Earthquake warning system: Detecting earthquake precursor signals using deep neural networks

Mustafa Al Ibrahim (malibrah@stanford.edu), Jihoon Park (jhpark3@stanford.edu), and Noah Athens (nathens@stanford.edu)

Introduction

In the earth sciences, earthquake prediction is one of the great unsolved problems. Despite decades of research attempting to identify earthquake precursors, there is no proven analytical method to predict earthquakes before they occur. The problem is challenging for several reasons: (1) any precursor signal that may exist is likely dwarfed by the background seismic noise; (2) seismic data is multidimensional recording different directions of motion, and is therefore difficult to analyze; (3) different seismic stations will record the same event differently due to heterogeneity in the earth and attenuation of propagating waves; and (4) earthquakes occur infrequently, resulting in an unbalanced prediction problem. Nevertheless, the application of deep learning is rare in this field of research, and therefore may offer a new avenue forward (e.g., Wang et al., 2017; Lipski et al., 2017). In this project, we aim to create an earthquake warning system utilizing deep neural networks that is capable of predicting an earthquake 60 seconds prior to the event occurring.

There are several different approaches to building a prediction model from seismic waveform data. In one approach, the seismic waveform is pre-processed to extract features from the time-series data such as various summary statistics, autocorrelation at different lag values, and fft wavelet coefficients (Chu and Maurer, 2016; Addair, 2012). A related approach condenses information in the waveform into a binary "fingerprint" for a fast data mining algorithm by thresholding a spectrogram of the waveform (Yoon et al., 2015). In our initial testing, we found that the feature extraction approach is too slow for an earthquake warning system. Therefore, we analyze the raw seismic waveform and the spectrogram of the waveform. In the following sections we describe our dataset, deep learning approach to earthquake prediction, and preliminary results.

Dataset

Seismic waveform data is available through publicly funded seismic station networks that monitor and catalog seismic events. Accessing the data is relatively simple using the Obspy python library (Krischer et al., 2015), which allows seismic waveform data to be queried based on location, time, magnitude of events, seismic station, and waveform component. In order to create an earthquake early warning system, we define our data variable to be some nominal "precursor period" prior to a 60 second "warning period" before the earthquake event (Figure 1). The prediction variable is a binary variable (0 - no earthquake; 1 - earthquake).

Because there is no proven method for identifying earthquakes precursors, we have no information about the characteristics of a potential earthquake precursor signal. For example, the precursor signal may only exist prior to large magnitude earthquakes, it may ramp up slowly over a long period of time, or exist as discrete bursts of high-frequency signal. Therefore, in order to find an optimal dataset, we created several datasets with different properties such as magnitude of earthquake events, time period, and number of stations.

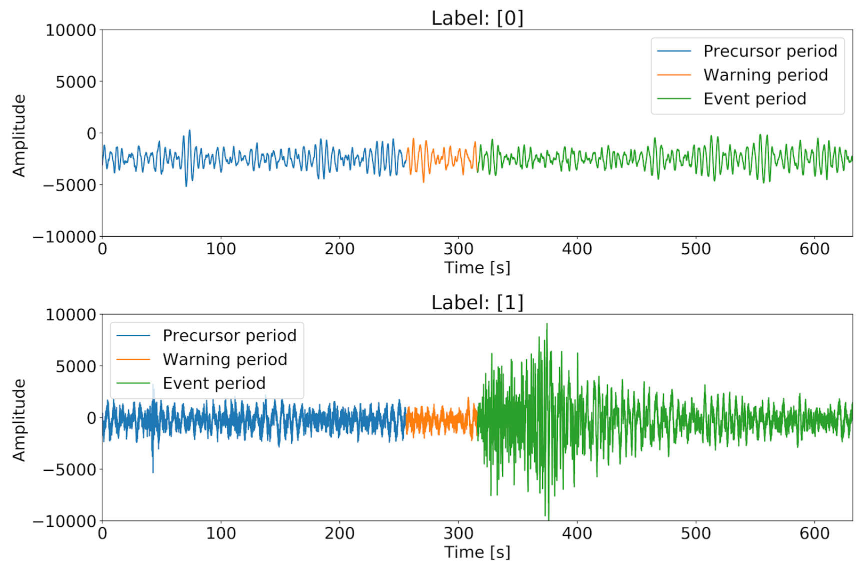

Figure 1. Two samples from the dataset. (Top) A negative example in which there is not earthquake during the event period. (Bottom) A positive example in which an earthquake occurs during the event period.

Approach and initial results

We first developed a simple detection model, i.e. is it possible to automatically detect that an earthquake has occured from changes in the seismic waveform. This problem is far simpler than predicting earthquakes because the signal is easily detected by a human given that (1) the earthquake is large enough, (2) the earthquake is close enough to the station, and (3) the noise level in the data is relatively low. To solve this problem, we developed a one-dimensional conventional neural network with three hidden layers, two conventional and one fully connected. The loss function used is softmax with an L2 regularization. The results are encouraging with a training error of 80% and test error of 72% (Figure 2 shows the training error as a function of the number of iterations). Next, we attempted to train the same network architecture on the prediction problem. The initial results are not satisfactory yet with training error of 66% and test error of 58%. This indicate that we are slightly above the base “guess” answer of 50%. It seems that the networks commonly either overfit the training set, or yield errors that are consistent with guessing the answer (based on the proportion of the positive and negative labels provided).

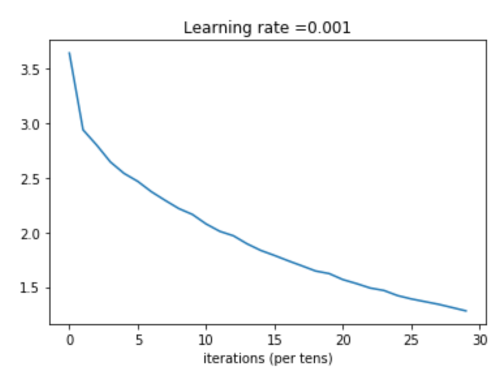

Figure 2. Training error as a function of iteration for the detection problem.

The low performance of the prediction network could be a result of a number of issues including (1) the seismic precursor period does not contain any information about the upcoming seismic event. If this is the case, the problem cannot be solved by neural networks using the current dataset; (2) the seismic precursor signal contains information about the upcoming seismic event but the information is "too hidden". In this case, preprocessing the data using physical principles might be helpful (see spectrogram discussion below); and (3) the network architecture is not suitable for this problem. If this is the case, we need a new architecture (e.g., using RNN).

To enhance the performance of neural network, hyperparameters are tuned via random sampling. From the given distribution in Table 1, 50 examples of hyperparameters are sampled with Latin hypercube sampling. The examples are split to train (70%), validation (15%), and test (15%) sets. The highest test accuracy of 58% is obtained (training accuracy=66%, see Table 1 for details).

Hyperparameter	Distribution	Values at highest test accuracy
Learning rate (log10)	Unif[-5,-2]	0.001
L2 regularization parameter (log10)	Unif[-3,1]	5
Size of the first filter (log2)	[0,1,2,3,4,5,6,7,8]	128
Size of the second filter (log2)	[0,1,2,3,4,5,6,7,8]	64
Minibatch size, (log2)	[0,1,2,3,4,5,6,7,8]	50

Table 1. Hyperparameters and their distributions

Future direction

Our next step is to create a two-dimensional conventional neural network on the spectrogram of the signal (e.g., Figure 3). Running the analysis on the spectrogram might yield better results because the spectrogram separates the frequency and amplitude information which might make it easier for the network to identify patterns (Yoon et al., 2015).

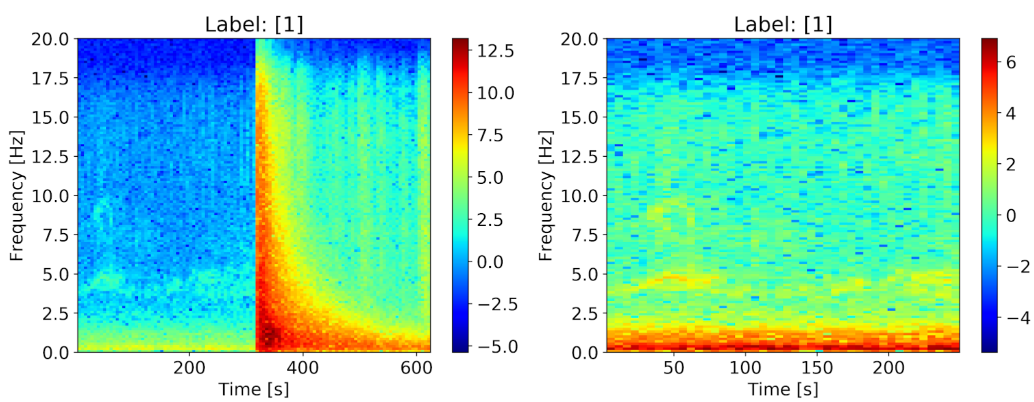

Figure 3. Log10 Spectrogram of a single sample. (Left) Spectrogram showing the precursor period and post-earthquake period. (Right) The same event, showing only the precursor period.

Another issue to address is using a balanced vs unbalanced dataset. Because the vast majority of seismogram data collected by monitoring stations is not related to an earthquake event, our prediction model must be capable of handling an unbalanced dataset, meaning there are more negative labels than positive labels in the dataset. One way to handle this is to reformulate the loss function to include a high penalty on misclassifying the positive samples (Buda et al., 2017).

Code availability

The current code of the project can be found in: https://github.com/MosGeo/TerraeMotus

Team member contributions

Mustafa Al Ibrahim: Focused on dataset retrieval and preprocessing; Some neural network testing and writing.
Jihoon Park: Focused on preparing CNN codes, neural network testing, hyperparameters optimization and writing
Noah Athens: Focused on organizing and writing the milestone report, background research, and generating figures.

References

Addair, T., 2012, A supervised learning method for seismic data quality control: CS 229 report, Stanford University.

Buda, M., Maki, A., and Mazurowski M. A., 2017, A systematic study of the class imbalance problem in convolutional neural networks: arXiv:1710.05381, 23 p.

Chu, S., and Maurer, J., 2016, Can machine learning determine physical source properties of earthquakes from a single station?: CS 229 poster, Stanford University.

Lipski, M., Argueta, C. L., Saunders, M. D., 2017, Earthquake prediction using deep learning: Proceedings of Modeling Complex Systems, University of Guelph, 4 p.

Krischer, L., Megies, T., Barsch, R., Beyreuther, M., Lecocq, T., Caudron, C., Wassermann, J., 2015: ObsPy: a bridge for seismology into the scientific Python ecosystem: Computational Science & Discovery.

Wang, Q., Guo, Y., Yu, L., and Li, P. 2017, Earthquake prediction based on spatio-temporal data mining: an LSTM network approach: IEEE Transactions on Emerging Topics in Computing.

Yoon, C.E., O'Reilly, O., Bergen, K.J., and Beroza, G.C., 2015, Earthquake detection through computationally efficient similarity search: Science Advances, 13 p.