Machine Learning Engineer Nanodegree

Capstone Proposal

Jan Nowakowski
January 12th, 2019

Proposal

Domain Background

(approx. 1-2 paragraphs)

Forecasting natural disasters is a critical part of the field of earth science, due to the impact of early alarming on decreasing the toll of such a catastrophe, both with regards to human life, as well as economic loss. One of the most deadly, dangerous and omnipresent disasters are earthquakes. Therefore, I chose to participate in the Kaggle competition that strives to improve the forecasting of when an earthquake will occur, based on real-time seismic data.

The Kaggle competition can be viewed here: https://www.kaggle.com/c/LANL-Earthquake-Prediction

Problem Statement

(approx. 1 paragraph)

The goal of the competition is to use seismic signals in order to predict when an earthquake will occur.

Datasets and Inputs

(approx. 2-3 paragraphs)

The datasets and inputs of this Kaggle competition are described here: https://www.kaggle.com/c/LANL-Earthquake-Prediction/data

To summarize, the train dataset is a single, continuous training segment of experimental data and the test dataset is a set of many small segments of a longer experiment.

The training data contains two fields:

- acoustic_data, which is the seismic signal
- time_to_failure, which is the time in seconds until the next failure (i.e. Earthquake)

The test data has only the first of the two fields.

All of the data used in this competition has been generated by a laboratory setup described in [1].

Solution Statement

(approx. 1 paragraph)

The solution is to predict the time to failure from the last row of the test segment to the next laboratory earthquake.

Benchmark Model

(approximately 1-2 paragraphs)

A basic feature benchmark has been given in the Kaggle competition, and can be viewed here:

https://www.kaggle.com/inversion/basic-feature-benchmark

To summarize, it uses the mean, standard deviation, min and max of the data to calculate the time_to_failure.

Evaluation Metrics

(approx. 1-2 paragraphs)

The models are evaluated using the mean absolute error between the predicted and real time before the next lab earthquake.

Project Design

(approx. 1 page)

In the first step, I will sample and cut the a part of the long train datafile in order to have many segments, with which I will be able to validate and test my model (hence the test segments provided do not contain the time_to_failure that needs to be calculated). I plan to cut in in segments of varying (pseudo-random) lengths in order to not bias my model towards fixed-length datasets.

One simple (i.e. a second benchmark) approach I will try will be to use a Convolutional Neural Network (CNN) to predict the time to failure. I will have to, however, be careful to design it such, that it will be able to take inputs of varying lengths. I do not plan to focus on for a long time, but I will play with different model architectures and hyperparameters in order to test if this approach can at all work. If I find a good, fitting pre-trained model, I will also use transfer learning.

In the second approach, that sounds the most promising for achieving a good result, I will use a cannonical way of treating time-series data, namely using a Recurrent Neural Network. It will be a first project of mine using this kind of NN, so there will definitely be a learning curve involved. It has, however, been shown to be succesfull in many different applications including, but not limited to Natural Language Processing [2], financial forecasting [3] and anomaly detection [4].

Literature:

[1] J. R. Leeman et al., *Laboratory observations of slow earthquakes and the spectrum of tectonic fault slip modes*, Nat. Commun. 7, 11104 (2016). (https://www.nature.com/articles/ncomms11104)
[2] https://medium.com/explore-artificial-intelligence/an-introduction-to-recurrent-neural-networks-72c97bf0912
[3] https://www.math.kth.se/matstat/seminarier/reports/M-exjobb18/180608m.pdf
[4] https://blog.statsbot.co/time-series-anomaly-detection-algorithms-1cef5519aef2