Ground Shaking Prediction

with Graph Neural Network and Semi-Supervised Learning

Phiphat Chomchit

August 19, 2021

Chiang Mai University

Ground Shaking Prediction.

- Ground Motion Prediction Equations (GMPEs)
- Temporal and Spatial Prediction
- Imbalanced Data

My work

- Graph Convolutional Neural Network
- Self-Supervised Learning
- Semi-Supervised Learning

Why do we need machine learning?

1. Seismic data
   - Imbalanced Data
   - Temporal and Spatial Data
2. Purpose of model
   - rapid warning
   - peak ground motion prediction

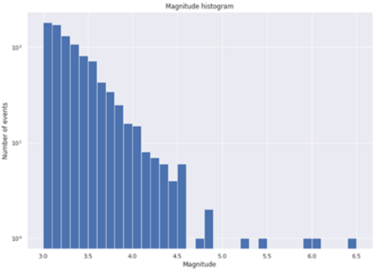


Figure 1: seismic data are long tail data. (Dario, 2020)

1. **Earthquake** is the sudden fracture and movement of rocks inside the Earth. Part of the energy released produces seismic waves.

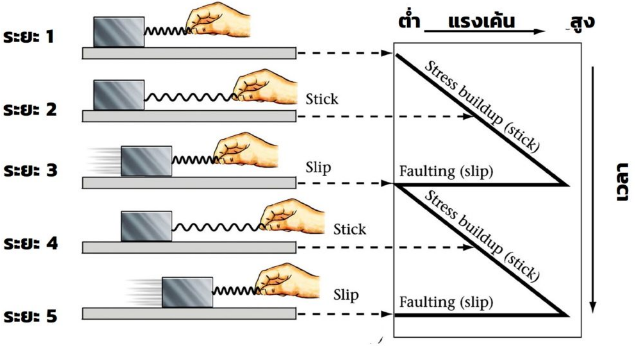


Figure 2: stick-slip model (Santi Pailoplee, 2021)

2. **Hypocenter** or Focus the point below the epicenter at which an earthquake begins.
3. **Epicenter** the point (map location) on the Earth's surface directly above the hypocenter or focus of an earthquake.

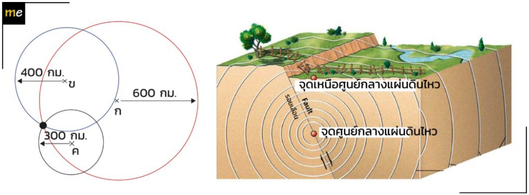


Figure 3: Epicenter and Hypocenter (Santi Pailoplee, 2021)

4. **P Wave** is the primary body wave; the first seismic wave detected by seismographs; able to move through both liquid and solid rock. P wave is the fastest wave.
5. **S Waves** is shear waves that are secondary body waves that oscillate the ground perpendicular to the direction of wave travel.

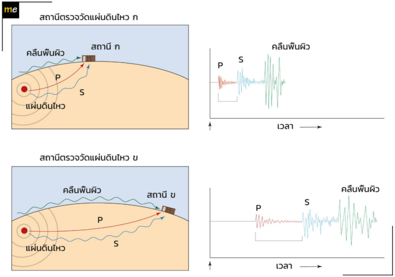


Figure 4: P-wave, S-wave and surface wave (Santi Pailoplee, 2021)

6. **Peak Ground Acceleration** is a largest increase in velocity recorded by a particular station during an earthquake. (Commonly called PGA)

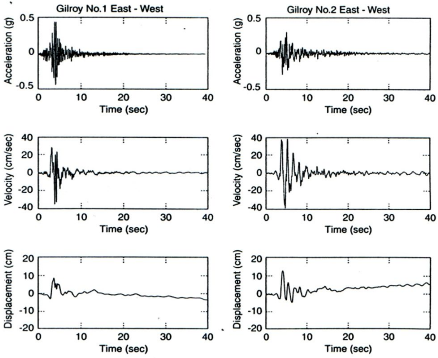


Figure 5: PGA, PGV, PGD (Santi Pailoplee, 2021)

7. **Attenuation** is decrease in wave size, or amplitude, away from source. When you throw a pebble in a pond, it makes waves on the surface that move out from the place where the pebble entered the water. The waves are largest where they are formed and get smaller as they move away.

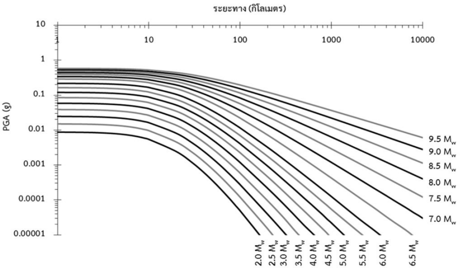


Figure 6: Attenuation (Santi Pailoplee, 2021)

8. **Ground motion prediction model (GMPEs)**, also called ground-motion models (GMMs) and attenuation relations, estimate the shaking (strong ground motion) that may occur at a site if an earthquake of a certain magnitude occurs at a nearby location.

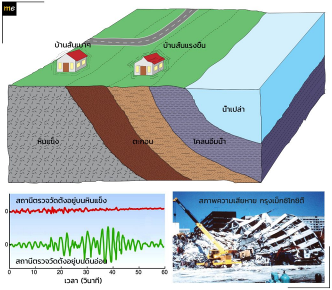
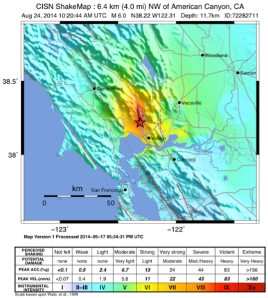


Figure 7: Ground Shaking (Santi Pailoplee, 2021)

9. **ShakeMap** The system is designed to combine instrumental measurements with information about **local seismic site conditions** and **the earthquake source** to estimate continuous shaking variations throughout a spatial area



PERCEIVED SHAKING	Not felt	Weak	Light	Moderate	Strong	Very strong	Severe	Violent	Extreme
POTENTIAL DAMAGE	none	none	none	Very light	Light	Moderate	Mod./Heavy	Heavy	Very Heavy
PEAK ACC.(%g)	<0.1	0.5	2.4	6.7	13	24	44	83	>156
PEAK VEL.(cm/s)	<0.07	0.4	1.9	5.8	11	22	43	83	>160
INSTRUMENTAL INTENSITY	I	II–III	IV	V	VI	VII	VIII	IX	X+

Figure 8: ShakeMap(USGS, 2018)

Review Papers

Why do we need machine learning?

1. Parameters
   1.1 Multilayer Perceptron
   1.2 Hybrid GMPE
2. Waveform
   2.1 Multilayer Perceptron
   2.2 Convolutional Neural Network
   2.3 Graph Neural Network
   2.4 Self Supervised Learning

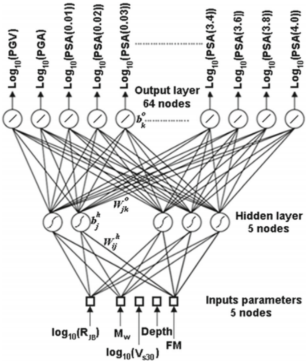


Fig. 4 Architecture of the ANN for *PGA*, *PGV* and *PSA*[0.01–4 s] prediction. The w_{ij}^h is the synaptic weight between the ith neuron of the input layer and the jth neuron in the hidden layer, b_j^h the bias of the jth neuron in the hidden layer. Also the W_{ij}^o is the synaptic weight between the jth neuron of the hidden layer and the kth neuron in the output layer, b_j^o the bias of the kth neuron in the output layer

Figure 9: PGA prediction with ANN (Derras, 2014)

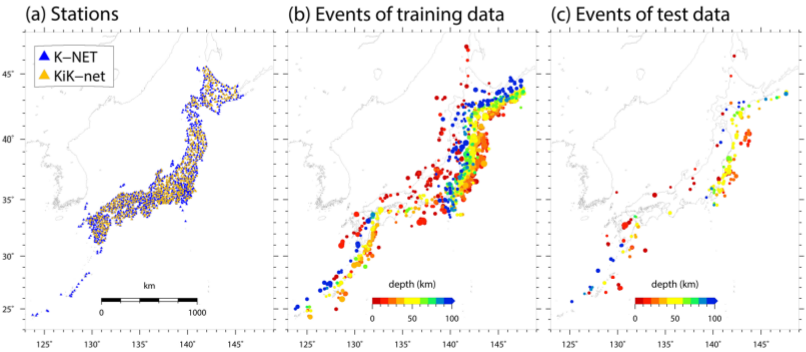


Figure 1. (**a**) Station distribution. Blue and yellow triangles indicate K-NET and KiK-net stations, respectively. (**b**) Spatial distribution of earthquake events in the training data. The circle color indicates the event depth. (**c**) Same as (**b**) but with test data.

Figure 10: Hybrid model(GMPE + ERT) (Hisahiko, 2020)

Convolutional Neural Network

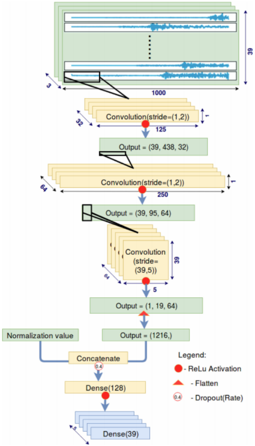


Figure 5. The architecture of the CNN model used. Boxes shaded yellow represent filter banks and operators, whereas boxes shaded green represent activations. Parenthesized vectors denote the dimensions of the outputs in question (height, width and depth).

Figure 11: CNN for rapid earthquake warning (Dario, 2020)

Graph Neural Network

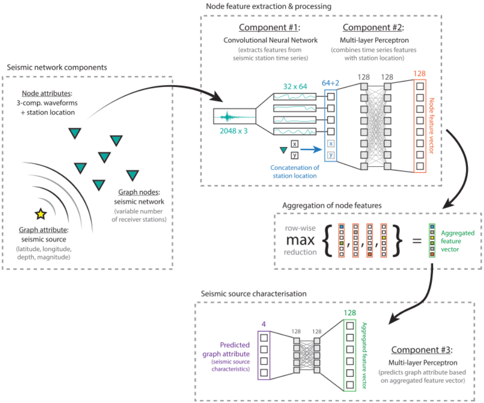


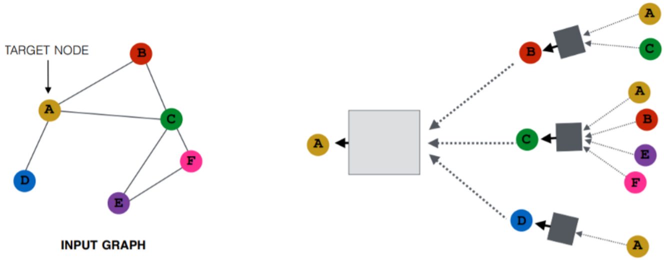
Figure 1. Synoptic overview of the adopted model architecture. The three-component waveforms from a receiver station are fed into a CNN, after which the extracted features are combined with the station's geographic location and further processed by an MLP. The resulting node feature vector of all the stations is aggregated, and this aggregated feature vector is passed through a second MLP that predicts the seismic source characteristics.

Figure 12: Characterize source of earthquake with GNN (M. P. A. van den Ende, 2020)

Focus on my work

1. Graph Neural Network
    - spatial model
    - temporal model
    - static graph, dynamic signal
2. Semi-Supervised Learning and Self-Supervised Learning
    - Unlabeled data
    - Imbalanced data

Graph Neural Network.



Graph definition

A graph $\mathcal{G}$ is defined as a tuple of a set of nodes/vertices V, and a set of edges /links E: $\mathcal{G} = (V, E)$. Each edge is a pair of two vertices, and represents a connection between them.

For instance, let's look at the following graph:

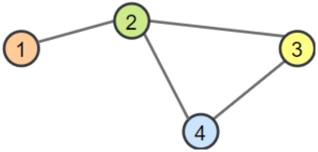


Figure 13: Example graph

The vertices are $V = \{1, 2, 3, 4\}$,

Definition of Adjacency Matrix

The **adjacency matrix** A is a square matrix whose elements indicate whether pairs of vertices are adjacent, i.e. connected, or not. In the simplest case, A_{ij} is 1 if there is a connection from node i to j, and otherwise 0.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

keep in mind that A is a symmetric matrix ($A_{ij} = A_{ji}$)

1. GCNs are similar to convolutions in images in the sense that the "filter" parameters are typically shared over all locations in the graph.
2. At the same time, GCNs rely on message passing methods, which means that vertices exchange information with the neighbors, and send "messages" to each other.

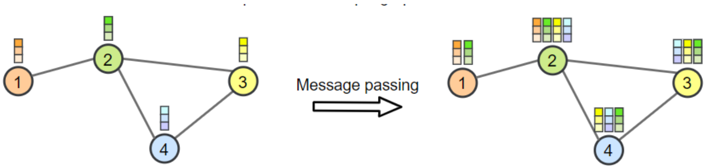


Figure 14: Message passing between nodes

Graph Convolutions

Given the previous features of nodes $H^{(l)}$, the GCN layer is defined as follows:

$$H^{(l+1)} = \sigma\left(\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}H^{(l)}W^{(l)}\right)$$

1. $W^{(l)}$ is the weight parameters with which we transform the input features into messages ($H^{(l)}W^{(l)}$).
2. To the adjacency matrix A we add the identity matrix so that each node sends its own message also to itself: $\hat{A} = A + I$.
3. $\hat{D}$ which is a diagonal matrix with D_{ii} denoting the number of neighbors node i has.
4. σ represents an arbitrary activation function.

Semi-Supervised Learning and Self-Supervised Learning.

Self Supervised Learning?

Contrastive self-supervised learning has outperformed supervised pretraining on many downstream tasks like segmentation and object detection.

What if we can get labels for free for unlabelled data and train unsupervised dataset in a supervised manner? We can achieve this by framing a supervised learning task in a special form to predict only a subset of information using the rest. In this way, all the information needed, both inputs and labels, has been provided. This is known as self-supervised learning.

How it work?

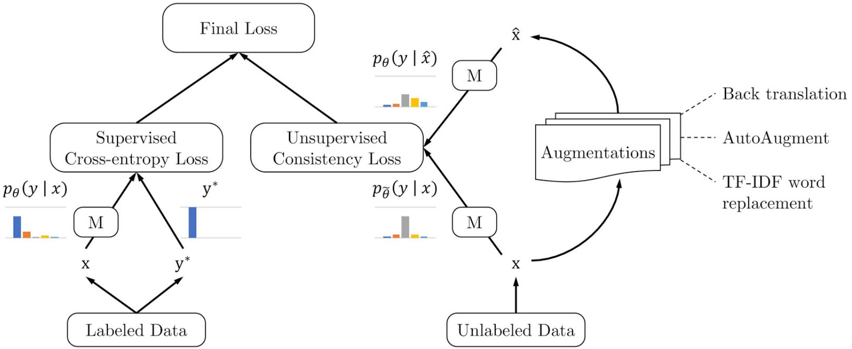


Figure 15: Semi and Self supervised learning (Qizhe Xie, 2019)

Data Augmentation(Image data)

1. Colorization
2. Autoencoder

Figure 16: change color tone (fast.ai, 2021)

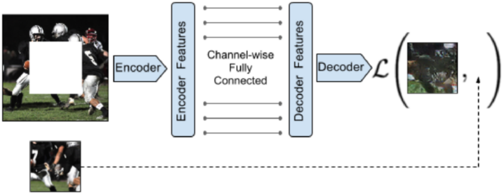


Figure 17: Autoencoder (Lilian Weng, 2019)

Fomulation

$$\mathcal{L}(\theta, \eta) = \mathbb{E}_{x,\mathcal{T}}\left[\|\mathcal{F}_\theta(\mathcal{T}(x)) - \eta_x\|_2^2\right]$$

$$\min_{\theta,\eta} \mathcal{L}(\theta, \eta)$$

1. $\mathcal{F}$ is a network parameterized by θ.
2. $\mathcal{T}$ is the augmentation.
3. x is an image.
4. The expectation $\mathbb{E}[\cdot]$ is over the distribution of images and augmentations. For the ease of analysis, here we use the mean squared error $\|\cdot\|_2^2$
5. η_x is the representation of the image x

Clustering Image with Self-Supervised Learning (Birds)

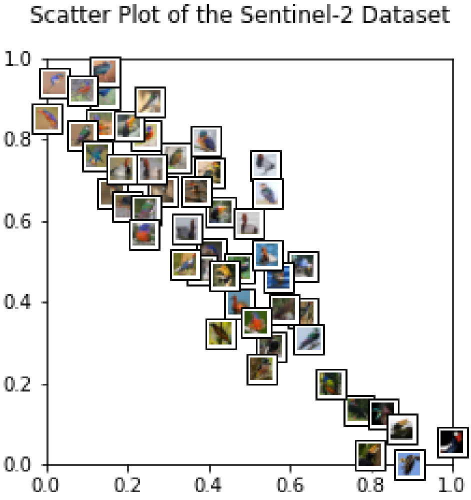


Figure 18: Embeding images on grid

Clustering Earthquake Source with Self-Supervised Learning

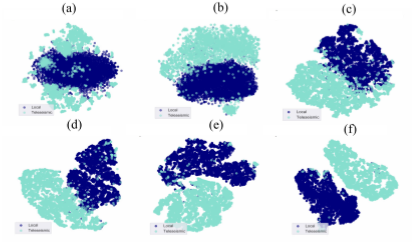


Fig. 3. (a) t-sne visualizations of local and teleseismic data in time domain, (b) time–frequency domain, (c) feature domain after the pretraining, (d) feature domain after 1350 iterations, (e) 1800 iterations, and (f) 5100 iterations during the fine-tuning. The accuracy of the *k*-mean clustering is given on top of each plot and samples are color-coded based on class labels. Note that as the method is trained, the two populations are increasingly divided into well-separated clusters.

Figure 19: self-supervised learning cluster local and teleseismic (S. Mostafa Mousavi, 2019)

Why Graph and Self-Supervised?

1. Resolve Imbalanced data problem

Table 1: Top-1 test errors (%) of ResNet-32 on long-tailed CIFAR-10 and SVHN. We compare SSL using 5x unlabeled data ($\mathcal{D}_U$ @5x) with corresponding supervised baselines. Imbalanced learning can be drastically improved with unlabeled data, which is consistent across different ρ_U and learning strategies.

(a) CIFAR-10-LT

Imbalance Ratio (ρ)	100				50				10			
$\mathcal{D}_U$ Imbalance Ratio (ρ_U)	1	$\rho/2$	ρ	2ρ	1	$\rho/2$	ρ	2ρ	1	$\rho/2$	ρ	2ρ
CE	29.64				25.19				13.61			
CE + $\mathcal{D}_U$ @5x	**17.48**	18.42	18.74	20.06	**16.79**	16.88	18.36	19.94	**10.22**	10.48	10.86	11.04
LDAM-DRW [7]	22.97				19.06				11.84			
LDAM-DRW + $\mathcal{D}_U$ @5x	**14.96**	15.18	15.33	15.55	**14.33**	14.70	14.93	15.24	8.72	**8.24**	8.68	8.97

(b) SVHN-LT

Imbalance Ratio (ρ)	100				50				10			
$\mathcal{D}_U$ Imbalance Ratio (ρ_U)	1	$\rho/2$	ρ	2ρ	1	$\rho/2$	ρ	2ρ	1	$\rho/2$	ρ	2ρ
CE	19.98				17.50				11.46			
CE + $\mathcal{D}_U$ @5x	**13.02**	13.73	14.65	15.04	**13.07**	13.36	13.16	14.54	**10.01**	10.20	10.06	10.71
LDAM-DRW [7]	16.66				14.59				10.27			
LDAM-DRW + $\mathcal{D}_U$ @5x	**11.32**	11.70	11.92	12.78	**10.98**	11.14	11.26	11.51	8.94	9.08	**8.70**	9.35

Figure 20: SSL and imbalanced data (Yuzhe Yang, 2020)

2. Self-supervised Training of Graph Convolutional Networks

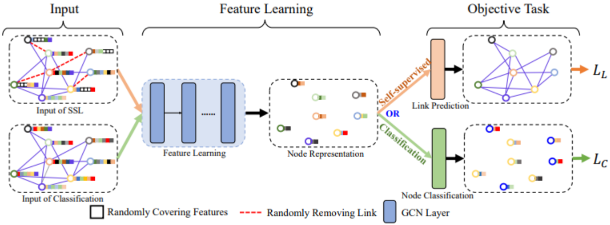


Fig. 1. Overview of the proposed self-supervised training of graph convolutional networks.

Figure 21: Self-supervised Training of Graph Convolutional Networks model (Qikui Zhu, 2020)

Tool & Data

1. Represent Temporal and Spatial data

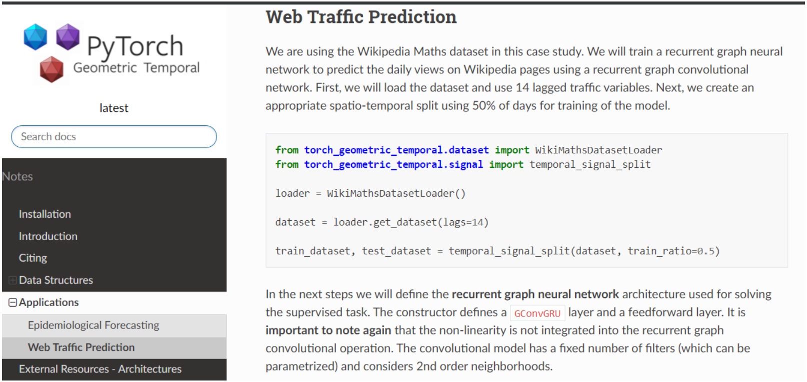


Figure 22: web traffic prediction (GNN application example)

1. Retrieving Data from Data Centers

ObsPy   GitHub   Documentation   Mailing Lists   Developer Resources   Search Docs

ObsPy Documentation (1.2.0) / ObsPy Tutorial   index | modules | next | previous

5. Retrieving Data from Data Centers

This section is intended as a small guide to help you choose the best way to download data for a given purpose using ObsPy. Keep in mind that data centers and web services are constantly changing so this recommendation might not be valid anymore at the time you read this. For actual code examples, please see the documentation of the various modules.

Note

The most common use case is likely to download waveforms and event/station meta information. In almost all cases you will want to use the `obspy.clients.fdsn` module for this. It supports the largest number of data centers and uses the most modern data formats. There are still a number of reasons to choose a different module but please make sure you have one.

5.1. The FDSN Web Services

5.1.1. Basic FDSN Web Services

Available Data Types	Format
Waveforms	MiniSEED and optionally others
Station Information	StationXML and Text
Event Information	QuakeML and Text

Note

Not all data providers offer all three data types. Many offer only one or two.

If you want to requests waveforms, or station/event meta information **you will most likely want to use the** `obspy.clients.fdsn` **module**. It is able to request data from any data

Figure 23: Load Data with Obspy

Q/A