Earthquake Classification and Magnitude Prediction for Earthquake Early Warning

Hassan Aljama and Fatimah Al-Ismail

Introduction

Accurate and early detection of earthquakes is of critical importance in formulating an effective response. Hundreds of thousands of humans are in a constant state of danger due to living in areas vulnerable to earthquakes. The problem is compounded if the area experiences constant low magnitude earthquakes that, although does not pose danger to human lives, present a challenge in evaluating the potential danger of the earthquake. Generally, an earthquake of magnitude less than 4 does not constitute a danger. However, a higher magnitude earthquake demands an immediate response. The current detection system generally evaluates the earthquake magnitudes correctly in roughly 90% of the time.

The purpose of this project is to detect the magnitude of an earthquake given a set of characteristics. These characteristics were extracted from the earthquake waveforms for different snippets of time, ranging from 1 to 4 seconds. The main goal is to be able to use the smallest time interval possible to accurately predict the earthquake magnitude. Achieving this goal would be of huge value in earthquake early warning systems.

Data

We have 373,731 earthquake observations, each recorded on a 3-component geophone. For each observation, we have a number of 24 extracted features of the waveform, such as peak amplitude, cumulative absolute velocity, and maximum step between consecutive samples. We also have different attributes for each earthquake, such as magnitude, hypocentral distance and hypocentral depth. We also have >800,000 non-earthquake data. Figure 1 below shows the distribution of our dataset, and figure 2 shows a sample waveform

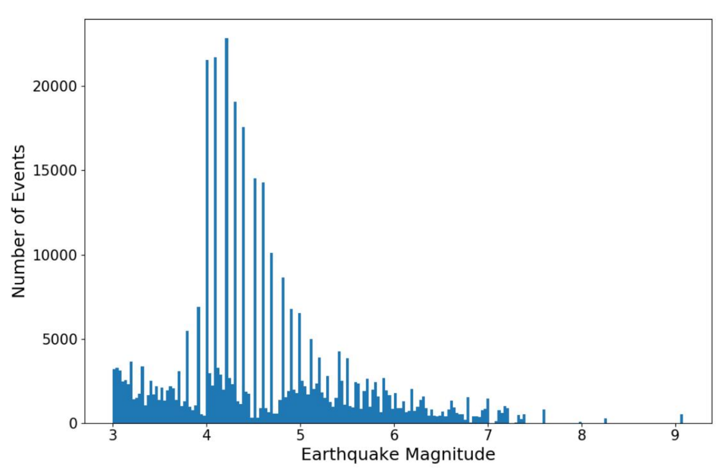

Figure 1: Histogram of the earthquake magnitude in the labelled data

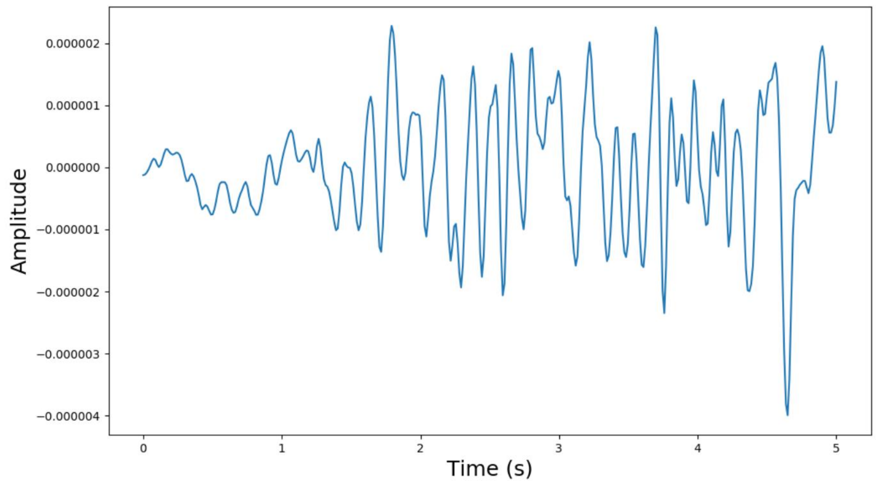

Figure 2: Sample earthquake waveform from dataset

Algorithm

We approached this problem by developing two independent neural networks that we envision to work hand-in-hand. The first approach is a 1-D convolutional neural network that takes a 4-second waveform as an input (figure 2), and performs a binary classification to determine whether it's an earthquake (1) or noise (0). The second approach is a fully-connected neural network that, when the signal is of an earthquake, it predicts its magnitude from a given set of features. Both algorithms are explained below:

1. <u>1-D Convolutional Neural Network</u>

The architecture we chose for the 1-D Convolutional Neural Network is shown in figure 3. The waveform in figure 1 is used as an input, with shape (m, 400,1). The four convolutional layers consists of 18, 36, 72 and 144 filters, respectively, and kernel size of 2. Each convolutional layer is followed by a max-pool layer, and then batch normalization, and dropout of 0.2 for regularization. The final dense layer uses sigmoid activation to give either 0 or 1.

This part of the code utilized was done in Keras. We looked through the available codes it Github repository to decide on an architecture design and coded accordingly. The data was split as 90% for training, and 10% for validation/testing.

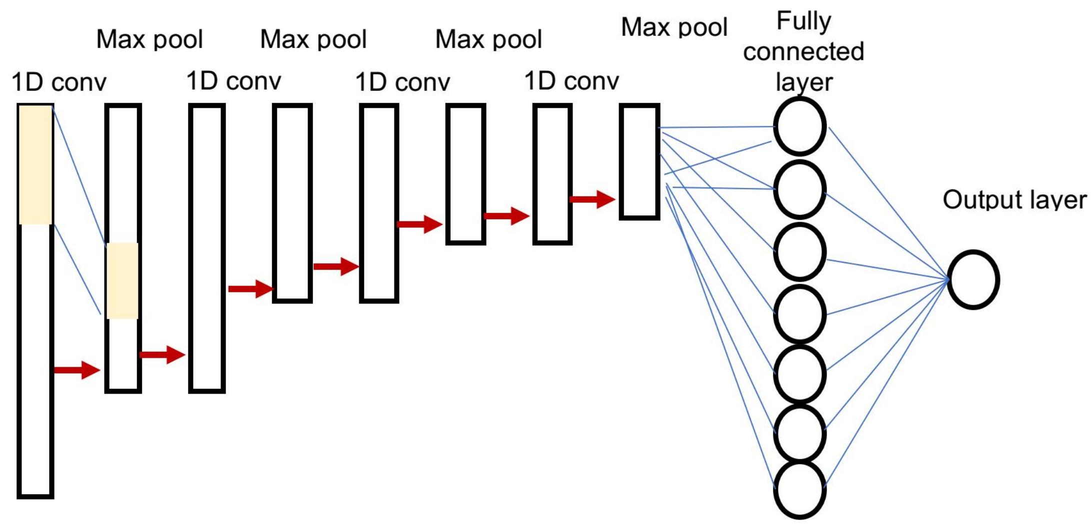

Figure 3. Architecture of 1-D CNN

2. Fully-Connected Network

In the second network, a fully connected neural network is used to predict the earthquake magnitude. We extracted 24 different features based on the waveform that formulated the input to the neural network. The data have generally been split to a 90-10 split between the training set and test/development set. We modified the model used in class based on TensorFlow implementation. The model generally has the following architecture:

Linear -> Relu -> Linear -> Relu -> Linear -> Softmax

We tested the impact of different hyperparameters: number of layers, β parameter of the L2 normalization, and the number of non-earthquake events used (since we have many more non-earthquake events than real earthquakes). We started with simple prediction (earthquake and non-earthquake) and then explored the model's ability to predict the magnitude.

Results

1. 1-D Convolutional Neural Network

We explored the impact of a different number of layers, different optimizers and different learning rates. The summary of the trials is listed in the table below. Although we managed to get an almost perfect accuracy on the training set, we couldn't manage to bring up the test accuracy, which was constant at 55%. We tested different fixes for overfitting: we normalized the input data, applied dropout, applied L-2 regularization in the dense layer, and tried a simpler CNN architecture to reduce complexity, but the test accuracy didn't change.

The model that gave the best results for training accuracy was the model with 4 layers, and Adam optimizer with 0.01 or 0.001 learning rate. In this model, the input data wasn't normalized, but we applied batch normalization after each Max-pool layer. Figure 4 shows the accuracy of this model for training over 80 epochs.

Table 1. Summary of different 1-D CNN trials

#Layers	Optimizer	Learning rate	Normalized input	Accuracy	test acc
4	SGD	0.001	no	52%	44%
4	SGD	0.01	no	55%	55%
2	Adam	0.001	yes	55%	55%
3	Adam	0.001	yes	91%	55%
4	Adam	0.001	no	99%	55%
4	Adam	0.01	no	99%	55%

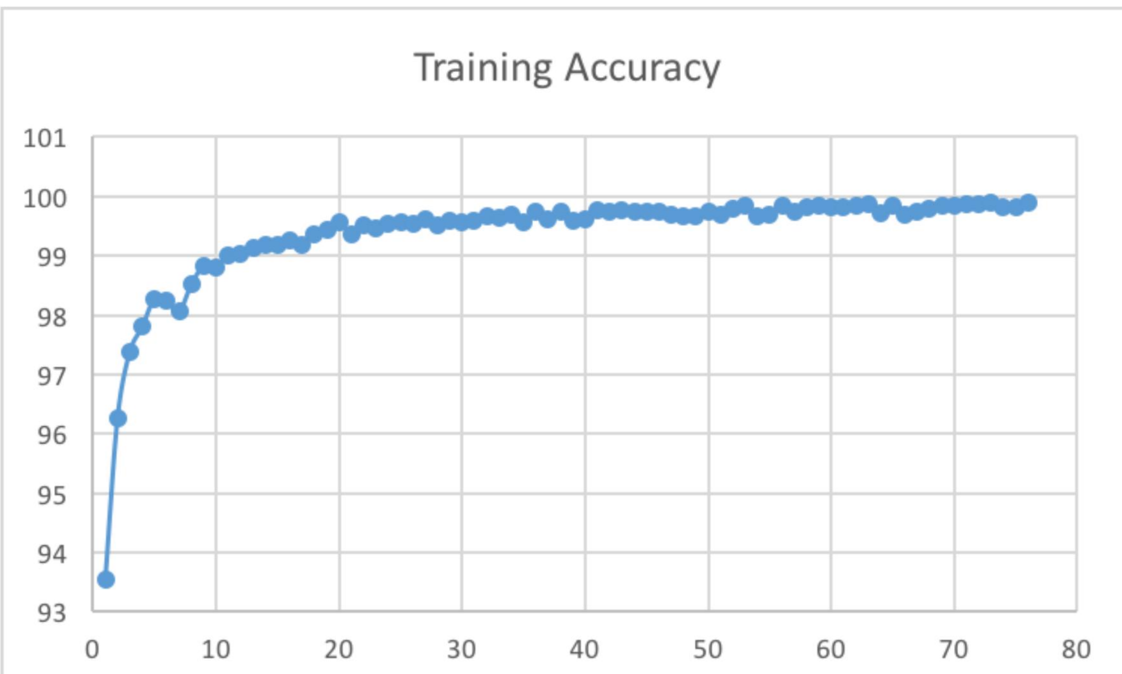

Figure 4 1-D CNN training accuracy

2. <u>Fully-connected network</u>

In the fully connected network, we explored the impact of different number of layers on predicting whether the event is an earthquake or not (based on equal distribution of earthquake and non-earthquake events. Results, shown in Table 1, indicate that a simple architecture can give accurate predictions. Test set also give comparable results with close to 98% accuracy.

Table 2: Impact of number of layers on training accuracy for a binary classification in the fully connected network (equal distribution of earthquake and non-earthquake)

3 layers	5 layers	10 layers
98%	99%	99%

We also tested the impact of the β hyperparameter in L2 normalization. Results, shown in Table 2, indicate results close to 99%. Test accuracy in this case is also close to 98%

Table 3: Impact of β hyperparameter on training accuracy for a fully connected network with binary classification (5 layers and equal distribution of earthquake and non-earthquake).

0.1	0.01	0.001
96%	98%	99%

Lastly, we explore the neural network ability to predict the earthquake magnitude: either <3, between 3-4, between 4-5, or >5). We explored the following hyperparameters: Number of layers and number of non-earthquake events used. The best case scenario with the simplest architecture was found with 5 layers and close to equal distribution of earthquake and non-earthquake events. The test accuracy in that case was 88%.

Table 2: Impact of number of layers and number of non-earthquake events used on training accuracy for a fully connected network with a softmax optimization

	Number of non-earthquake data used		
# of Layers	50k	150k	300k
3 layers	82%	86%	86%
5 layers	83%	86%	89%
10 layers	84%	87%	89%

Future Work

First of all, we want to investigate why our test accuracy for the 1-D CNN is almost random while we are overfitting the training dataset. We will collect more earthquake data. Then we will train the model using the full data set and then test the accuracy on the test set. We will also examine the hyperparameters to see how that impacts the results. We also want to test a different architecture, namely a 2-D Convolutional Neural Network using spectrograms instead of 1-D temporal, to see if it gives favorable results. We also would like to test if all 24 features are needed, or if only a sub-set is needed.

Lastly, we want to minimize the time window that we use for our magnitude detection. In our initial run, we used a 4-second time interval. Moving forward, we will run our algorithm on the smaller intervals, and then compare the accuracies achieved for each time interval. Using the smallest time interval possible with high accuracy will be very of huge importance for early warning systems, and could reduce the damage that is caused by a few-seconds delay.

Contributions

Hassan Aljama took care of data pre-processing, and the fully connected network. Fatimah Al-Ismail took care of the 1D CNN. Both participated in writing the project report and poster.