Seismology - Summer Internship 2022 at GFZ Potsdam

Moshe Beutel

September 7, 2022

1 Overview

Seismic earthquake research is full of annotated data. Open data sets from seismographic stations contains millions of manually annotated interesting events and is open to seismographic research community. Efficient deep learning models developed in the last 4 years to deal with the following main tasks: *Earthquake detection, Phase Identification* and *Onset Time Picking* which correspond to the general algorithmic tasks of: Detection, Classification and estimation respectively.

2 Tasks

The tasks we are examining are the following:

Task	Input	Output	Metric
Event Detection	30s window of seismic waveform	Contains First Arrival	AUC (or F1)
Phase Identification	10s window of seismic waveform	Detrmine P or S	MCC
Onset Time Picking	10s window contains one S or P wave (known)	Determine Onset Time	RMSE

3 Terminology

- **P and S Waves** - Short for primary ad secondary waves. *Body Waves* are energy travelling through solid volumes and *Surface Waves* travel through free surfaces. The Body waves travel faster hence called primary or P Waves and the - slower - surface waves are called Secondary or S Waves.

- **Arrival Time** - The time of first discernible motion of a seismic phase.
- **Picking** - Measuring (Estimating ???) the arrival time

4 Models

The following models were tested in the benchmark:

- BasicPhaseAE (Woollam et al., 2019)
- CNN-RNN Earthquake Detector (CRED; Mousavi, Zhu, et al., 2019)
- DeepPhasePick(DPP; Soto & Schurr, 2021)
- Earthquake transformer (EQTransformer; Mousavi et al., 2020)
- PhaseNet (Zhu & Beroza, 2019)

5 Limitation

The noted models, although preformed well on the given tasks using the defined metrics, are still limited in the view of real life applications like early warning scenarios.

- **Datasets Limitations** -
  - Uncertainties and nonuniqueness of manual labels owing to limited resolution, presence of noise, **different levels of expertise**, cognitive biases, and inherent ambiguity of tasks is a limiting factor. In image object classification for example tasks this is generally non-issue because normally most annotators would agree about pictures of cars,cats,tables and other daily life objects.
  - Not all seismic signals classes and typical noise are covered in the datasets - e.g. data from nodal seismometers at local distances, mine blasts, or volcanic signals.
- **Tasks Limitation** - The tasks defined above does not exactly represent real life scenarios where:
  - There are no defined time windows
  - More than one event may occur in a given time frame
  - The metrics defined does not take into account how early the tested algorithms would be able to identify an event onset
  - In continous time setup the false positive rate needs to be significantly lower than in post-processing.
- Transfer Learning - It is yet unclear which datasets are most suitable for pretraining models

6 Detailed Actual Benchmark Protocol

The following section describes the benchmark protocol as described in "Which picker ..." (- I will call it "the paper " in this section) and implemented in code of the pick-benchmark GitHub Repo. the repo is based on the Seibench package and pytorch-lightning.

6.1 Training

The training used the Seisbench's data generation module for building training pipelines.

Window Length - the paper states that with probability of 2/3 a window with exactly a single pick is selected and with probability of 1/3 a random unchanged window is selected (that can also be a single pick window so the actual probability of a single pick window is more than 2/3). The paper explicily states that "*The windows selected for the tasks are identical across all models and their length is independent of the specific model.*". Looking at the code at benchmark/models.py it seems that window lengths are different e.g. EQTransformer gets a window of size 12000 samples and PhaseNet gets a window of size 6000 samples.

Preprocessing -

EQTransformer - Defines block 1 and block. At train time there is another block of preprocessing between.

CRED preprocessing includes resampling normalization and stft on top of the windowing.

hyperparameters - How were they chosen. different by models ???

7 Research Question Formulation

The research question subject for this internship term deals with quantifying the uncertainty of a given model performance. The 6-week time frame is of course not suitable to solve that big problem so the exact task is yet to define.

7.1 Uncertainty Issues

The following are different sources or symptoms of uncertainty that rises during benchmarking the pickers:

- The variance (???) between human labeling is about 0.2 seconds
- Different window limits for the same recording results with different picking (Had the training protocol contained overlapping windows ???)
- Noisy environments - lower SNRs - are expected to perform worse is it always what is happening?

- Different seismic machine learning models present different probability levels to describe the same certainty

7.2 Proposed Naive Checks

This section describes some preliminary tests and actions that intuitively arise from the described issues

- Normalize output probabilities of the different machine learning models (Guy's work ???).
- Gradually add (Gaussian) noise to existing datasets untill model predictions are worthless. Analyze different model probabilities as a function of the noise level added (Joachim's offer). This test will also output model robustness to noise.
- If overlapping windows were not included in the original benchmark try it.
- The previous 2 points are just data augmentation that is very often applied in Image Detection and Classification training protocols. The Seisbench generator module contains augmention module that implements the following augmentation functions:
  - Normalizing - demeaning, detrending and amplitude normalization (in this order).
  - Filter - based on scipy.signal.butter
  - FilterKeys - Filter out features that are not needed for training ???
  - ChangeDtype - Cast data type to a given desired type.
  - OneOf - a meta-augmentation that runs a single augmentation according to a given probabilities list w.r.t an augmentation list.
  - NullAugmentation - NoOp for OneOf
  - ChannelDropout - Zeroes out 0 to c-1 channels randomly. Scales up the remaining channels s.t. overall energy remains unchaged.

  Consider implementing data augmentation module to Seisbech that will transform original datasets in memory in the torchvision manner and preform:

  - Cropping in time
  - Amplitude scaling
  - Frequency scaling
  - Filtering
- Add noise check to all tasks - Event Detection, phase identification and onset time picking

8 TODOs

- **Understand benchmark protocol using pick-benchmark repo - window size, augmentations specifically overlapping windows and filters, output thresholds ...** Most important for now need to check whether different window sizes were used for different models and generally whether different augmentations were used.

- Bar.py contains implementation ??? of the "Baer-Kradolfer picker (Baer & Kradolfer, 1987), as baseline. The Baer-Kradolfer picker depends on four parameters: a minimum required time to declare an event, a maximum time allowed below a threshold for event detection, and two thresholds. For details on the parameters, we refer to (Baer & Kradolfer, 1987) or (Kueperkoch et al., 2012)." from which picker paper

- What is the role of bayesian-optimization package in pick-benchmark repo if GPs are needed use gpytorch

- GPs for distribution learning

- Generative models for distribution learning