TRIMODAL LEARNING FOR STOCK PREDICTION

SARTHAK RASTOGI

CSE 6DSML | 2K19CSUN04024

INTRODUCTION

Multimodal learning is a technique that integrates data and signals in multiple modes to learn from. In this project I use data on AAPL stocks in three different modes -- time series, text, and tabular -- and use them to predict future prices. These three modes are represented by historical prices, news headlines and tweets about Apple and the AAPL stock, and Apple's financial statements.

OBJECTIVE

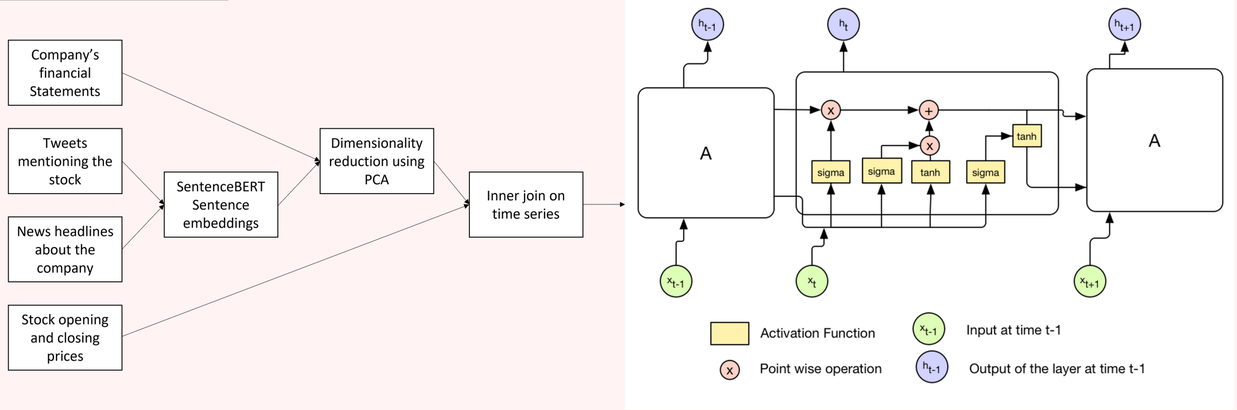
The aim is to formulate an efficient and effective technique to utilise the data available on stocks in many different modes, and train an LSTM model that can predict how the stock is going to perform in the near future.

DATASET

The dataset consists of the following information on the AAPL stock in between the years 2012 - 2020:

1. Yearly financial statements of Apple obtained from ROIC.AI. These contain features such as revenue, gross profit, R&D expenditure, operating expenses, EBITDA, net income, inventory, total assets and liabilities, etc.
2. News headlines and main points of articles from on Apple Inc. taken from Investing.com.
3. Tweets mentioning AAPL and Apple along with their timestamps, scraped from Twitter using the Python tweetscraper library.
4. Daily time series data on the opening and closing prices, the daily highs and lows of AAPL stock taken from Yahoo! Finance.

METHOD



I used SentenceBERT to generate embeddings for the news headlines and tweets. The resulting embeddings have 768 dimensions which I lowered down to 32 by using PCA, without suffering too much loss in variance. Since there were multiple headlines for every date, I averaged out all the embeddings for a date because I only needed the sentiment behind what happened to the stock on that day. I repeated the same for the tweets.

The financial statements contained over a hundred features; I dropped any features which had missing values and then used PCA to reduce the dimensionality of the resulting data because a lot of the features were correlated with each other. I also normalised the result by min-max scaling.

I then performed an inner join on the date feature, in between the four tables. The timestamps of each table varied in granularity: the tweets had timestamps down to the second, and the news headlines and stock prices had corresponding dates, while the financial statements were only available on an annual basis. In my final dataset I used a daily scale for the time series, repeating the financial statement features for entire years. Finally, I trained an LSTM on the resulting dataset.

RESULTS

The LSTM finally resulted in an RMSE of 1.63 on the training set and 7.88 on the test set. The SentenceBERT is the step that takes the longest to compute, while the remaining steps run fairly quickly on inference time. Provided that the news headlines and tweets have already been scraped, the resulting model can make a prediction in under a minute.