An Automatic P Phase Picking Toolbox Using Deep Learning Method

Chun-Ming Huang*[1], Hao Kuo-Chen[1], Pei-Yu Zhong[1], Zhuo-Kang Guan[1]

[1]Department of Earth Sciences, National Central University, Zhongli, Taiwan
*Address: Dep. Earth Science, No. 300, Zhongda Rd., Zhongli District, Taoyuan City 32001, Taiwan
*Email: jimmy60504@gmail.com, Phone: +886-986040820

Photography encouraged

Please star us on Github!!

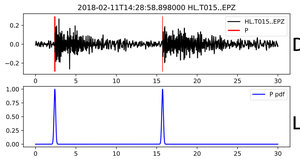
Dataset

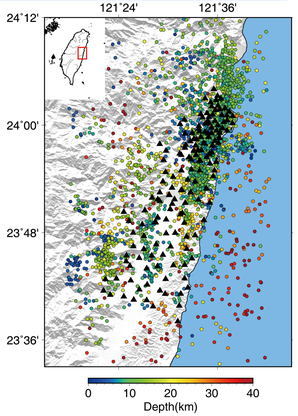
Hualien, Taiwan
20170120-20170505
20180209-20180218

170 Stations, 3,474 Events, 85,684 P phase picks

Training data:
- Z component
- Time window: 30 sec
- Sample rate: 100 Hz

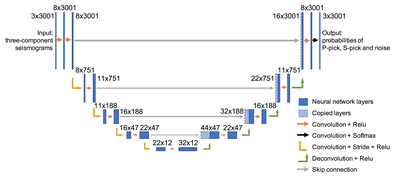
Training Label:
- P phase picks
- Normal distribution mask
- Standard deviation: 100 ms





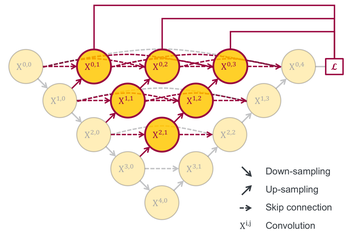
Station and event locations

Model



Modified U-Net[1] from PhaseNet[2]

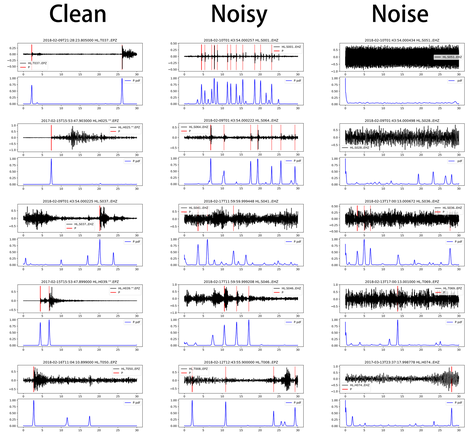
The U-net is originally designed for medical image semantic segmentation, using multiple upsampling layers and adding skip pathways in the fully convolutional net (FCN) to predict a better segmentation map.
We use one channel only instead of three from the PhaseNet.



Unet++[3] Architecture

The difference between the original U-net and UNet++ is the re-design skip pathways, which close the gap between encoder and decoder. It makes the training procedure converges faster.

Results



Summary

We develope an efficent toolbox to manipulate datasets, which let users focus on the training process.

We are able to get similar result from the PhaseNet only using Z component.

The code is open-sourced and the QR code is on the top. https://github.com/jimmy60504/SeisNN

Reference

Paper:

1. Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.

2. Zhu, W., & Beroza, G. C. (2018). PhaseNet: a deep-neural-network-based seismic arrival-time picking method. Geophysical Journal International, 216(1), 261-273.

3. Zhou, Z., Siddiquee, M. M. R., Tajbakhsh, N., & Liang, J. (2018). Unet++: A nested u-net architecture for medical image segmentation. In Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support (pp. 3-11). Springer, Cham.

Code:

a. Obspy https://github.com/obspy/obspy
b. U-net & UNet++ model https://github.com/MrGiovanni/UNetPlusPlus

Preprocessing

```
signal_processing:
    demean
    detrend
    normalized
    resamp(100)
```

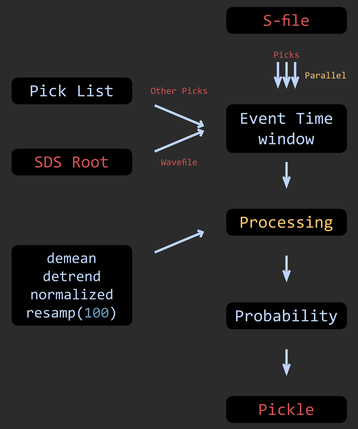
Training datasets are generated from the SEISAN S-file, the start time will be the event origin time.

It's possible to have multiple events in 30 sec window. Pick list will keep all the picks for searching.

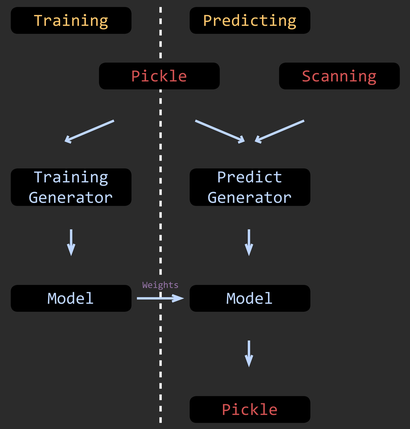
Pickle format (.pkl) can perserve extra attribuites which are not originally in the Obspy trace object.

Using multi-thread pooling for faster I/O from the hard disk drive.

Preprocessing, training and predicting can be run seperately.



Training & Predicting




```
training_parameter:
    Epoch = 1
    Batch_Size = 2
    Optimizer = Adam(lr=1e-4)
    Loss_Function = binary_crossentropy
```

Tensorflow and Pytorch are supported. Docker image and Dockerfile are avaliable.

The Generator only reads one batch at a time, reduce the RAM usage for larger dataset.

Scanning will scan through all avaliable stations in the given time window.

Tensorflow 2.0 Docker Image

Pytorch 1.0 Docker Image

Discussion

```
test_bench:
    CPU = i7-6850K @ 3.60 GHz x 12
    GPU = GTX 1080 Ti FE
    Memory = DDR4-2400 16 GB x 2
    Storage = 1 TB NVMe SSD + 4 TB HDD
```

All the results are from the random scanning datasets, which are unseen in the training process.

In the training procedure, smaller batch size tends to converge faster than the bigger ones. Batch size 2 takes approximately 40 mins to train about 90,000 datasets, one epoch on a single GPU.

Pre-training the model using 1,000 datasets will make the model converge faster.

Some of the picks are triggered by the surface wave can cause false alarms. We need to investigate the convolution kernel to have a better interpretation.

Future Work

```
features:
    Picks quality control
    Picks combine into an event
    S-file output

others:
    Unittest
    Github Pages for documentation
```

Feedback

If you have any comments or questions, please fell free to leave an issue on our github repository.

We appreciate your precious advice.