Seismogram and 3D Model: Similarity and Its Use in Seismic Classification, Prediction and Detection

Tianyu Hu*
Department of Intelligence Science and Technology
Peking University
arcturus@stu.pku.edu.cn

Abstract

In this era of big data, automatic classification, prediction and detection of seismic events with high efficiency has been made possible due to the appearance and development of neural network. However, because of the existence of attenuation, far earthquakes($10°$ to $105°$), which is the main component of my dataset, are quite noisy. Moreover, methods using the P-wave and S-wave selecting often requires annotated data which could consists errors. This project not only focuses on developing a new way to effectively classify and locate seismic data into earthquake events, glacial earthquakes and a new catalog of anomaly seismic events, but is also aimed to figure out whether it is possible to use computer-vision methods(especially from 3D vision) in seismic tasks. Here I show a successful transform of PointNet which achieved 100% accuracy in the classification task, and I will also give an cursory idea to achieve higher accuracy in the prediction stage. Furthermore, the results reveals that viewing a set of seismographs as points as possible, which indicates that there may be some kind of similarity between seismogram and 3D vision.

1 Introduction

Global Seismographic Network (GSN) is deployed worldwide to listen to the sound of the Earth. The recorded signals, also referred to as seismograms, contains rich information on the environmental processes, seismic hazard, and could also be used to monitor human activities. Thus, how to deal with and fully use the large amount of data acquired by GSN becomes an important problem. Since the processing of these data mainly contains three stages: classification, prediction and detection, here we present three corresponding tasks. Through these tasks, I will also reveal the similarity between seismic tasks and 3D-vision tasks.

The first is to distinguish earthquakes and glacial earthquakes(caused by glacial calving). Its major challenge lays on two problems: 1) we only have real-time time flow of the network station waveform data and have no previous knowledge about the source's location, depth or magnitude parameters. 2) Sometimes signals can be pretty weak and the signal-to-noise ratio is relatively low.

The second task is to determine the source parameters of a given (glacial) earthquake. From this stage, relative tasks becomes significantly harder than classification. But traditional method [2] has actually achieved an impressive result in 2006, so I tried to use machine learning methods. The result is slightly worse than the traditional surface-wave method [2]. But given its simplicity while using, I consider that it is still a good method. Meanwhile, I also gave an idea to combine the traditional method and the 3D-vision learning strategy.

The last is to detect an anomaly seismic event. To categorize them, I apply the network used in classification task to these data. And the global feature, which is gained from the network, can be viewed as this event's type's pattern.

*Use footnote for providing further information about author (webpage, co-first authors, *etc*.).

2 Dataset

The dataset includes two typical seismic events, 537 normal earthquakes, and 347 glacial events, and an unusual category, 136 anomaly seismic events(used in task3). The binary data of each event is formed by a Numpy 2D array (129x7200 size, dtype="float64"). Vertical indices indicate the index number of the GSN station (129 stations in total), horizontal indices represent the time series of the seismogram waveform data (The sampling rate is fixed at one data point per second, and each trace contains 7200 data points). For detailed information, please see the "data" folder in code attached.

3 Classification

3.1 Related works

STA/LTA algorithm Stevenson [3] used P and S waves the detect earthquakes. This method calculates the mean value of the characteristic function of seismic motion signal in a pair of specific long window (LTA) and short window (STA). When the ratio of STA to LTA exceeds a certain threshold, which indicates that the energy of the P wave significantly exceeds the energy of the background noise, it can be considered that the seismic event has happened. However, there are mainly two drawbacks when applying this algorithm to my classification task. One is that P wave is often not significant in far earthquakes seismograms. Another is that different threshold values can produce entirely different triggers and accurate thresholds are hard to generalize for large number of events.

PointNet Charles [1] did not designed it for solving seismic problems originally, instead, PointNet is suggested to deal with point cloud's irregular format which is exchangeable between points. However, taking each seismic station as a point on the spherical coordinate system, we can assume that a set of seismograms is just the same as 2D pointcloud, thus, the classification task becomes determining which category a certain "point cloud" belongs to. So I consider that such a symmetric function perfectly matches the seismic data format and the task. For example, from this perspective, STA/LTA method generates a point's local feature. And parameters such as whether it is an earthquake, source latitude, etc. are exactly the global feature we need.

3.2 GeoNet

Pipeline First we need to process the raw data and shape it to (Batchsize,129,7202), then throw it into the Symmetric Encoder to get the global feature. Finally we simply put the global feature through a two-layer fully-connected network and dropout layers with another fc-layer which determines what category it should belong to.

Why choose symmetric structure? This is based on the properties of the traces in a general seismogram acquired by GSN. Each trace is related to a certain seismic station, and changing the order of the traces will not make any actual difference to the data.

This property, invariance under transformations, is the obstacle when simply applying CNN to seismograms. If we simply put the raw data into a CNN network, we can find that even if only two traces' positions (indexs) are switched, a huge difference occurs in the result. Many may also choose to sort the traces in a fixed order, such as using the station index. However, when the number of stations gets greater or we add and delete traces, this sorting strategy is not capable any more. So sorting fails to be generalized wider conditions.

Thus, we need to guarantee that the encoder (or network) itself is a symmetric function. Inspired by PointNet [1], and the property of symmetric function:

$$f(\{x_1, ..., x_n\} = g(h(x_1), ..., h(x_n)$$

where g is a symmetric function, then f is also a symmetric function

I used a symmetric encoder (Figure 1), where a max pooling over the traces serves as the g function.

Data processing In addition to the data points contained in the raw trace, corresponding station's location (latitude and longitude) should be concatenated to the data, in order to fully use the information, especially in the next task (prediction). A tiny flaw in my dataset also needs to be solved, that is, some values in certain traces are nan (missing) in the raw data. So all nans are converted to 0.

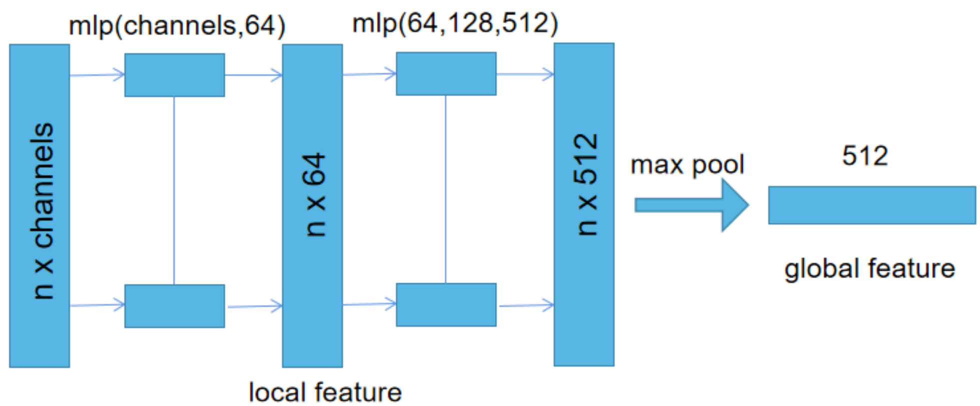
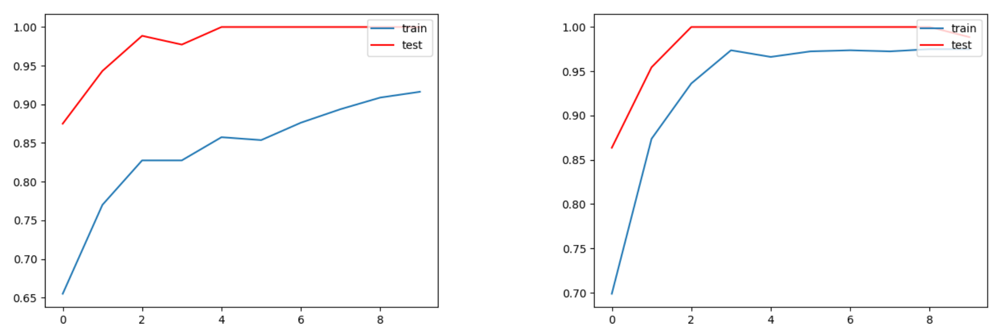


Figure 1: Symmetric Encoder. This most important part of network serves as a symmetric encoder which extracts the global feature of a set of seismograms. *n* and *channels* stand for the number of traces in the gram and the number of data points in a trace, here they are 129 and 7202 (7200 data points together with the longitude and latitude of the corresponding seismic station). *mlp* means multi-layer perceptron. Batchnorm and ReLU layers are used after each fully connected layers.



(a) **Subfigure 1**. Using Dropout.

(b) **Subfigure 2**. Not Using Dropout.

Figure 2: **Dropout test.** Both reaches one hundred percent accuracy in ten epoch.

The waveform data has been filtered between 20s to 150s. (use this period to denote the low-frequency band, meaning 1/20.=0.05 Hz to 1/150.=0.0067 Hz)

After shuffling the data, I split the total 884 pieces of them into train-set (500 earthquakes and 300 glacial) and test-set(57 earthquakes and 37 glacial).

Data augmentation Besides, since my dataset only contains several hundred pieces of data, I need to do data augmentation. However, I have tried to add Gaussian noise to the data, but it does not make remarkable improvement in this stage.

So I used a special dropout strategy which stochastic throw away a random ratio of traces in the training phase. This method not only increases the number of different data, but also ensures that the network will not overfit the train-set, which occurs easily without it. Actually, after applying this dropout function, test-time accuracy will be prominently greater than the train-time accuracy. (shown in Figure 2) The difference between using dropout or not may seems to be tiny according to the two grams, but actually, if the signal-to-noise ratio is a lot higher than this simple dataset, the power of dropout will reveal.

This phenomenon is caused by the fact that this dropout function is deactivated during test stage. That means it is equal to train a large set of weak classifiers on part of the data and use them all in the test-time (like voting strategy).

Results After only **ten** epochs, my GeoNet reaches a 100% accuracy in classification task. I suppose that its reason is the simplicity to distinguish earthquakes from glacial earthquakes. But this

Table 1: **ratio of Difference ranges.**

Name	0.7	2	10
sum difference	0.025	0.075	0.825
Lat difference	0.325	0.65	0.85
Lon difference	0.025	0.075	0.825

part of experiment still preliminarily reveals that there is probably some kind of similarity between seismic tasks and 3D-vision tasks, so that sometimes we can apply same methods on them.

4 Prediction

4.1 Related works

Surface-wave method Ekstrom [2] gives this traditional earthquake detection and location method which is nearly perfect. In the detection stage, he mainly used a deconvolved surface-wave propagation operator and an envelope of the seismogram to improve signal-to-noise characteristics, and thus give predictions about when and where a seismic event may occurs, together with a confidence and predicted strength. And in the location stage, it performs backward searching and finetuning. In my opinion, this method is kind of a combination of forward modelling and backward modelling. The result this essay acquired is pretty good, the mean of distance between real epicenters and predict ones is merely 48 km (which is approximately 0.45 degree) and all of the errors lie within 200 km (approximately 1.8 degree) threshold. (shown in Figure 3.1)

4.2 Geonet

Since the traditional method has achieved great success, here I choose to give learning methods a try because of its simplicity and easiness of understanding.

Difficulty The real challenging task is to determine the depth and magnitude. As the majority of the dataset is far earthquakes (even the nearest station in GSN is beyond 1000 km range), the main component is not P wave and S wave any more, instead, surface waves contributes a large part of energy. In another word, an 80km-depth earthquake and a 100km-depth one shows no significant difference when you stand 1000km far away. Though reflected P wave and S wave do exist, but it is hard to pick them out because of the noise and the attenuation. Also, the traditional surface-wave method [2] does not deal with the depth of the source as well.

So I simply abandoned the prediction of depth and magnitude, and focus on get the location of the epicenter. And I also only applied it to the earthquake category.

Train separately or together? Due to the symmetry of latitude and longitude, I choose to train them together. Some experiments also indicate that the accuracy will get higher slowly if train them separately.

Result After 2000 epochs (about 6 hours) training, more than 80% of the difference (the max of difference of each lat and lon) between real epicenter and the predicted one lies in the range of $[-10^\circ, +10^\circ]$. (shown in Table 1)

Weird phenomenon As can be observed from the Table 1, the difference of source latitude is always smaller than the one of longitude. This is quite interesting and requires further study.

5 Detection

Here I simple trained the classification network to distinguish earthquakes, glacial earthquakes and these unusual ones. Then we can take the global feature as their each pattern.

References

[1] Su H. Kaichun M. Guibas L. J. Charles, R. Q. Pointnet: Deep learning on point sets for 3d classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern*

Recognition (CVPR), 2017. "doi:10.1109/cvpr.2017.16". 2

[2] G. Ekstrom. Global detection and location of seismic sources by using surface waves. *Bulletin of the Seismological Society of America*, 96(4A):1201–1212, 2006. 1, 4

[3] R. Stevenson. Microearthquakes at flathead lake, montana: A study using automatic earthquake processing. *Bulletin of the Seismological Society of America*, 66(1):61–79, 1976. 2