes that were infeasible before because of computational cost.

There are many reasons why these flows are complex, but we can mention the most obvious two: a) seismic data are multidimensional (5 dimensions). b) These dimensions are irregularly and poorly sampled.
On the other hand, processing with neural networks contains four stages:

- Preprocessing: data are converted into a format that can be read in later stages, reformatting them into regular shapes, and applying scales and normalizations. For supervised training, labels have to be assigned if not available.
- Learning: data are fed into neural networks to predict outcomes that match the labels. Predictions errors are used to recalculate the neural network weights until the prediction error (residuals) are minimized enough.
- Evaluation: models are tested and results are interpreted in terms of the network.
- Prediction: results are calculated for new data and monitored to make sure the network does not deteriorate with time.

A large part of the ML dataflow is data preparation, with a strong similarity with seismic, where migration of data requires significant preparation work (the seismic processing flow).

A common algorithm of ML is deep learning (DL) which is used in computer vision. For example, autoencoders and similar network patterns are essentially compression algorithms. By reducing information to a minimum number of degrees of freedom and mapping the data to a lower dimensional manifold (Chollet, 2021), these networks can reduce noise, improve the resolution or detect features or shapes in an image. The input to these algorithms resembles the format of a migrated seismic section, that is regular grids in space, where every pixel represents a numerical encoding of a property. Since seismic interpretation is typically done on migrated sections, it is easy to see why computer vision algorithms are easy to use for seismic interpretation. For example, salt bodies, facies and faults identification, all can be done with semantic segmentation algorithms by assigning a probability to each pixel that it belongs to a particular target feature. However, when we go from interpretation to processing, this similarity breaks. The world of computer vision typically is represented by multidimensional matrices, grouped in channels, each channel representing a property, for example, Red, Blue and Green intensities (RBG). Neural networks are very efficient in handling regular volumes like these by using a convolutional pattern algorithm for which GPUs excel at. The world of seismic processing, on the other hand, is formed by irregularly sampled groups of seismic traces with 5 dimensions, typically separated into subsets with common physical properties. These subsets contain large sampling gaps with irregular distributions, and these gaps change very rapidly from location to location or different acquisitions. The continuity of the seismic events can only be accounted for when true irregular sampling is taken into account. Minor variations on the irregularity, for example, introduced by binning, produce a major deterioration of high frequencies when algorithms stack across coherent events. In essence, patterns are seriously affected by sampling. These complications have led geophysicists to create complex dataflows, where data are manipulated on different types of groups from beginning to end. Sorting on these groups is key for the functionality of the processing modules, so data re-arrangement is carefully planned ahead of time with serious consequences for efficiency. In addition, algorithms need to be able to account for irregular sampling and only then, patterns or coherence can be detected and applied for further developments.
This discussion hints at the problem that processing images is not the same as processing seismic data so applications of ML to processing require a hybrid classical/ML combination. Research in this area typically uses classical processing to create datasets and then move them into ML where

different techniques are attempted. In practice, this approach is broken and makes research difficult because there is a disconnection between the two stages. A more useful approach is to run both stages simultaneously in one dataflow, in a hybrid setup to try different combinations of inputs and labels and the same time that different network designs. This hybrid flow shows that solving processing problems with DL seems to be much more dependent on the data processing than the network architecture itself. It is possible to solve a wide variety range of problems using the same DL architecture but varying the processing part. In this abstract, I show this by addressing the following four problems: Least Squares Reverse Time Migration, multiple attenuation by Radon transform, Ground Roll attenuation, and interpolation of shot gathers, in all cases using the same network: a classical U-Net (Ronneberger et al., 2015).

Examples

Example 1 - Image-domain Least-Squares Reverse Time Migration: LSRTM in the image space (Yu et al., 2006) takes a regular RTM image as the input and approximately deconvolves the Hessian by using filters. This can be done by training a network with regular RTMs as inputs and LSRTM results as labels (Torres and Sacchi, 2022), or a theoretical reflectivity from synthetic velocity models (Huang and Trad, 2022). For the goal of this abstract, we will attempt this goal by using a U-Net although many other networks are possible. In addition to the network architecture, there are still many choices to make: what do we use for input (channels) and outputs (labels)? This has to be partially decided by intuition and partly by experimentation. A first step is usually the choice for the labels (what we want). In this case is the band-limited reflectivity. The input problem requires to consider what kind of information the network needs to produce that output. This information is set in the form of channels. For the first channel, it makes sense to use the RTM, since this will be the information available for real problems. As a second channel the smooth velocity we used for the migration and a third channel with the reflectivities from the smooth velocity model used during RTM. The setup and result from training can be seen in Figure 1a. In Figure 1b appears a prediction obtained for a completely different model where we have no labels. The enhanced migration shows a more detailed structure, suggesting that the information captured from the training with the Marmousi model generalizes to a different geology.

Example 2: Multiple attenuation by Hyperbolic and Parabolic Radon transforms: let us use the same network but change the inputs and labels to address a completely different challenge: the separation of primaries and multiples by Radon transforms. In Figure 2 we see one shot gather extracted from a data set of 60 shots with 380 receivers each. This data set was calculated from a synthetic model, where we can create primaries and multiples separately and add them to create input data and labels. Two possible approaches are 1) train a network with the input data and labels (primaries or multiples alone). 2) convert the data to a different domain and input the transforms of the data and the transform of the labels. Both are possible but not equal in complexity. Working with transforms has the additional advantage of regularizing the input to the network for irregular sampling, which is very difficult to deal with for convolutional neural networks since the network cross-correlations assume samples are equally spaced in all dimensions.

A good transform to separate primaries and multiples is the Radon transform. Rather than try to teach the network how to calculate a Radon transform, we can provide the network with the transforms which we already know how to calculate efficiently. Radon transforms are calculated in the CMP domain because they require events to have their apexes at zero offsets, a condition approximately satisfied for common midpoints (CMPs) but not for shot or receiver gathers. The sampling on the CMP domain is usually coarser than on the shot and receiver domains unless

the geometry is perfectly symmetric in terms of shot and receiver intervals. Here I have chosen for illustration a typical case where shots are far apart, producing much better shots than CMPs.
Figure 2 shows the hybrid dataflow. The beginning and end of the flow involve efficient classical calculations using conventional signal processing techniques. The process is an end-to-end “hybrid” dataflow, which makes it easier to experiment by changing simultaneously both types of parameters: inputs and outputs to the network and the network itself. It is interesting to note that 1) compared to the previous example, the network is the same, but the inputs and outputs are different. 2) the computational cost of training the U-Net in this example was almost one order of magnitude larger than the conventional part of the dataflow (9 minutes versus 1 minute for 380 CMPs). Therefore, the training effort should be always taken into account, although a good generalization power from the network would require training only once.
In this example, these transforms suffer from many artifacts and noise, mostly on the left and right of the panels. This noise is a result of poor sampling and limited aperture on the CMPs. An approach to address this is the *sparse* Radon transform (Thorson and Claerbout, 1985). In this dataflow, changing between these two variants of RT amounts to changing just one parameter. This is an example of how a hybrid classical ML dataflow becomes essential for experimentation. Understanding the behaviour of the network when modifying the transform requires testing.
Still, the final output of the dataflow, i.e. the shot gathers, shows significant noise and artifacts. Poor CMP sampling can be fixed by adding input shots. Taking advantage of the end-to-end hybrid dataflow, we can corroborate this by increasing the number of shots during modelling (changing one parameter in the dataflow). The numerical experiments show which factors we should take into account. Many other tests can be performed, for example, replacing the HRT with a parabolic RT (PRT) or combining them both as inputs for the network. Since the PRT and the HRT use different curvature parameters, combining both is not straightforward, but it can be done by using an additional network to map one to the other. Once proper mapping is achieved, it is possible to incorporate both of them into the same U-Net by using different channels.

Example 3: Eliminating Ground Roll: Here the goal is to train a network to remove ground roll. Just as in the previous case, it is possible to work directly with seismic gathers or apply some kind of transformation. Figure 3 shows a) simulated data (input), b) reflections only (labels), and c) we see the predictions (reflections). Inputs and labels are obtained by elastic finite-difference modelling from topography (Sanchez et al., 2022). In this case, the data and labels were the Hybrid Radon transforms (linear+parabolic) of the seismic gathers (Trad et al. 2000).

Example 4: Interpolation: The last example involves the interpolation of seismic traces. For simplicity, we will consider only 2D interpolation. This example is an illustration of the flexibility of DL, but multidimensional interpolation across all domains using coherence methods like Fourier, Low rank, or Radon transforms in 5 dimensions works better than this ML approach. The extension of ML to 5D interpolation would be computationally more expensive than conventional techniques. Here we will just explore the generalization capability for this problem. Using a data set to train a network and then predicting missing traces contained on the training labels is guaranteed to work well because of the large number of parameters that neural networks use. The network can memorize every missing sample from the labels. What is difficult is to train the network on a set of data sets and apply it to interpolate on a completely different dataset, that is train a network with generalization power. A way to improve on generalization is to train the network with many different data sets. The more diverse the training data, and the more abundant, the better the generalization power. The ideal approach would be to continue the training each time a new data set (or computer resources) are available. This approach can be challenging since retraining with new data sets tends to deteriorate the previously calculated weights. A way around this could be freezing the layers closer to the input and retraining the final layers as in transfer learning. That could make the network learn new structural patterns but reuse details like the nature of seismic data (amplitude and phase variations from trace to trace). In this example, all the datasets were trained together, but that makes the problem more dependent on I/O constraints but it improved generalization. Each of the 4 training datasets contained 20 shots and 400 receivers each, with ten random gaps along receivers on each shot gather. The gap sizes were variable between 40 and 160 meters. The full datasets (before introducing the gaps) were used as the training labels. Figure 4 left-hand side, shows the training results for two of the 4 datasets. On the right, we see the prediction for the testing data set. A comparison with a physics-based interpolation using Fourier showed both methods have a similar quality, but the physics-based approach does not require the training. A fairer comparison would have been with a full multidimensional approach (3D in this case since data is only 2D).

Conclusions

This abstract illustrates a methodology that can be applied to different problems by integrating machine learning tools into conventional dataflows for rapid testing. Four different seismic problems are addressed by using a hybrid conventional+ML dataflow. Increasing the complexity of the network is not the best solution to address these types of problems. A better approach is to experiment with the dataflow by trying different inputs and labels. A combined dataflow that permits simultaneous experimentation with both the ML and the conventional processing brings a significant speed up to the process. The examples illustrate the flexibility of neural networks to change their functionality completely without architectural changes. The training process is sufficient to change the filters and obtain different results. A major challenge in all problems involving seismic data processing and migration is to achieve a good generalization power, such that data training does not need to be repeated for new types of data sets. Transforms are the key to eliminating sensitivity to the geometry, source signatures and geological complexity.

Acknowledgements

Thanks to Sam Gray for many fruitful discussions, CREWES sponsors, and NSERC for support through the grants CRDPJ 461179-13, CRDPJ 543578-19 and NSERC Discovery Grant.

References

Chollet, F., 2021, Deep Learning with Python: Simon and Schuster.

Huang, S., and Trad, D. O., 2022, Convolutional neural network-based reverse time migration with multiple energy, *in* GeoConvention, Conference Abstracts.

Sanchez, I., Agudelo, W. M., Sierra, D., and Trad, D. O., 2022, Simulated removal of near-surface scattered waves by elastic wave modelling, *in* GeoConvention, Conference Abstracts.

Thorson, J. R., and Claerbout, J. F., 1985, Velocity-stack and slant-stack stochastic inversion: Geophysics, **50**, No. 12, 2727–2741.

Torres, K., and Sacchi, M., 2022, Least-squares reverse time migration via deep learning-based updating operators: Geophysics, **87**, No. 6, S315–S333.

Trad, D., Sacchi, M. & Ulrych, T. (2000). A Hybrid Linear-Hyperbolic Radon transform. *Journal of Seismic Exploration*, *9*, 303-318.

Yu, J., Hu, J., Schuster, G. T., and Estill, R., 2006, Prestack migration deconvolution: Geophysics, **71**, No. 2, S53–S62.

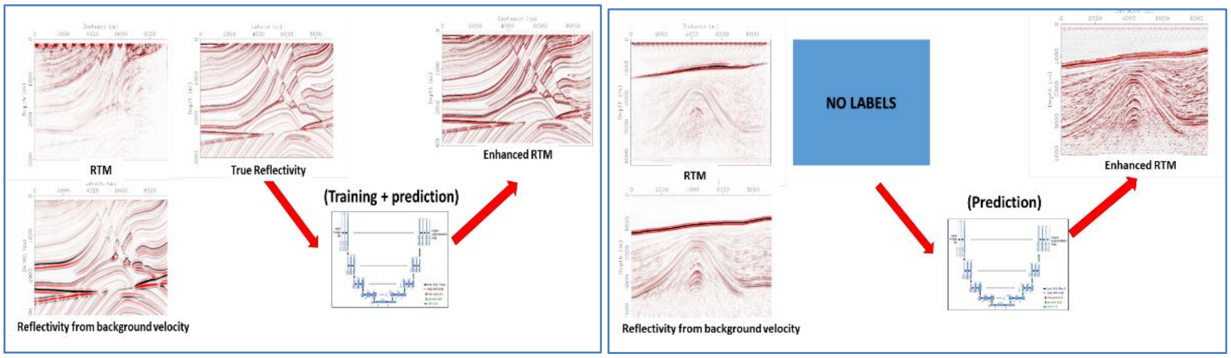

Figure 1 - Image-domain LSRTM by using a U-NET. Different types of inputs are combined to predict a more resolved RTM. Left training, right, application. Channels can contain different attributes, like illumination, derivatives of velocity, and others.

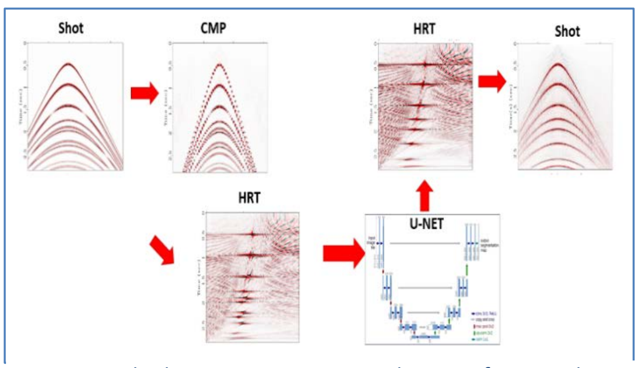

Figure 2- Multiple attenuation using Radon transforms and ML. The network can learn the filters in different complex situations. Different transforms can be combined in different channels.

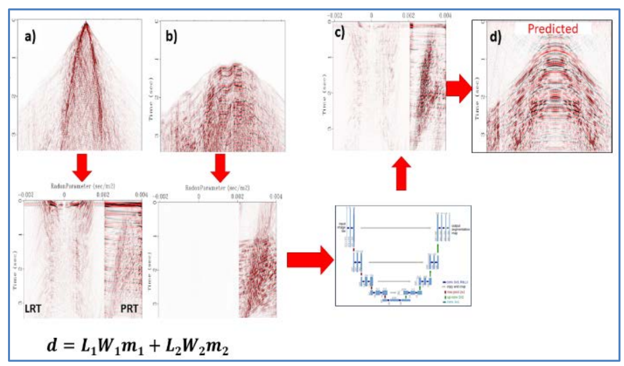

Figure 3- Ground Roll attenuation dataflow using U-NET. Data and signal can be mapped using different transforms. a)data,b)signal)c)HybridRT predicted,d)Predicted data

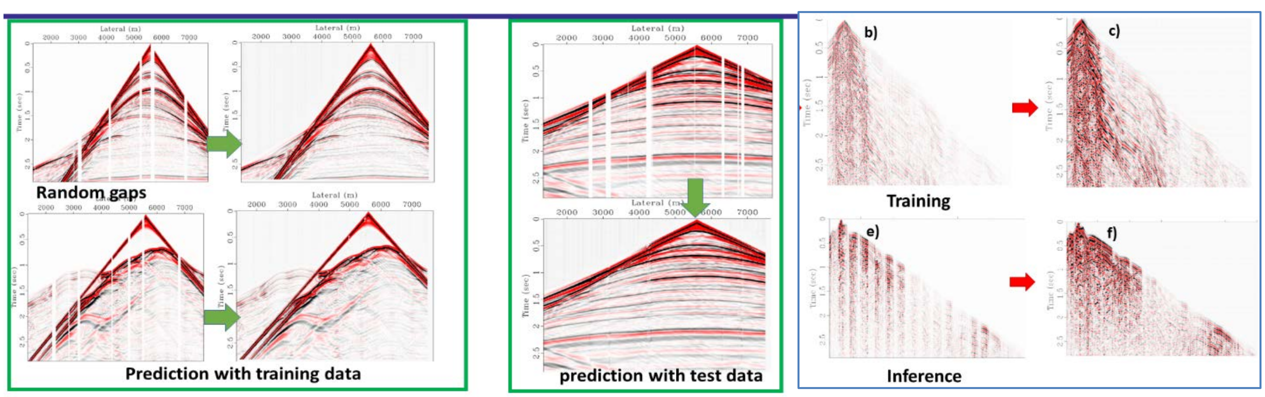

Figure 4- Interpolation using U-NET- Left, training with four different datasets.Middle, prediction on a new dataset. Right, Interpolation for Ground Roll during training and inference. The main challenge is to achieve generalization through training different datasets.