A seismic signal processing framework using machine learning on an IoT devices for in the field pre-processing

Marcus Landschulze[1]

[1]Western Norway University of Applied Sciences

November 21, 2022

Abstract

Using machine learning in geophysics is often considered as a fast approach to process or interpret seismic data, but the challenge is to get enough data to train the machine learning core. This framework uses a combination of real noise data and synthetic reflection seismograms generated from e.g. real source signal or band-limited pulses for training the machine learning core. The trained core can be stored on (IoT) devices which can be used in the field to preprocess the data, for e.g. QC, before sending it to the office for further processing. This will decrease the turn-around time and will help geophysicists to decide whether the data is useful for further processing or needs to be re-collected. I will explain the framework, discuss the results, and show how the framework improves the seismic data quality. The framework can deconvolve the seismic data to zero-phase band-limited pulses with simultaneous noise reduction.

A seismic signal processing framework using machine learning on an IoT devices for in the field pre-processing

M. Landschulze[1]

[1] Department of Computer science, Electrical engineering and Mathematical sciences, Western Norway University of Applied Sciences (HVL), Bergen, 5020 Norway.

Marcus Landschulze (marcus.landschulze@hvl.no)

Key Points:

- Seismic signal processing
- Machine learning
- IoT in geophysics
- framework

Abstract

Using machine learning in geophysics is often considered as a fast approach to process or interpret seismic data, but the challenge is to get enough data to train the machine learning core. This framework uses a combination of noise data collected from the field and synthetic reflection seismograms generated from e.g. real source signal or band-limited pulses for training the machine learning core. The trained core can be stored on Internet of Things (IoT) devices which can be used in the field to preprocess the data, for e.g. QC, before sending it to the office for further processing. This will decrease the turn-around time and will help field engineers and geophysicists to decide whether the data is useful for further processing or needs to be re-collected. I will explain the framework, discuss the results, and show how the framework improves the seismic data quality. The framework can deconvolve the seismic data to zero-phase band-limited pulses with simultaneous noise reduction.

1 Introduction

Using machine learning for seismic interpretation or inversion (Zheng, Zhang, Yusifov, & Shi, 2019) is a growing area in geophysics which shows great potential for the oil & gas industry. However, using machine learning for seismic signal processing is still a niche and the focus is mostly on denoising. The main reason for that is the huge amount of training data needed to acquire reliable results, data which are often not accessible. There are approaches using a mixture of conventional denoising methods and machine learning to overcome this problem (Li, Zhang, & Mosher, 2019), so-called hybrid frameworks. These frameworks usually consist of two steps, where the first step uses conventional denoising, such as f-x filtering, as an initial estimate. The second step uses machine learning, like a dictionary learning approach, together with signal inversion to recover the desired signals.

I present a full machine learning framework which can deconvolve the seismic data to a zero-phase band-limited pulse with simultaneous denoising, in one step. The fast and efficient framework is well-suited for and optimized to run on an IoT device to increase flexibility but is not limited to IoT.

2 Framework

The signal response of subsurface geological layers can be described with seismic reflection coefficients. The reflection coefficient R is the ratio of seismic impedance contrast (ρV) at a boundary between two different velocity layers and can be expressed as:

$$R = \frac{\rho_2 V_2 - \rho_1 V_1}{\rho_2 V_2 + \rho_1 V_1}, \qquad (1)$$

with the density ρ_2 and the seismic wave velocity V_2 of the medium below a reflecting interface and density ρ_1 and seismic wave velocity V_1 of the medium above the same interface. The reflection coefficient ranges from -1 to 1, where one means full signal reflection and -1 full negative reflection (phase inversion).
One seismic trace can contain several reflection coefficients, depending on the assumed velocity model. Due to low seismic signal frequency (here 45 Hz) and the resulting resolution, the seismic

resolution for layer thickness is some tens of meters and only those thick layers generate measurable reflections.

Due to its describing nature, I used the reflection coefficient to label each seismic trace, as shown in the framework sketch fig. 1.
The framework to train the ML core uses real noise from the field area combined with synthetic reflections coefficients generated by using measured velocity and density and convolved with the source signal.
In order to get synthetic seismic traces as training input, I generated reflection coefficient traces using real velocity and density data from well measurements (fig. 1b). These traces are convolved with the source signature and real noise from the same receivers, but without source signals (fig. 1c). To avoid training on noise patterns the noise in the training data must be different on each trace. I used a 1-hour-long record of noise data and randomly selected a part of it with the same time length as each data trace before adding the noise to the convolved trace.
The framework is not limited to any number of samples, but the number of samples defines the size of the input layer.

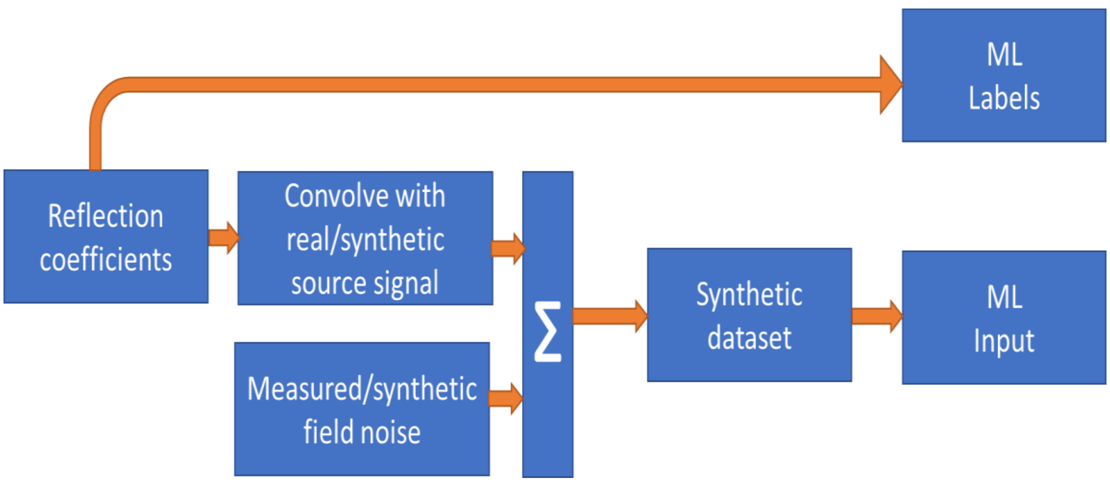


Fig. 1 Framework sketch

The framework focuses on a fast and efficient machine learning (ML) approach and uses multi-layer perceptron (MLP) as ML architecture (HassanAitLaasri, Akhouayri, Agliz, & Atmani, 2013; van der Baan & Jutten, 2000). The MLPs are optimized for different IoT hardware processing units such as NVIDIA Jetson Nano with Maxwell GPU or Google's tensor processing unit (TPU).
In order to simplify the topology, but to fulfil the accuracy, a 5-layer architecture is used in such a way that the number of perceptrons in the first input layer is equal to the number of samples in a trace. But each of the following 3 hidden layers have less perceptrons compared to the input layer, only a fifth. The final output layer has again the same number of perceptrons as samples (Fig. 2).

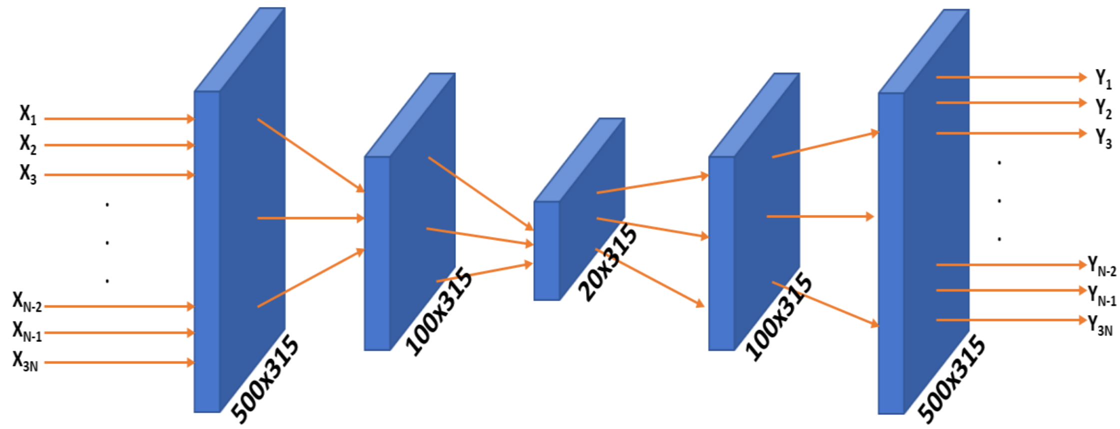


Fig. 2 ML topology with 3 hidden layers.

This architecture forces the machine learning core to "forget" information (noise) during training, since there are less perceptrons to train the hidden layers than samples in a trace. The loss function is defined as mean-squared-error with linear activation functions. The optimizer used is the RMSprop which is similar to the gradient descent algorithm but converges much faster.
This approach assumes that effects of dispersion or anisotropy on the traveling seismic signal are neglectable, which means that the frequency band and the phase of the signals remain constant.
Furthermore, to train noise the architecture needs more perceptrons than the number of samples to get reliable estimations. Hence, reducing the number of perceptrons means losing noise.

3 Use case and results

I tested the trained ML core on a PC and on a Jetson Nano with synthetic and field data[1], but show here only the synthetic data example.
The used source receiver configuration is a common setup for well measurements. In this configuration, the seismic source is located on the surface, usually a vibrator or an airgun, in a water pit for land surveys, or an airgun-cluster offshore. The receivers are installed inside the wellbore. The seismic signal travels through the subsurface down to the receivers decreasing in amplitude due to geometrical spreading and reflections back to the surface. The amplitude at 3000 m depth contains usually less than 5% of the original amplitude and the signal-noise-ratio is often below 1.
Figure 3 shows an example of synthetic field data with 315 well receiver locations (Fig. 3a), the assumed velocity model (Fig. 3b) and real noise measured on the surface (Fig. 3c). The source signal was a Ricker pulse in the velocity domain (derivative) with a corner frequency of 45 Hz. I used a trace length of 1 second and a sample rate of 500 Hz. The resulting size of the input layer is 500 (Fig. 2).

[1] not permitted to show the field data due to property right of the oil company

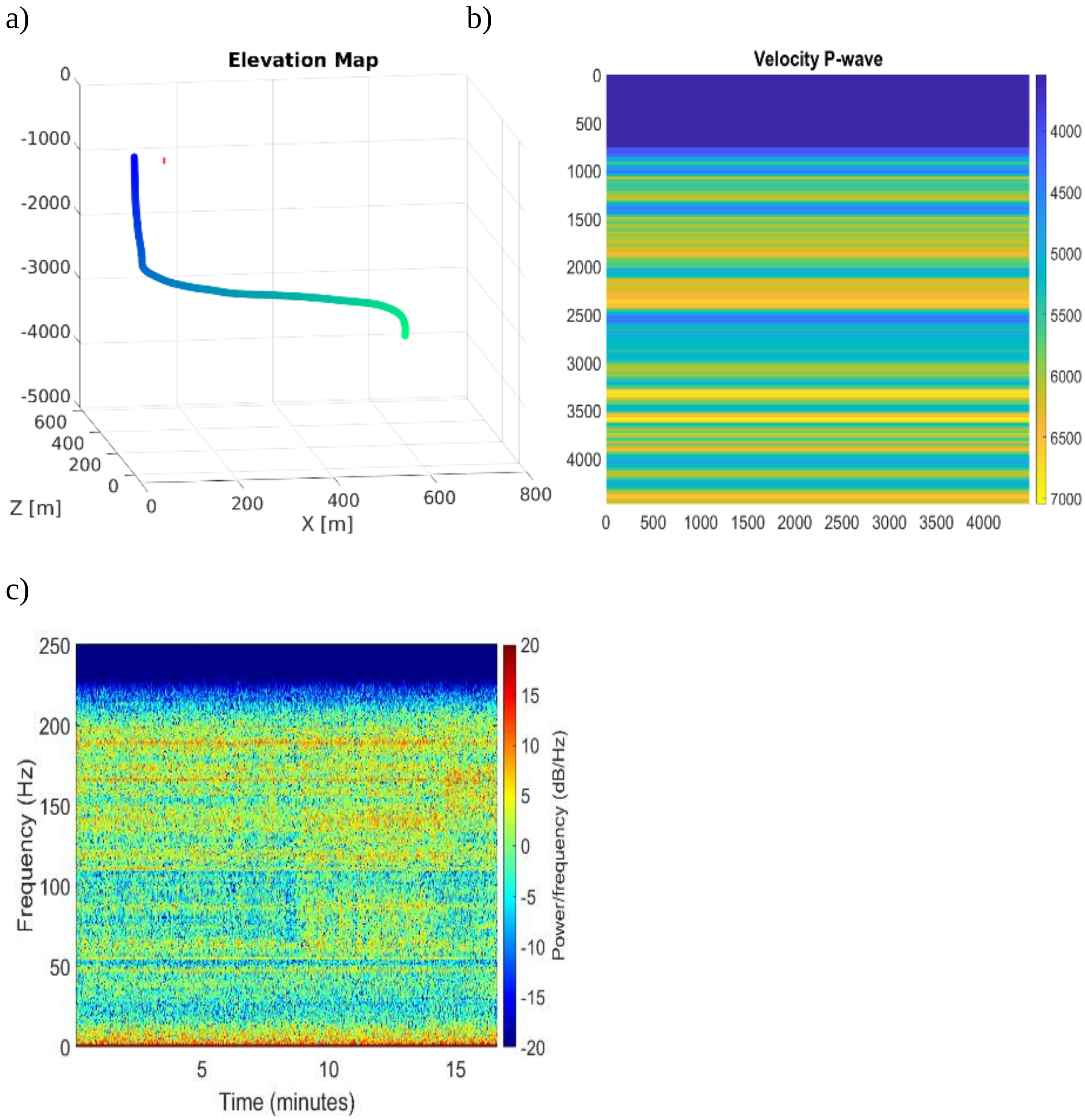


Fig. 3 shows the parameters for the use case. 1a) well elevation map with the source at the location of the red star, 1b) 2D velocity model and 1c) real noise measured from the surface.

The training input of the ML core is a sequence of 500 synthetic traces (blue) and the reflection coefficient labels (orange) of the sequence are the reflectivity coefficients (Fig. 4). The trained output is shown in green.

80,000 traces with up to 8 reflection coefficients per trace are sufficient for training. Generating the synthetic dataset takes approx. 5 min on a standard PC. The training of the ML core with the 80,000 synthetic traces takes approx. 6 min using GPU, and 20 min using CPU.

After training, the training-accuracy should be between 0.6 and 0.8. In my experience, a higher training-accuracy may imply noise estimation.

The resulting trained ML is stored in a file and may be uploaded to the designated IoT device or used on the PC.

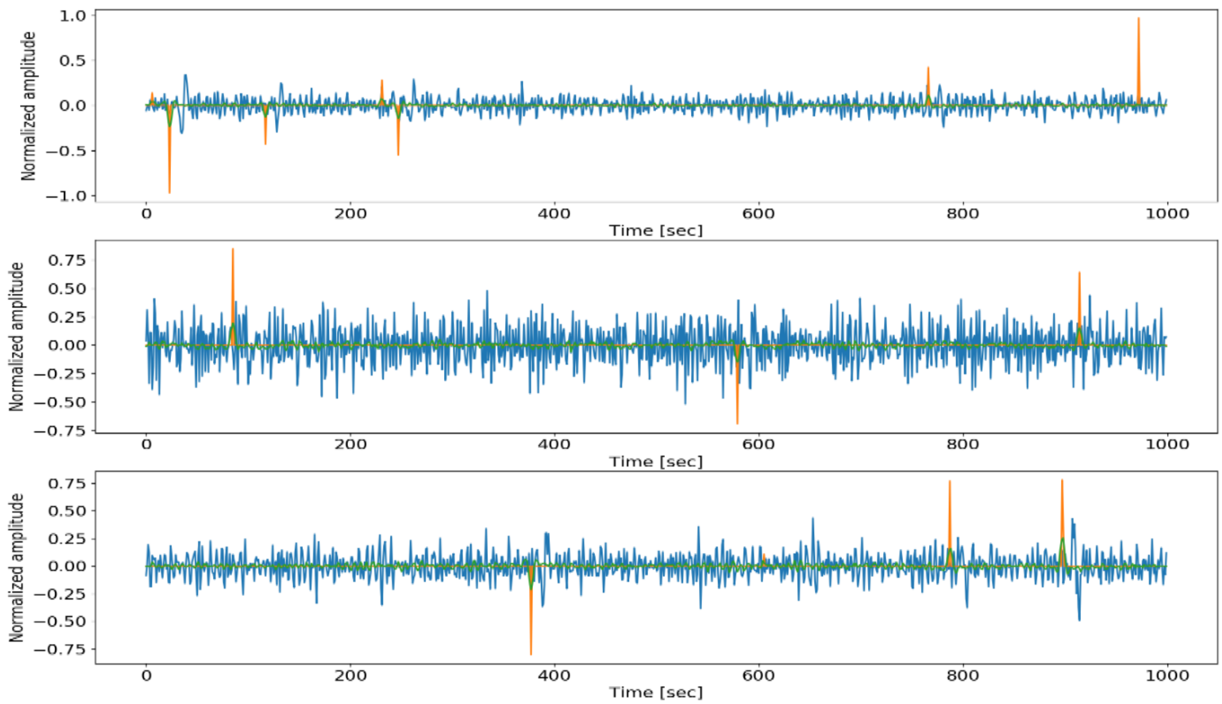


Fig. 4 shows three examples of 1-second-long test data for building the ML core. The spikes on the orange curve represent the reflections coefficients. The blue line represents the geophone response for a near phase Ricker-pulse in the velocity domain, and the green curve is the result of the machine learning on a band-limited zero-phase pulse. The training-accuracy of these examples was 73%.

In order to build a more realistic synthetic dataset I used the receiver array geometry (Fig. 3a) and the kernel matrix $\boldsymbol{A}$ as described in (Song & Toksöz, 2011).

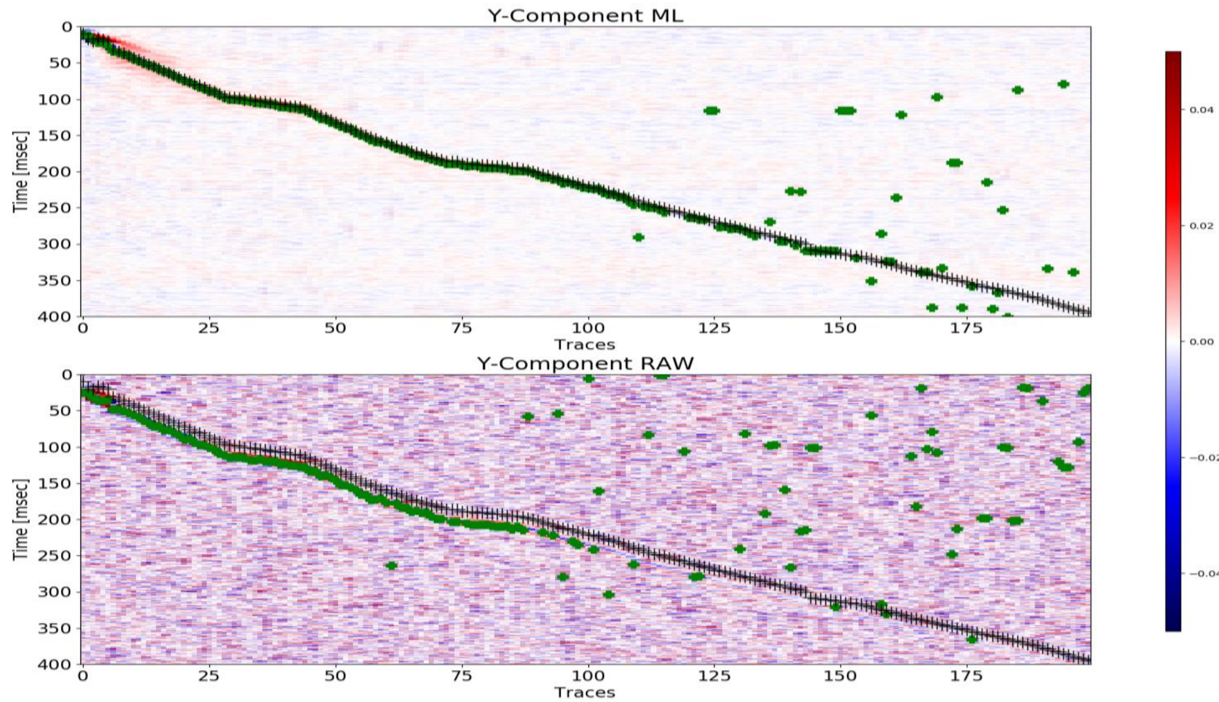
$$A_{ik}(x_s) = \sum_{n=1}^{N} \sum_{k=1}^{3} \sum_{0}^{Nt} g_{kj}(x_r^n, x_s, t) \cdot g_{ki}(x_r^n, x_s, t)\,, \qquad (2)$$

where g_{ki}, g_{kj} represents the Green's function, $\boldsymbol{x_r}$ and $\boldsymbol{x_s}$ the receiver and source position, t the time, N the number of receivers and k the number of receiver components. The real field noise, which was used for training, was added to the synthetic dataset in a different, arbitrary combination of noise clips to the synthetic dataset. These two different approaches (synthetic data generated by reflection coefficients and Green's functions) guarantee independency between the synthetic dataset used for training (reflection coefficients) and the synthetic dataset used for prediction (Green's functions). Furthermore, the approach used for the prediction dataset changes the reflections signal shape depending on the distance and incidence angle for P- and S-waves.

Figure 5 shows the datasets used for prediction. 5a and 5b show the results after prediction using the ML in the upper plot and the original noisy datasets from two receiver components (one horizontal and one vertical) in the lower plot. The black crosses indicate the real position of the seismic reflections, the green dots the automatically picked reflection positions and the red-blue color-coded image the amplitudes of the trace signals and noise.

The predicted results show a much closer match to the real seismic reflections as shown for example in Figure 5a. The ML core reduced the noise floor yielding more accurate automatically picked reflections on the predicted data.

a)



b)

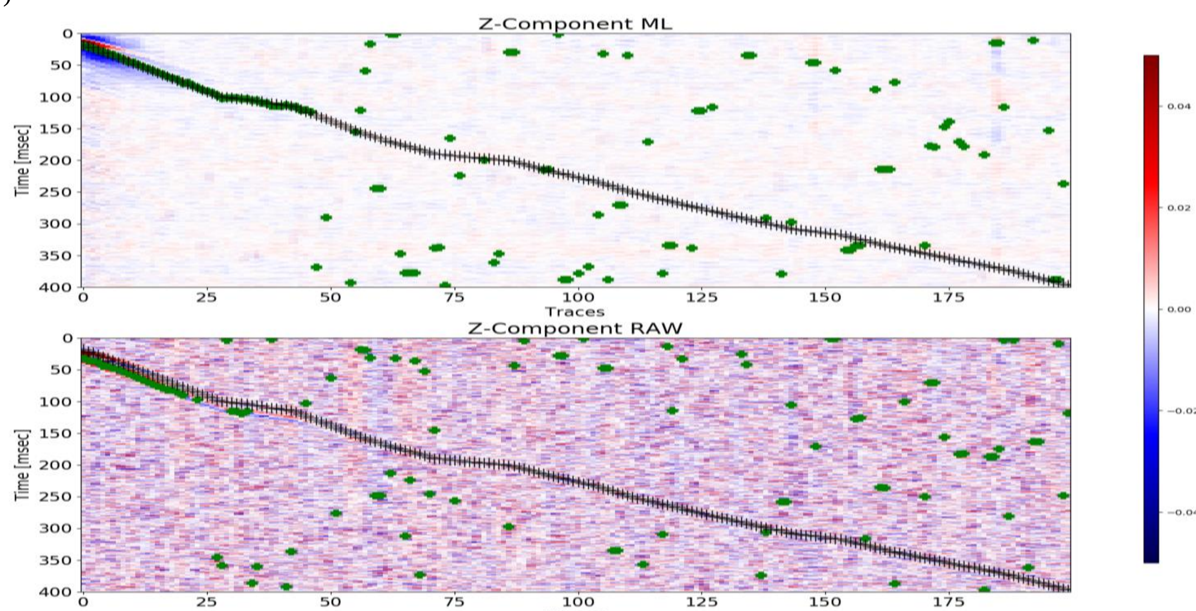


Fig. 5 The prediction output (upper plots) for the trained ML core for a) a horizontal component and b) the vertical component, and the corresponding synthetic input dataset (lower plot). Black crosses indicate the real position of the seismic reflections, the green dots the automatically picked reflections. Traces are the number of receivers in the well and the time axes shows one-way travel time.

The first ca. 20 traces have some strong artefacts over the first 50 msec which are possibly related to the high source energy near the surface. Those artefacts are above the area of interest for the use case industry, which is usually close the reservoir and deeper than 50 msec.

In order to illustrate how the signal phase in the dataset changes from near-phase to zero-phase Figure 6 shows three single traces at different receiver depths. The orange line represents the zero-phase reflection coefficient, the blue line the convolved near-phase receiver response and the green the prediction from the ML core.

With increasing the depth, the amplitude decays and the noise becomes more dominant. Figure 6 shows that the noise is reduced, and, in addition, the convolved amplitude (blue) is deconvolved to a nearly zero offset bandpass-limited signal (green). Figure 6b you can see two reflection coefficients (orange) in close proximity. Due to the seismic signal resolution these two coefficients cannot be separated and appear as one signal response. The value of the reflection coefficient in Figure 6c is below the seismic noise (blue) and the convolved seismic signal is not distinguishable from noise. The ML result reduces the noise, but the seismic reflection (green) is not clearly visible and can be misinterpreted.

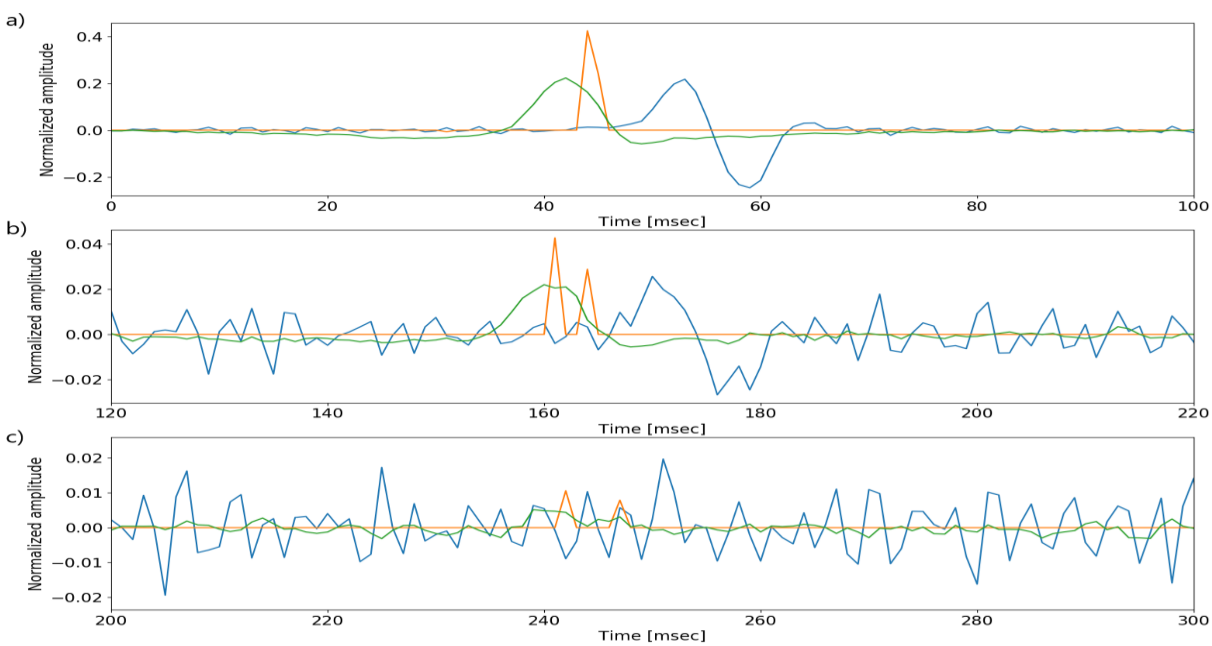


Fig. 6 three 0.1 second long "zoomed in" examples where the spikes on the orange curve represents the reflections coefficients, the blue the geophone response with near phase Ricker-pulse in the velocity domain and the green curve the result of the machine learning as a bandlimited zero-phase pulse.

5 Conclusions

The presented framework proposes a method to train a machine learning core to recover seismic signals. Using synthetic reflection coefficients convolved with the source signature and adding real noise provides a fast method to generate tens of thousands of traces in a short time to train the ML core. In addition, using reflection coefficients as labels predicts zero-phase band-limited seismic signals.

A reduction of perceptrons in the hidden layers helps the ML core to "forget" the noise and to preserve the reflection signals.

Copying the trained machine learning core to an IoT device makes the data processing independent of an office and can be done in the field as a pre-processing stage for data QC or to reduce the turn-around time. The processing time on the IoT device for 315 traces with 1sec data length is less than 2min in average which is suitable for QC in the field.

This framework is not limited to seismic data and can be used for any type of time-series data where the source signature and the noise are known.

Acknowledgments

Datasets for this research will be available at Pangaea (https://pangaea.de) as soon as I can upload the data. Data submissions to Pangaea will not be processed before 16.01.2021. The dataset is attached to this submission.

References

HassanAitLaasri, E., Akhouayri, E.-S., Agliz, D., & Atmani, A. (2013). Seismic Signal Classification using Multi-Layer Perceptron Neural Network. *International Journal of Computer Applications*, *79*(15), 35–43. https://doi.org/10.5120/13821-1950

Li, C., Zhang, Y., & Mosher, C. C. (2019). A hybrid learning-based framework for seismic denoising. *Leading Edge*, *38*(7), 542–549. https://doi.org/10.1190/tle38070542.1

Song, F., & Toksöz, M. N. (2011). Full-waveform based complete moment tensor inversion and source parameter estimation from downhole microseismic data for hydrofracture monitoring. *Geophysics*, *76*(6), WC103. https://doi.org/10.1190/geo2011-0027.1

van der Baan, M., & Jutten, C. (2000). Neural networks in geophysical applications. *GEOPHYSICS*, *65*(4), 1032–1047. https://doi.org/10.1190/1.1444797

Zheng, Y., Zhang, Q., Yusifov, A., & Shi, Y. (2019). Applications of supervised deep learning for seismic interpretation and inversion. *Leading Edge*, *38*(7), 526–533. https://doi.org/10.1190/tle38070526.1