Machine Learning Engineer Nanodegree

Capstone Proposal

Lalit Yadav May 7th, 2019

Proposal

Domain Background

Time Series is one of the most challenging probelms in the field of Machine Learning.In this project, we will use machine learning algorithm to forecast the future web traffic for approximately 145,000 Wikipedia articles. The field of time series encapsulates many different problems, ranging from analysis and inference to classification and forecast. Predicting this type of future traffic for an web page will also help in time series prediction in other domains.

The motivation to work on this project is to get an understanding the different factors responsible for traffic on a website and if understanding these features will help us in predicting future traffic of any websites.

Problem Statement

Sequential or temporal observations emerge in many key real-world problems, ranging from biological data, financial markets, weather forecasting, to audio and video processing. The field of time series encapsulates many different problems, ranging from analysis and inference to classification and forecast.

This challenge is about predicting the future behaviour of time series' that describe the web traffic for Wikipedia articles. The data contains about 145k time series and comes in two separate files: train_1.csv holds the traffic data, where each column is a date and each row is an article, and key_1.csv contains a mapping between page names and a unique ID column (to be used in the submission file).

Datasets and Inputs

The datasets are provided by Google on Kaggle competition website.

The training dataset consists of approximately 145k time series. Each of these time series represent a number of daily views of a different Wikipedia article, starting from July, 1st, 2015 up until December 31st, 2016. For each time series, we are provided the name of the article as well as the type of traffic that this time series represent (all, mobile, desktop, spider). We may need to use this metadata and any other publicly available data to make predictions. For public data I think googletrend which gives trend of certain google keywords over time will help with prediction.

Solution Statement

I am planning to use a deep learning algorithm with RNN GRU/ LSTM architecture. Recurrent Neural Networks—Long short-term memory (LSTM), Gated Recurrent Unit (GRU) are good for time series forecasting comprised of separate autoencoder and decoder sub-models. The skill of the proposed LSTM architecture at rare event demand forecasting and the ability to reuse the trained model on unrelated forecasting problems.

Benchmark Model

For a benchmark model we will be using tha Kaggle leader board to check our model's SMAPE score to the others on the leaderboard. A good model should have a SMAPE score of approximately 35-40.

Evaluation Metrics

The Evaluation Metrics used for this Kaggle competition is SMAPE(Symmetric mean absolute percentage error (SMAPE or sMAPE)) SAMPE is an accuracy based on percentage (or relative) errors. It is defined as:-

```
1/n summation over 1 to n (|Ft - At| / (|At+ + |Ft|) / 2)
```

where At is the actual value and Ft is the forecast value.

The absolute difference between At and Ft is divided by half the sum of absolute values of the actual value At and the forecast value Ft. The value of this calculation is summed for every fitted point t and divided again by the number of fitted points n.

Project Design

1. First step would be to import the training data, then analyzing and cleanning the data.
2. We will have to add some features that we help the model in predictions. Features like Days, Months, Years, language of text, country are interesting to forecast with a Machine Learning Approach or to do an analysis.
3. We will create a encoder decoder model to fit the data. One conern that I ca see is that on longer sequences LSTM/GRU works, but can gradually forget information from the oldest items. We will use try to add attention to the model for effectiveness.
4. As the total datasize is around 145k, will will take subsample of the original data and train the model. Once the model seems to generalize well on small subset of data we will try to use all data for prediction.

References

- Kaggle
  - https://www.kaggle.com/c/web-traffic-time-series-forecasting/discussion
- Kernels
  - https://www.kaggle.com/c/web-traffic-time-series-forecasting/discussion/43795#latest-525730
  - https://www.kaggle.com/headsortails/wiki-traffic-forecast-exploration-wtf-eda
  - https://www.kaggle.com/muonneutrino/wikipedia-traffic-data-exploration
- SMAPE
  - https://en.wikipedia.org/wiki/Symmetric_mean_absolute_percentage_error