Earthquake Prediction using Deep Learning

Created for ENGG3130 at the University of Guelph

Mark Lipski
School of Engineering
University of Guelph
50 Stone Road East
Guelph, Ontario N1G 2W1
mlipski@mail.uoguelph.ca

Carlos Lopez Argueta
School of Engineering
University of Guelph
50 Stone Road East
Guelph, Ontario N1G 2W1
clopezar@mail.uoguelph.ca

Matthew D. Saunders
School of Engineering
University of Guelph
50 Stone Road East
Guelph, Ontario N1G 2W1
msaund05@mail.uoguelph.ca

ABSTRACT

Major earthquakes have long been considered unpredictable due to the complex patterns of foreshocks and aftershocks. We attempt to build a recurrent neural network with the capacity to identify these patterns to predict major earthquakes, defined as those with a moment magnitude of greater than 5.0. While the model fails to successfully predict earthquakes, we discuss a number of improvements which could increase its accuracy.

CCS CONCEPTS

•Computing methodologies → Neural networks; •Applied computing → *Environmental sciences;*

KEYWORDS

Earthquakes, seismology, recurrent neural networks.

ACM Reference format:
Mark Lipski, Carlos Lopez Argueta, and Matthew D. Saunders. 1997. Earthquake Prediction using Deep Learning. In *Proceedings of Modeling Complex Systems, University of Guelph, April 2017 (ENGG 3130),* 4 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Earthquakes have long been considered unpredictable: the result of highly non-linear systems with unknown feedback loops. Earthquakes occur where two tectonic plates are forced against each other by currents of molten rock in the Earth's mantle. The tectonic plates have interlocking edges (called ”asperities”) which prevent the plates from sliding past each other. Energy continues to be transfered to the plate from the flow of the molten rock: as the energy builds up, the plate's ”slip deficit” — the distance the plate should have moved from the applied force — increases. Finally, in one catastrophic moment, the strain is large enough to overcome the tensile strength of the asperities, and all the energy is released.

This model attempts to predict the unpredictable: by using modern deep-learning techniques to identify patterns in the fault behaviour preceding major earthquakes, we aim to assess the probability of a major earthquake occurring on any given day.

The TensorFlow implementation of the model, the dataset, and an iPython notebook with visualizations can be found at our Git repository[1].

2 BACKGROUND

2.1 Earthquake clustering

Earthquakes are known to be sequential in nature, occurring in clusters. These clusters can take on multiple forms.

Foreshocks and aftershocks. After a seismic event, the tectonic plates reach a new equilibrium. If this equilibrium is unstable, little energy is required to cause another tectonic plate movement. The plates may shift many times before finding a new, more stable equilibrium, in which the asperities lock firmly together and elastic strain energy begins to build up.

Swarms and storms. Earthquake swarms are similar to aftershocks: a series of earthquakes in the same area within a short period of time. However, no earthquake in a swarm stands out as the main quake. An earthquake storm is similar, but occurs when a shift on one part of a fault causes shifting on adjacent parts of the fault, as the whole plate's distribution of elastic strain is altered.

2.2 Earthquake energy

Earthquake energy is measured and reported using the moment magnitude scale (M_W), which superseded the Richter scale in January 2002 [7]. The moment magnitude scale is a logarithmic scale, with energy determined using Equation 1.

$$E = 10^{3/2 M_W} \tag{1}$$

2.3 Earthquake data

Every earthquake detected by a seismic station has its time, location, depth, and intensity recorded, along with other parameters relating to the uncertainty and the locations of monitoring stations. The data is correlated by the Advanced National Seismic System (ANSS) into its comprehensive catalog (the ANSS ComCat). The ANSS ComCat data is collected from 16 separate contributing networks, each of which may contain multiple monitoring stations and institutions.

For the purposes of this model, data was chosen from the two contributing networks in the California region which cover the San Andreas Fault: the Southern California Seismic Network (SCSM, at Caltech) and the Northern California Earthquake Data Center (NCEDC, a collaboration between the United States Geological

[1] Github repository can be found at https://github.com/nevelo/quake-predict.

Survey and the Berkeley Seismological Laboratory). These networks are considered authoritative sources for earthquakes which occur within the region of the San Andreas fault. The map in Figure 1 gives the exact locations of the seismographs, and the authoritative areas covered by these networks, with SCSM shown in purple and NCEDC in orange.

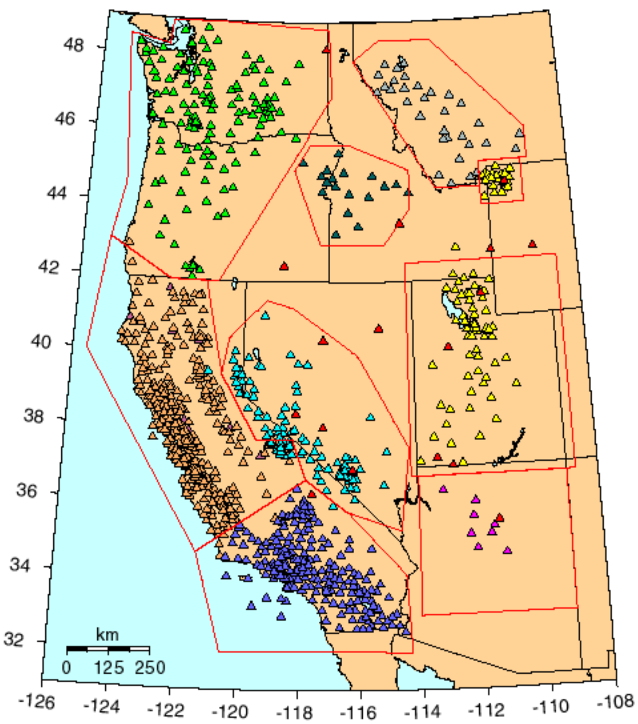

Figure 1: Map of the western United States, showing the seismic monitoring stations along the San Andreas fault and elsewhere. Image credit NCEDC Berkeley.

The San Andreas fault is a continental strike-slip fault, located in a densely-populated area. As a result, it is one of the most monitored fault systems in the world. Additionally, the data for this fault system is publicly available through the ANSS, making it an ideal choice to train a neural network.

2.4 Deep learning

As mentioned above, earthquakes are sequential. To model earthquake patterns effectively, the time-dependent nature of the data must be acknowledged. Recurrent neural networks, which are designed for pattern recognition in time-based data, are used in this model. In particular, the long short-term memory module [4] was used as it is less prone to the diminishing gradient problem.

As important contextual data must be obtained from further in the past, the amount of calculations between that data input and the desired output grows. As the amount of long-term dependencies grows, gradient descent becomes inadequate: the repeated multiplications drive the gradient either to zero or infinity [1].

2.5 Previous work

The survey paper by Florido et al. [3] discusses the current state-of-the-art of earthquake prediction using neural networks. Currently, no earthquake prediction model uses the long short-term memory module, though a sophisticated neural-network based system is currently in use to predict timing and location of earthquakes in Chile [6].

This paper describes a much more complex model for the earthquake-prone country, differentiating based on seismogenic area, and accounting for variable rates of tectonic plate movement. However, the inputs to the model are passed directly into a feed-forward neural network without any cycles or recurrence. Additionally, the performance of the model is evaluated by considering true and false positives and negatives, allowing for a more accurate measurement of performance.

Geophysically, the distribution of earthquake magnitude follows a power law [5]. Previous work has determined a parameter, called the size distribution factor or *b-value*, which describes the distribution of earthquake magnitude, and has been empirically estimated for various regions of the planet. The b-value reflects the geophysical properties of the rocks and the tectonic activity in a region, and analysis of its variation has been useful in earthquake prediction [6].

3 METHODOLOGY

3.1 Neural network topology

The neural network is set up as a pair of LSTM modules. The inputs to the LSTM module are, for each day, the number of recorded earthquakes and the total earthquake power. The LSTM is unrolled 400 times, thus using, for any given date, the last 400 days worth of data.

Long short-term memory. The long short-term memory cell is implemented in `my_lstm_cell.py`. This cell implements an earthquake model based on remembering two parameters: one which represents the amount of energy stored within the fault system, and one which represents the current instability of the fault. For each training pass, each of the 400 LSTM modules determines, based on that day's data, what emphasis to put on remembered data, using a fully-connected network. It then adds the previous data, the current data, and a bias to produce an output. Two parallel LSTMs compute the instability and the current energy parameters, which are summed together and passed to a logistic function to determining the probability of the earthquake.

Cost function. Error is computed by comparing the computed probability of a major earthquake to the actual occurrence of a major earthquake on that date. Using logistic regression, a false negative or false positive is penalized heavily, and uncertainty is penalized less heavily. Error is computed using cross-entropy between the prediction output and the actual value for that date.

A diagram of the neural network topology is shown in Figure 2.

3.2 Data collection

The data for this model was collected from the ANSS ComCat database, which records in text file format every detected earthquake in the USA, with each line in the text file representing a single

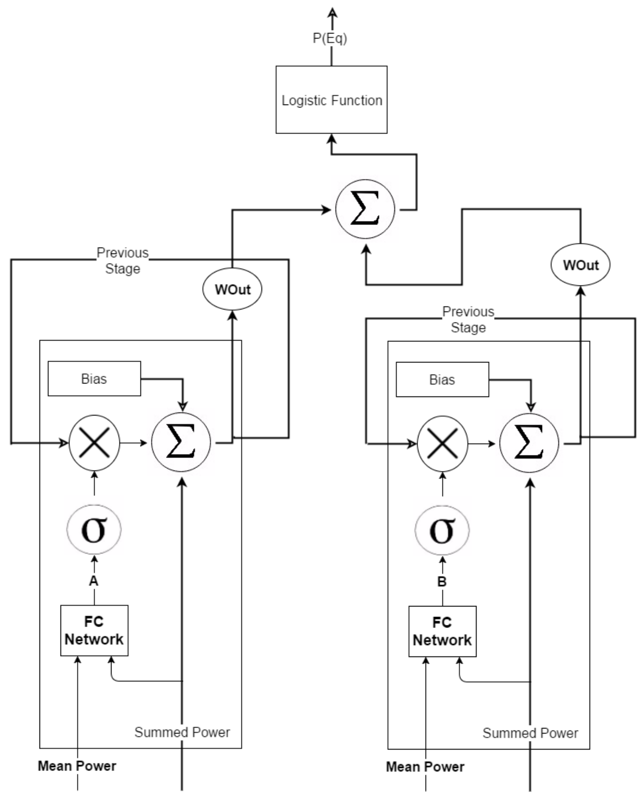

Figure 2: Topology of LSTM module.

earthquake. To prepare this dataset for use in a neural network, we filtered the data by contributing network, to only include earthquakes detected by the California networks.

Next, we grouped each day's data into one entry, as the neural network operates on discrete time chunks of one day. For each day, the number of recorded earthquakes and the cumulative energy dispersion was determined. Additionally, the number of recorded earthquakes with a magnitude higher than 5.0 on that day was recorded.

For simplicity, much data was discarded. All earthquakes in the dataset happened within California, but exact latitude, longitude, and depth are not provided to the model.

4 RESULTS

The neural network was largely unsuccessful at predicting earthquakes in the future. Unfortunately, due to the computational power of the systems available, as well as the inefficiency of the existing neural network, implementing a complicated model on a complex dataset wasn't feasible.

4.1 Predicting future earthquakes

The output of the model can be viewed in Figure 3. It can be seen that the model doesn't do an effective job of predicting the occurrences of earthquakes at all. Actual earthquakes occur approximately fifteen times throughout the testing time period, however, the neural network reports numerous false negatives until a threshold is reached, after which it reports false positives for the remaining period.

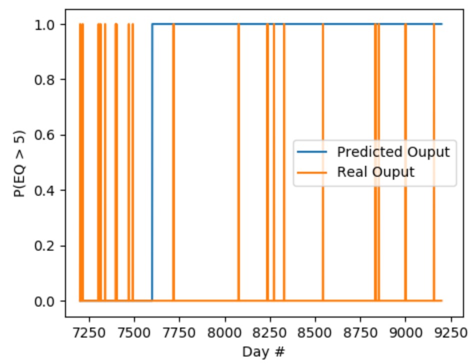

Figure 3: Comparison of the model and actual results.

5 CONCLUSIONS

Plate tectonics is a complex system. Each section of each fault moves semi-independently, and can store energy for hundreds of years before release. Shocks on one part of a large fault system likely have no predictive value towards earthquakes occurring hundreds or thousands of miles away. Additionally, though the San Andreas fault is the most studied fault in the California region, much is still unknown about the region. Most major earthquakes happen on previously-unknown faults, where data is not actively collected. It is also difficult to use the measurements to extrapolate information about the mantle magma currents which impart energy to the system: the seismic measurements only give indirect information about the underlying system stocks and flows.

Secondly, it is important in earthquake prediction to eliminate false negatives. Any system which attempts to accurately predict earthquakes has a high value to the general public, and is likely to be used to make public safety decisions. False negative earthquake predictions have led to jail sentences for scientists [2], and should be penalized heavily by the model. False positives should also be penalized heavily as they can lead the public to disregard predictions. Implementing a cost function which effectively models this requirement would make for a more accurate model.

To further increase prediction accuracy, the model must account for location data. It should also use as inputs the previously-identified earthquake parameters, such as the b-value, to improve location-based prediction by identifying deviations from typical seismic activity. Finally, adding a fully-connected neural network using the output of each unrolled LSTM module would allow the network to learn a more complex model which more directly includes past information.

ACKNOWLEDGMENTS

The authors would like to thank the esteemed Dr. Graham W. Taylor, whose first offering of the ENGG 3130 course opened a window on the fascinating world of non-linear, complex systems analysis. This is why we're ES&C.

REFERENCES

[1] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks* 5, 2 (March 1994), 157–166.

[2] Edwin Cartlidge. 2014. Updated: Appeals court overturns manslaughter convictions of six earthquake scientists. *Science Magazine* (November 10 2014).

[3] Emilio Florido, José L. Aznarte, Antonio Morales-Esteban, and Francisco Martínez-Álvarez. 2016. Earthquake magnitude prediction based on artificial neural networks: A survey. *Croatian Operational Research Review* 7 (December 2016), 159–169. DOI: http://dx.doi.org/10.17535/crorr.2016.0011

[4] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural Computation* 9, 8 (December 1997), 1735–1780. DOI: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[5] M. Ishimoto and K. Iida. 1939. Observations sur les seismes enregistrés parle microsismographe construit dernièrement. *Bulletin Earthquake Research Institute* 17 (1939), 443–478.

[6] J. Reyes, A. Morales-Esteban, and F. Martinez-Alvarez. 2013. Neural networks to predict earthquakes in Chile. *Applied Soft Computing* 13, 2 (February 2013), 1314–1328. DOI: http://dx.doi.org/10.1016/j.asoc.2012.10.014

[7] USGS. 2017. Measuring the Size of an Earthquake. https://earthquake.usgs.gov/learn/topics/measure.php. (2017).