Collection, Compression and Classification of Seismic Signals from the Solar System
NASA Spaceapps challenge

László Párkányi[1], **Márk Dudás**[2], and **T. Dániel Lipovics**[3]

[1]Department of Electrical Engineering (Student), Technical University of Budapest
[2]Faculty of Law (Student), Pázmány Péter Catholic University
[3]London Centre of Nanotechnology (Student), University College London

October 10, 2024

Contents

1 Introduction

Seismology is a vital part of earth-sciences – it revealed a lot about the inner structure of our own planet, and with the dawn of the space age, it can reveal just as much about other celestial bodies. In our project, we tackled the challenges of collecting, compressing and categorising seismic data.

The data itself is time against velocity type – a seismograph in essence is a very sensitive motion sensor, tuned to detect tremors in the ground. However, seismic events are relatively rare, which results in a low proportion of the collected data being actually scientifically relevant.

Furthermore, the environment is highly noisy as natural – but irrelevant – processes close to the sensor can easily overshadow events of interest. Transmitting the raw data across large distances is therefore generally not favourable.

2 Overview

Our solution consists of three major stages. Firstly, the event collection. We operate on an auto-correlation function basis – also known as a match filter. An auto-correlation function represents the similarity between two sets of data and is therefore suitable for finding all the seismic events, provided we can identify a few initial ones.

As the volume of collected data is still large after this, we employ an auto-encoder. An auto encoder is a neural network architecture, that has input and output channels corresponding to the dimensions of our data, but in the middle the channel number is drastically decreased. After training, we can cut it in half, have the 'encoder' on the device, and transmit only the latent state, which – in our case – is barely a percent of the original volume. The decoder can reconstruct the data on Earth.

And thirdly, we use a dense neural network architecture to classify our data. The auto-correlation method has false positives, which we identify as a data-class, and even beyond that, we were given three distinct types of events that we aim to classify, thereby reducing the work of the ground crew.

Finally it is important to note that in our proposition the data is stored on the device, which means that if the reconstruction is insufficient, scientists can still pull the original as necessary.

3 Auto-correlation filter

The idea of using auto-correlation to find similar events in a noisy signal coming from telecommunications. In any channel where the noise can be modeled as an Additive White Gaussian Noise (AWGN), the maximum likelihood detector can be realized with autocorrelation. This application is sometimes referred to as matched filtering, as the correlation function can be realized with a convolution with the flipped conjugated of the sample we're looking for.

In our code, this matched filter was derived from the training dataset, by choosing samples from the marked arrival for 7000 seconds, which we determined is roughly the time it takes for the activity to dampen below the noise floor. The chosen regions were then averaged, resulting in a 'mean sample' that has the characteristic of all provided samples combined. If we had geophysical expertise, this filter could be refined by the use of finer models, but still our algorithm has proven to be sufficient for the current task.

To decrease the computational requirements of running auto-correlation on many hours, we decimated both the input signal and the matched filter to $1/3^{rd}$ of the original sample while applying a low-pass filter with a cutoff frequency of 1 Hz. The original signal was sampled with a rate of 6.6 Hz which we found overkill, as the events were confined between frequencies 1 and 0.5 Hz and according to Nyquist's criterion the minimum sampling rate for any real signal is twice the maximum bandwidth of the signal. With band-pass filtering a decimation factor of 6 would be possible, but due to lack of time this was not experimented with. In the end a 70000 seconds long signal can be analyzed in 70 seconds on a modern x86 processor using less than 200MB of RAM, so it's possible to execute our algorithm in real-time.

The results of the algorithm can be seen on 1. The errors mostly come from false positives, but as they were small in numbers and we can compress them down further (see 4) they aren't a major hindrance in the practical application.

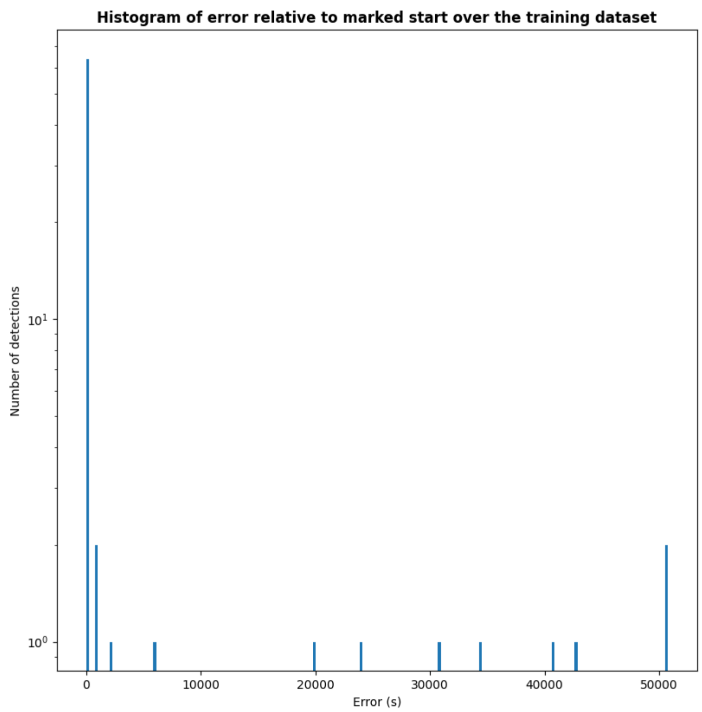

Figure 1: Results of the detection algorithm

4 Autoencoders

Autoencoders are a type of artificial neural network used primarily for unsupervised learning tasks, particularly in dimensionality reduction and feature extraction. They consist of two main components: the encoder, which compresses the input data into a lower-dimensional representation (or latent space), and the decoder, which reconstructs the original data from this compact representation. By training on the input data, autoencoders learn to capture the essential features and patterns, allowing for efficient data encoding without losing significant information.

In the context of transmitting data from the moon, autoencoders can be particularly useful for compressing large volumes of data, such as images or sensor readings, before transmission. The reduced dimensionality allows for faster data transfer, which is crucial given the limitations in bandwidth and latency associated with lunar communication. Additionally, by encoding the data into a more compact form, autoencoders can help mitigate the impact of noise and errors during transmission, ensuring that the most relevant information is preserved and reconstructed accurately at the receiving end. This makes them an effective tool for optimizing data transmission in remote space exploration scenarios.

Our autoencoder was capable of compressing from 46376 channels to a 128 channel latent space - and reconstructing the data with a reasonable fidelity. One could argue that moonquakes are inherently noisy and chaotic events, and therefore their high-level study is more meaningful than examining the low level noise - therefor the autoencoder's signal loss is not impacting our process significantly.

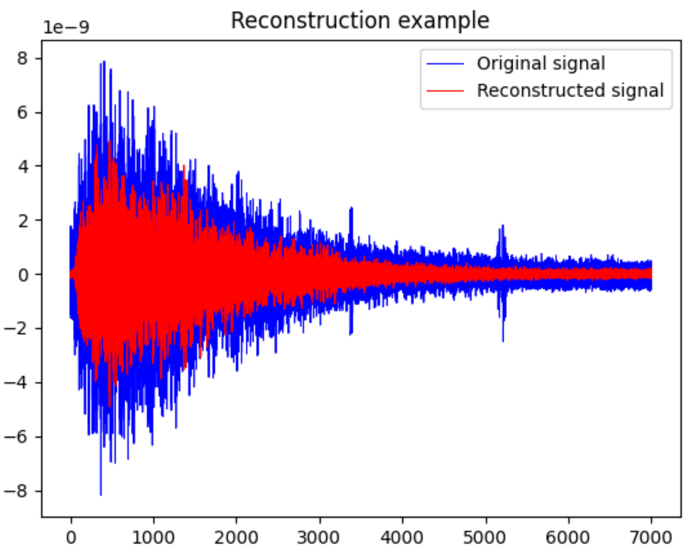

Figure 2: Reconstruction of the original signal with autoencoders

5 Classifier

Dense classifying neural networks, also known as fully connected neural networks, are a type of artificial neural network where each neuron in one layer is connected to every neuron in the subsequent layer. These networks consist of multiple layers, including an input layer, one or more hidden layers, and an output layer.

The input layer receives data, which is then transformed through the hidden layers using weighted connections and activation functions. Each neuron's output becomes the input for the next layer, allowing the network to learn complex patterns and representations. During training, the network adjusts its weights based on the error of its predictions using backpropagation and optimization algorithms, enabling it to minimize the difference between predicted and actual outcomes. One-hot encoding is often used to represent categorical target variables, where each class label is transformed into a binary vector, allowing the neural network to output probabilities for each class and making it suitable for multi-class classification tasks.

Dense classifying neural networks are commonly used in various applications, including image and speech recognition, natural language processing, and classification tasks, as they excel at capturing intricate relationships within data.

[t]0.3

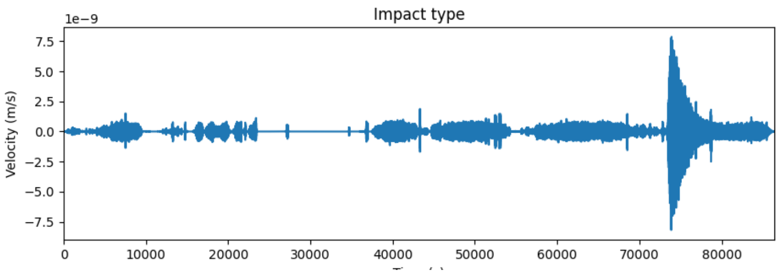

Figure 3: Impact type event

[t]0.3

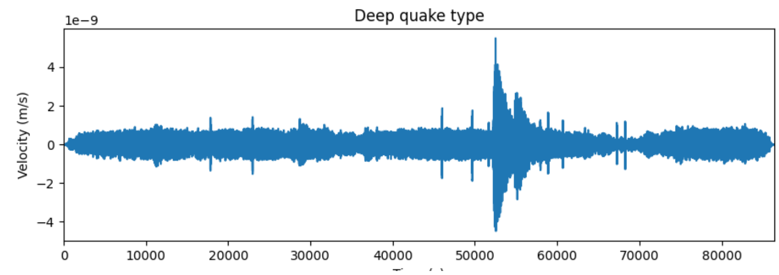

Figure 4: Deep-quake type event

[t]0.3

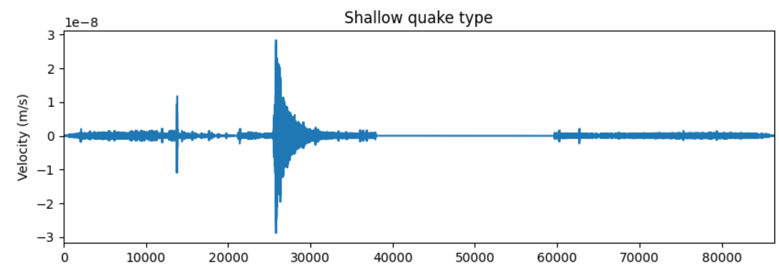

Figure 5: Shallow-quake type event

Figure 6: Different types of earthquake events

6 Summary

Our current challenges mainly consist in the lack of training data – with increased training set sizes, we could significantly improve the performance of both the autoencoder and the classifier. With multithreading we could improve the runtime of the autocorrelation method as well. In summary, our proposal collects, compresses and classifies – aimed specifically at streamlining data collection, saving both on the transmission costs and the time needed for data analysis.