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)

"Journalists and the general public rush to any suggestion of earthquake prediction like hogs toward a full trough... [Prediction] provides a happy hunting ground for amateurs, cranks, and outright publicity-seeking fakers."

Charles Richter, 1977

Abstract

Earthquake prediction is one of the great unsolved problems in the earth sciences. In recent years, the number of seismic monitoring stations has increased, thereby enabling deep learning and other data-driven methods to be applied to this problem. In this study, we test the performance of 1D CNN, 2D CNN, and RNN neural networks on predicting an imminent earthquake given 100 seconds of seismic data. Preliminary results show that RNN with class weighting is preferred. We also show the performance of these methods on earthquake recognition, a simpler problem with applications to data mining earthquake statistics and early-earthquake detection.

Introduction

Earthquake seismology is a major topic relevant to understanding hazards due to natural and induced earthquakes as well as understanding physical properties of the earth's crust. In the past decade, the number of seismic monitoring stations has increased dramatically, leading the field of research to transition from an observation-based science to a data-driven science (Havskov and Ottemoller, 2010). In general, earthquake seismology problems fall into three categories: earthquake recognition for data mining and early-earthquake detection (Allen, 1978; Joswig, 1990; Satriano et al., 2011; Yoon et al., 2015; Petrol et al., 2018), earthquake prediction for a warning system (Scholz et al., 1973; Allegre et al., 1982), and probabilistic risk assessment (Nishenko and Buland, 1987; Kagan and Jackson, 2000; Moustra et al., 2011; Wang et al., 2017; Lipski et al., 2017).

In this paper, we address the earthquake recognition (P1) and earthquake prediction problems (P2) illustrated in Figure 1. We test the performance of 1D CNN, 2D CNN, and RNN neural networks on both problems using a dataset of seismic waveforms from 46 stations near the Geysers geothermal area, California. In the P1 problem, the goal is to predict whether an earthquake *has* occurred given a seismic waveform. State-of-the-art performance is generally high on this problem, largely depending on the magnitude of earthquakes considered. In the P2 problem, the goal is to predict whether an earthquake *will* occur given a seismic waveform. In contrast to the P1 problem, there is no proven analytical method to predict an imminent earthquake (Geller et al., 1997), and therefore state-of-the-art performance is non-existent.

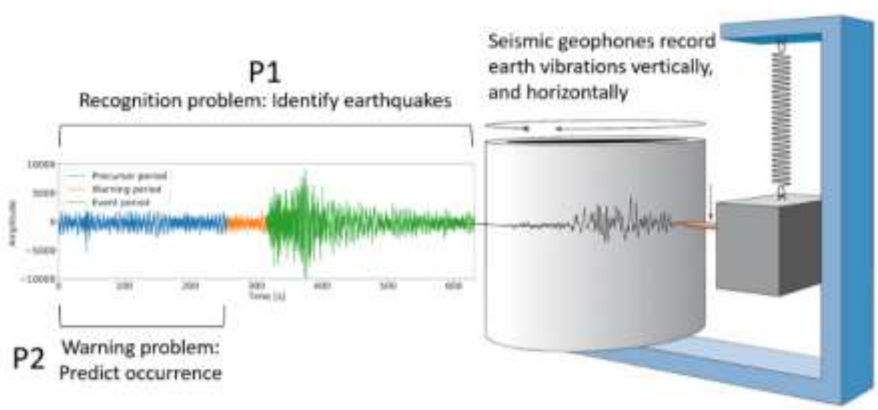

Figure 1. A schematic illustration of the problems tackled in this paper. A seismometer (geophone) records the displacement of the earth as a function of time.

Related work

In recent years, significant progress has been made towards the earthquake recognition problem using machine-learning methods. In these studies, there are several approaches to defining the model inputs. The simplest approach uses the raw time-series of the waveform partitioned into windows of equal time-steps (Perol et al., 2018). An alternative approach is to use the spectrogram of the waveform, which contains the same information as the time-series but is oftentimes a more convenient representation for machine-learning problems (Yoon et al., 2015). A final approach is to extract features from the time-series data such as summary statistics, autocorrelation at different lag values, and Fast-Fourier Transform wavelet coefficients (Chu and Mauerer, 2016; Addair, 2012). However, feature extraction has the drawback that extracted features are not informed by the prediction variable of interest, and therefore may add noise while reducing dimension of the input variable.

Datasets are generally assembled from earthquake catalogs from various seismic station networks around the world. For studies related to small-magnitude earthquakes, there are often ample earthquake events for a machine-learning or deep-learning approach. However, for studies related to large-magnitude earthquakes, which are more impactful to human lives, there are substantially fewer events to draw from. To bolster the number the number of samples, Perol et al., 2017 augmented samples by adding zero-mean Gaussian noise, which they found resulted in a regularizing effect on the model.

In contrast to the earthquake recognition problem, earthquake prediction has not seen significant progress due to the inability to identify earthquake precursor signals in seismic data (Geller et al., 1997; Uyeda et al., 2009). In addition to looking at the seismic signal, researchers have also studied possible earthquake precursors related to emittance of radon gas (Teng, 1980), ultra-low frequency electromagnetic signals (Karakelian et al., 2002), and abnormal animal behavior (Bhargava et al., 2009). Nevertheless, the application of deep learning is rare in earthquake prediction studies, and therefore may offer a new avenue forward.

Datasets and features

Continuous seismic waveform data is available through many publicly funded seismic station networks that monitor and catalog seismic events. Because there is no proven method for identifying earthquake precursors, there is little information about how to create an optimal dataset for the earthquake prediction problem. Therefore, we wrote a python script that generates multiple datasets for specified 1) seismic stations across different regions, 2) minimum earthquake magnitude, and 3) single (vertical displacement measurements) or multi-channel waveforms. After some initial experimentation, we determined that using many stations clustered in a geologically similar environment offered the best balance of containing a large number of samples drawn from the same distribution. We also decided to use a balanced dataset of positive and negative samples based on the conclusions of Buda et al. (2017) who found that training a convolutional neural network on undersampled negative samples yielded the best result.

The procedure used in the python script is as follows: 1) for each seismic station, we query the catalog for all earthquakes above a minimum magnitude and within ~10 kilometers of the station; 2) for each earthquake, we estimate the time of arrival of the seismic wave using the “iasp91” model; 3) we then download the seismic waveform for a specified period of time around the arrival time of the earthquake; and 4) we download a random seismic waveform from the same station to create a balanced dataset of positive and negative samples. The final datasets used in this paper included 1671, 614, and 176 positive samples (single-channel, vertical displacement) for minimum earthquake magnitudes of 3, 3.5, and 4 respectively. These samples are retrieved from 46 stations from the Berkeley Geysers Network (NCEDC, 2014), located ~110 kilometers north of the Bay Area, California (Figure 2). The region is seismically active and with a relatively dense array of monitoring stations.

Figure 3 shows an example seismic waveform for the 1D CNN model. For the earthquake prediction problem, the “precursor period” is used as input, whereas for the recognition problem a small window centered around the earthquake event is used as input. For the 2D CNN and RNN models, the spectrogram of the time-series data is used (spectrogram is a representation of the signal energy at different frequencies). Both inputs, the raw time series, and the spectrogram are normalized using the z-score transformation.

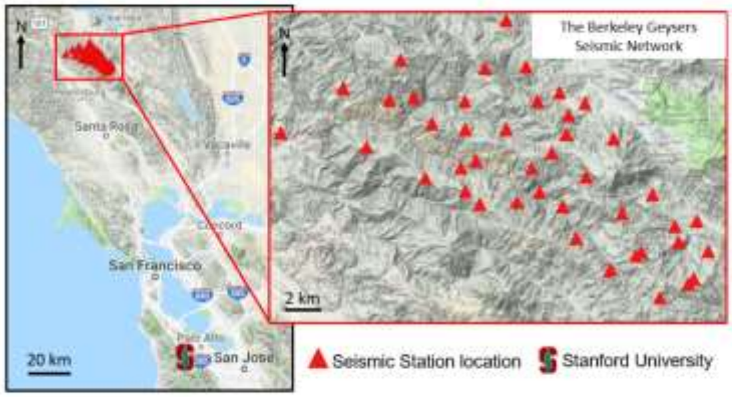

Figure 2. Location of the study area (the Geysers) and the seismic stations used to collect the dataset.

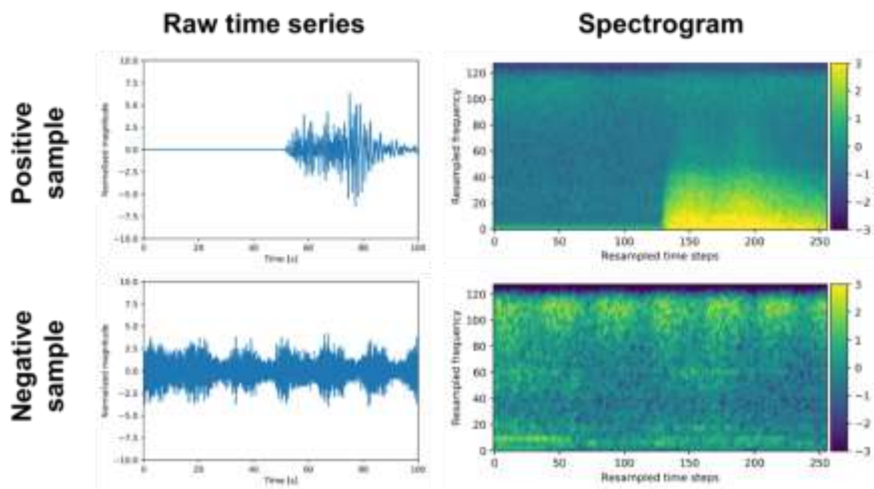

Figure 3. Examples of the raw data and spectrogram. A negative example in which there is not earthquake during the event period. (Bottom) A positive example in which an earthquake occurs during the event period.

Methods

Three neural network architectures, 1D CNN, 2D CNN, and RNN, are used for earthquake recognition and predictions (see Figure 4 for the architectures and types of hyperparameters explored). Models are developed using the Keras library. The 1D CNN model is trained with a series of 1D convolution and pooling layers taking raw seismic time-series data as inputs. The 2D CNN model uses the spectrogram as input (e.g., Figure 3) to train the neural network, consisting of 2D convolution and pooling layers. The RNN model consists of 1D convolution layer followed by two LSTM layers to predict the class using the spectrogram as input. The dataset is first split into training/test sets using a 9/1 ratio. 10% of training set is then used as a validation set. In all networks, binary crossentropy is used for the loss.

Figure 4D lists the type of hyperparameters explored for three neural network architectures. Hyperparameters are tuned by computing validation accuracy. The values of dropout rate and number of epochs are determined so that the neural network does not overfit to the training data. We also choose dilation as one of hyperparameters in order to explore a large area of the spectrogram without increasing the filter size. The class weights are tuned in the RNN architecture as explained in the discussion to force increasing accuracy in a specific class with possible sacrifice in some other accuracy.

Because the optimal dataset for obtaining accurate results is not known, hyperparameters for the data and data processing were explored. The size of the time interval was found to be the most important parameter. For example using a smaller window around the earthquake increases the accuracy of the recognition problem (~1 minute before and after the earthquake). The prediction problem is more complex especially in the 2D CNN and RNN as the size of the studied interval is related to the size of the spectrogram. Generally, increasing the size of the time interval without increasing the size of the spectrogram would decrease accuracy. However, using a very large spectrogram increases computations without enhancing prediction accuracy. Therefore it is preferable that increasing the size of the time interval should be matched by increasing the size of the spectrogram time domain axis.

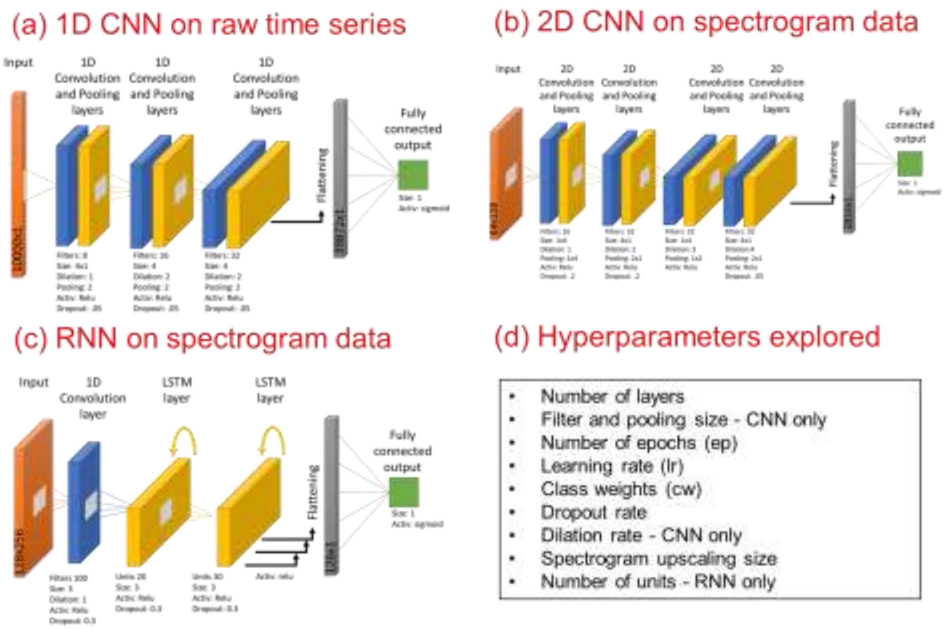

Figure 4. Three neural network architectures (a, b, c) and network hyperparameters explored (d).

Results and discussion

Table 1 shows selected results for the earthquake recognition problem (P1). See Figure 1 for the problems definition. Overall, all the architectures performed well (>90% accuracy) after hyperparameter tuning with 2D CNN and RNN obtaining a 100% test accuracy. This indicates that neural networks can be trained to identify earthquakes reliably.

Table 2 shows selected results for the earthquake prediction problem (P2). The results obtained are higher than 50% (i.e. the expectation of random guessing). This means that there is some pattern in the precursor signal that is indicative of earthquake occurrence in the next minute. It is important to note that these results are typical results of running the training and not the best results. In some runs, the test accuracy obtained reached 62% (especially when using the dataset with higher earthquake magnitudes (>4). However, because higher-magnitude earthquake occurrence is sparse, the dataset is small and so randomness prevails in estimating the true test accuracy.

Model	Parameters	Training Accuracy	Test Accuracy
1D CNN	M = 3.5, lr = 0.001, ep = 10	97.5%	94.4%
2D CNN	M = 3.5, lr = 0.001, ep = 10	100%	100%
RNN	M = 3.5, lr = 0.001, ep = 50	100%	100%

Table 1. Selected results of earthquake recognition problem (P1). M: minimum earthquake magnitude, lr: learning rate, ep: number of epochs.

Model	Parameters	Training Accuracy	Test Accuracy
1D CNN	M = 3, lr = 0.002, ep = 40	56.0%	54.2%
2D CNN	M = 3, lr = 0.001, ep = 12	60.0%	52.6%
RNN	M = 3, lr = 0.001, ep = 100, cw = [0.5, 0.5]	82.5%	54.5%
	M = 3, lr = 0.001, ep = 100, cw = [0.4, 0.6]	83.8%	56.4%
	M = 3, lr = 0.001, ep = 100, cw = [0.25, 0.75]	74.7%	53.9%

Table 2. Selected results of earthquake prediction (P2). M: minimum earthquake magnitude, lr: learning rate, ep: number of epochs.

One interesting test performed that resulted in increasing the overall test accuracy is modifying the class weights (Figure 5). By penalizing the network to prefer true-positives, the true-positive accuracy increased (as expected). Surprisingly, the overall accuracy increased too. This effect can be managed to a point where the network performance start to drop and it starts to predict too many positives.

Overall, earthquake prediction (P2) results are substantially less accurate than the earthquake (P1). This could be a result of a number of effects that can be generally grouped into three issues: 1) no mechanisms generate a signal before earthquake occurrence 2) the dataset and features used does not contain information about the upcoming earthquake, 3) the network architectures are not suitable for the problem tackled. If there is no signal that is generated before the earthquake (issue 1), the problem is unsolvable in its current formulation. If the dataset and features does not contain the information (issue 2), assembling a better dataset might yield a better accuracy. This can be done by a number of ways, for example, manually picking the arrival time of each earthquake signal in each station, choosing an area with higher number of earthquakes, or choosing an area with better seismometers that captures higher frequencies. If it is the network architecture used, exploring more complex architecture might be useful. Note that after the experimentation done in this paper, it is the view of the authors that dealing with issue 2 is the best course of action for the time being. Specifically, constructing a dataset with manual picking of arrival time would give confidence in the dataset, allow us to explore the different variations observed in the data, and experiment with smaller warning time without risking having an earthquake signal in the precursor data used in the training.

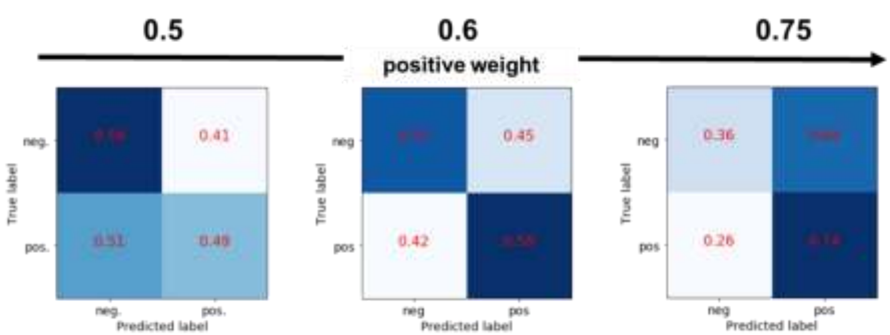

Figure 5. Confusion matrices with different class weights for the RNN network applied on a dataset with a minimum magnitude (M) of 3.5

Conclusions and final remarks

All of the presented neural network models achieved high performance on the earthquake recognition problem (P1). Predicting earthquakes before they occur (P2) is still a challenging problem. Based on the current analysis, some seismic precursor signal may exist. Future work planned includes: 1) experimentation with cleaner and bigger datasets, 2) studying the neural layers that activate for the true positive cases in the prediction problem (P2), and 3) exploring the relationship between warning time and prediction accuracy.

Code availability

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

Contributions

Mustafa Al Ibrahim: Dataset retrieval and preprocessing; Neural network code transfer to keras, 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 report, background research, and generating figures.

References

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

Allegre, C. J., Le Mouel, J. L., and Provost, A., 1982, Scaling rules in rock fracture and possible implications for earthquake prediction: Nature, v. 297, p. 47 - 49.

Allen, R. V., 1978, Automatic earthquake recognition and timing from single traces: Bulletin of the Seismological Society of America, v. 68, p 1521-1532.

Bhargava, N., Katiyar, V. K., Sharma, M. L., and Pradhan, P., 2009, Earthquake prediction through animal behavior: A review: Indian Journal of Biomechanics: Special Issue, p. 159 - 165.

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.

Geller, R. J., Jackson, D. D., Kagan, Y. Y., and Mulargia, F., 1997, Earthquakes cannot be predicted: Science, v. 275, 1 p.

Havskov, J., and Ottemoller, L., 2010, Routine data processing in earthquake seismology: With sample data, exercises and software: Springer, Netherlands, Dordrecht, 347 p.

Joswig, M., 1990, Pattern recognition for earthquake detection: Bulletin of the Seismological Society of America, v. 80, p. 170 - 186.

Kagan, Y. Y., and Jackson, D. D., 2000, Probabilistic forecasting of earthquakes: Geophysical Journal International, v. 143, p 438-453.

Karakelian, D., Klemperer, S. L., Fraser-Smith, A. C., and Thompson, G. A., 2002, Ultra-low frequency electromagnetic measurements associated with the 1998 Mw 5.1 San Juan Bautista, California earthquake and implications for mechanisms of electromagnetic earthquake precursors: Tectonophysics, v. 359, p 65-79.

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.

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

NCEDC, 2014, Northern California Earthquake Data Center. UC Berkeley Seismological Laboratory. Dataset. doi:10.7932/NCEDC.

Moustra, M., Avraamides, M., and Christodoulou, C., 2011, Artificial neural networks for earthquake prediction using time series magnitude data or seismic electric signals: Expert Systems with Applications, v. 38, p. 15032-15039.

Perol, T., Gharbi, M., and Denolle, M., 2018, Convolutional neural network for earthquake detection and location: Science Advances, 8 p.

Satriano, C., Wu, Y. M., Zollo, A., and Kanamori, H., 2011, Earthquake early warning: Concepts, methods, and physical grounds: Soil Dynamics and Earthquake Engineering, v. 31, p. 106-118.

Scholz, C. H., Sykes, L. R., and Aggarwal, Y. P., 1973, Earthquake prediction: A Physical Basis: Science, v. 181, p. 803-810.

Teng, T. L., 1980, Some recent studies on groundwater Radon content as an earthquake precursor: Journal of Geophysical Research, v. 85, p. 3089-3099.

Uyeda, S., Nagao, T., and Kamogawa, M., 2009, Short-term earthquake prediction: Current status of seismo-electromagnetics: Tectonophysics, p. 205-213.

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.