Machine Learning Engineer Nanodegree

Capstone Proposal

Jeremy Jordan
August 14th, 2017

Proposal

The following proposal discusses a planned implementation for leveraging unstructured text data in time series forecasting. Specifically, this project will attempt to learn correlations between the history of companies' quarterly shareholder earnings call transcripts with future stock price performance.

Domain Background

The field of quantitative finance began in the 20th century and has since grown to be a critical part of many firms' investment strategy. It is widely accepted that market activity is largely stochastic with a very low signal-to-noise ratio, presenting a unique problem in developing reliable and accurate forecasts.

Autoregressive integrated moving average (ARIMA) models are a popular class of techniques to perform univariate forecasts based on historical performance. These models consist of three components:

- **Autoregression (AR)** refers to a model which combines (1) a linear dependence of an observation on previous values with (2) a stochastic term for predicting future values.
- **Integrated (I)** makes use of differencing observations in order to enforce stationarity of the data. This differencing may occur multiple times before the data becomes stationary.
- The **moving-average model (MA)** models the dependency between an observation and residual error of previous values.

In the machine learning community, recurrent neural networks (RNNs) have gained in popularity for use in sequential predictions where the output is dependent on previous output values. There have been attempts to leverage long short-term memory (LSTM) networks, a popular type of RNNs, for use in financial forecasting. However, forecasting is difficult due to the low signal-to-noise ratio previously mentioned.

Recently, another approach was proposed known as a Significance-Offset Convolutional Neural Network (SO-CNN). The authors of the paper proposing this technique present SO-CNNs as "a deep convolutional network architecture for regression of multivariate asynchronous time series." Essentially, this architecture is similar to an autoregressive model, where the weighting scheme (for discounting previous values of the forecasted variable) is developed using a convolutional neural network.

Problem Statement

This project will attempt to develop a machine learning model for identifying features within text data that provide predictive power for forecasting price performance in the stock market. This presents a unique challenge given the level of stochastic movement in the market. Realistically, market performance depends on a variety of factors including competitor performance (ex. Blue Apron stock drops when Amazon announces a competing service), macroeconomic trends (consumer discretionary spending is generally higher during periods of economic prosperity), market confidence in the company's leadership, stochastic movement from emotional players in the market, and more. Further, the quality of information regarding these factors varies widely, from official company statements mandated by the SEC to speculation to activity on social media sites such as Twitter. The scope of this project is limited to extracting useful signal from a official company statements during quarterly earnings calls, which may be regarded as high-quality information. However, it would be beneficial to design an architecture which allows for additional sources of information to be added to the model in a modular manner.

Datasets and Inputs

The data for this project will originate from two sources: earnings call transcripts from Seeking Alpha, and stock price performance from Quandl.

Earnings call transcripts will be scraped from Seeking Alpha's website using a Python library, Scrapy. These transcripts will be preprocessed to remove punctuation, replace numerical quantities with a flag ($4 billion --> #monetaryamount), change all characters to lower case, and remove stopwords. Then, these statements will be converted to word embeddings - two dimensions will describe the word and a third dimension will correspond with the frequency of the word in a given document. In a sense, this will transform a body of text into a "picture" representation.

Daily stock prices will be retrieved from Quandl using an API call. It is possible that this information will need to be preprocessed to reduce the noise level. A recent paper reports the use of wavelet transforms to denoise the price data.

The two datasets exist on different timescales; company earnings call transcripts only occur four times a year, whereas we have access to the daily price performance of a company. This will require experimentation to determine the best way to structure a prediction. Generally, one could expand the text data to be constant input, in addition to the daily changing price, until new information arrives via the next quarterly earnings call. Another approach would be to develop a way to wrap up the daily price performance into a quarterly summary and simply provide a quarter-by-quarter forecast of price performance.

For the scope of this project, data from 200 companies will be collected over a 6 year period. This will provide 4,800 earnings call transcripts for the model to learn from. If performance is subpar and learning curves suggest that more data is needed to accurately learn from the transcripts, the early layers of the convolutional network will be pre-trained on a large corpus of text to develop general-feature extractors.

Solution Statement

Foremost, this project will attempt to create a recurrent neural network architecture that has strong predictive capabilities to generate reliable and accurate forecasts. This neural network will take both price history and company statements as input. Company statements will be translated into word embeddings that a convolutional neural network can learn relevant features from. The goal for this project is to outperform a standard ARIMA model.

If the model performance is unsatisfactory, I will attempt to incorporate some of the autoregressive techniques as mentioned in the Significance-Offset CNN paper.

Benchmark Model

This project will compare the predictive performance of three models: a univariate ARIMA model, a univariate LSTM model, and a multivariate LSTM model which combines previous price performance with company's quarterly earnings call transcripts. These statements are a combination of reflection of the previous quarter in addition to forward-looking statements regarding the company's future.

A comparison between the univariate LSTM model which makes predictions solely on price performance and the multivariate LSTM model which incorporates the quarterly earnings statements will reveal whether or not the text data was a good indicator of future performance. Further, a comparison between the multivariate LSTM and ARIMA model will reveal whether the proposed model offers any improvement over the currently adopted standard.

All models will be compared against the evaluation criteria discussed in the following section.

Evaluation Metrics

Models will be evaluated primarily according to the mean absolute percentage error of the true price performance, as defined below. This metric was chosen due to its scale invariance, as companies have a wide range of market capitalization and price performance does not occur on a standard scale. Forecasting models which have a strong predictive capability will have a lower mean absolute percentage error.

The mean absolute percentage error can be calculated as

$$\mathrm{M} = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|,$$

where A_t represents the true value at time t and F_t represents the forecasted value at time t.

Further, I will also plot the mean absolute percentage error as a function of steps predicted into the future to gauge how far into the future the model is capable of accurately forecasting.

Project Design

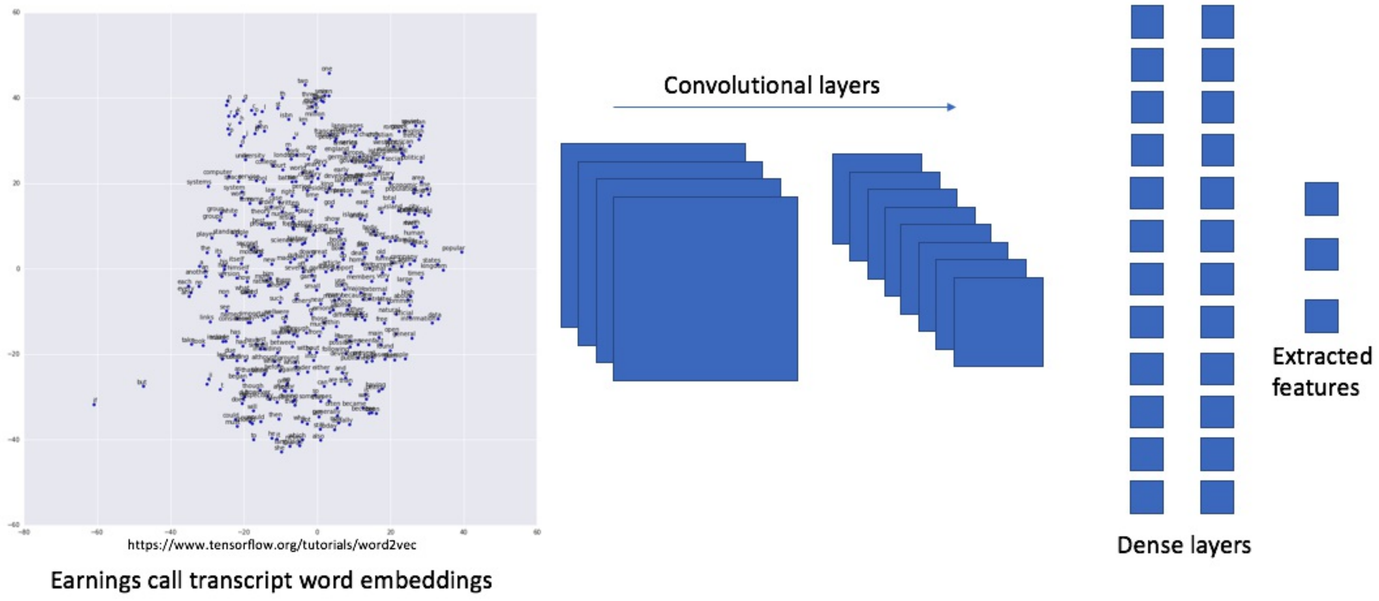
(approx. 1 page)

In this final section, summarize a theoretical workflow for approaching a solution given the problem. Provide thorough discussion for what strategies you may consider employing, what analysis of the data might be required before being used, or which algorithms will be considered for your implementation. The workflow and discussion that you provide should align with the qualities of the previous sections. Additionally, you are encouraged to include small visualizations, pseudocode, or diagrams to aid in describing the project design, but it is not required. The discussion should clearly outline your intended workflow of the capstone project.

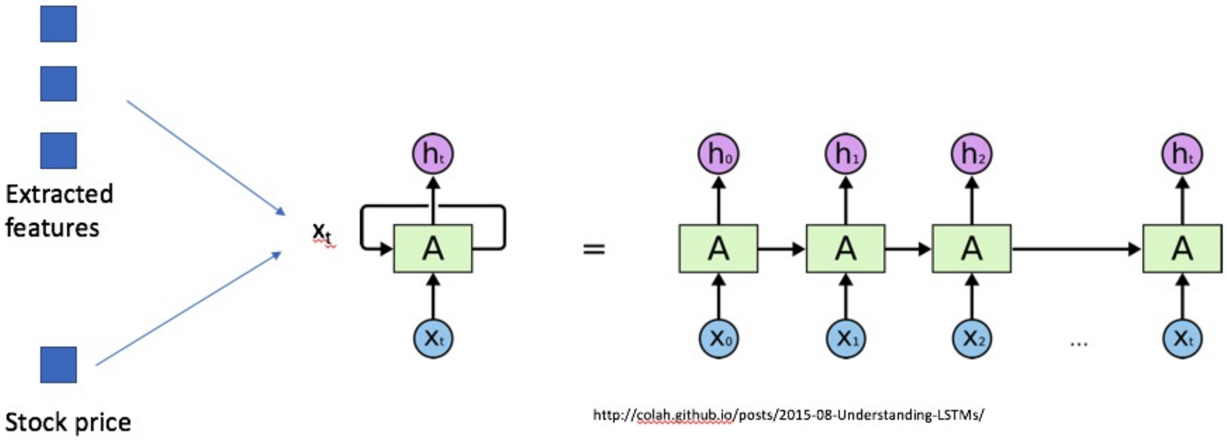
Company earnings call transcripts will be scraped from Seeking Alpha and transformed into word embeddings. These word embeddings will be a two-dimensional vector representation of the words that preserve semantic relationships. This two-dimensional spatial data will be binned into "pixels" where the pixel intensity (ie. a third dimension) will correspond with the frequency the words within a pixel appear in the document.

One anticipated problem is that the company earnings statements are too similar to each other to learn useful features. It might be necessary to use a TF-IDF score, instead of a simple frequency, in the third dimension to draw a more distinct separation between statements.

These word embedding "pictures" will be fed into a stack of convolutional layers which can learn which features from the body of text are important signals for making future predictions on price performance.


Convolutional layers
Dense layers
Extracted features
https://www.tensorflow.org/tutorials/word2vec
Earnings call transcript word embeddings

The extracted features from the earnings call transcripts will be combined with daily stock price closing prices to develop a forecasting model for future performance. A long short-term memory network will be used to perform recurrent one-step predictions.



This network should be able to identify long-term trends across multiple earnings report statements to identify macroscopic features, such as a change in the company's narrative over time.

The time series data will be split into training, validation, and test sets. This will be a 60-20-20 split, and segmentation will occur by company. In other words, 60% of the companies will be assigned to the training set, 20% of the companies will be reserved for validation, and 20% of the companies will be used in testing and evaluating the model. Companies will be randomly assigned to their respective sets.

References:

- Autoregressive Convolutional Neural Networks for Asynchronous Time Series
- Evaluating forecasting models
- A deep learning framework for financial time series using stacked autoencoders and long short-term memory
- MIT 18.S096 Topics in Mathematics with Applications in Finance - Time Series Analysis

Other resources:

- https://www.aclweb.org/anthology/C/C16/C16-1229.pdf
- https://forecasters.org/pdfs/IEEE_TNN_2007.pdf
- http://cs229.stanford.edu/proj2009/LvDuZhai.pdf
- http://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/
- http://machinelearningmastery.com/make-sample-forecasts-arima-python/
- http://www.statsmodels.org/dev/generated/statsmodels.tsa.arima_model.ARIMA.html
- http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- http://cslt.riit.tsinghua.edu.cn/mediawiki/images/5/5f/Dtq.pdf
- https://onlinecourses.science.psu.edu/stat510/node/48
- https://github.com/rouseguy/TimeSeriesAnalysiswithPython
- https://www.youtube.com/watch?v=tJ-O3hk1vRw