Machine Learning Engineer Nanodegree

STOCK PRICE PREDICTION

Mar 1st, 2019

Proposal

The project aims to build a machine learning model that can predict stock price by using historical time-series data.

Domain Background

All investment firms, hedge funds and even individuals have been using financial models to better understand market behavior and make profitable investments and trades. A wealth of information is available in the form of historical stock prices and company performance data, suitable for machine learning algorithms to process.

Stock price prediction is a very vital research area throughout many past decades. Using statistical methods and stochastic analysis to make stock price prediction is the mainstream in last 20 years, while using machine learning techniques to predict stock price is becoming more and more prevalent in recent years. The prevailing theories is that stock prices are totally random and unpredictable but that raises the question why top investment firms are continuously hiring quantitative analysts to build predictive models. In fact about 70% of all orders on Wall Street are now placed by software, we're now living in the era of the algorithmic world.

Problem Statement

This project focuses to build a stock price predictor that takes daily trading data over a certain date range (time-series data) as input, and outputs projected estimates for given query dates. Note that the inputs will contain multiple metrics, such as opening price (Open), highest price the stock traded at (High), how many stocks were traded (Volume) and closing price adjusted for stock splits and dividends (Adjusted Close), though we only need to predict the Adjusted Close price so this is the most prominent concern for us. The challenge of this project is to accurately predict the future closing value of a given stock across a given period of time in the future.

Datasets and Inputs

In this project, we will be using the daily prices of the S&P 500 from last five years Jan 2014 to Jan 2019, this is a series of data points indexed in time order or a time series. Our goal will be to predict the closing price for any given date after training.

We will be using Google(GOOG), Apple(APPL), Amazon(AMZN), Facebook(FB) and Netflix(NFLX) finance data for this project. For each stock, the data will contain Open, High, Low, Close, Adj Close and Volume. The Adj Close/Close of each stock can be the target, and rest variables of the stock can be the inputs.

Solution Statement

For this project, the time-series data can be effectively handled by LSTM-RNN model, hence, it's the best possible solution to use a LSTM Neural Net model capable of learning from such time series data. We will make use of Keras over tensorflow backend to create LSTM RNN models. Also we will utilize Pandas dataframe for performing significant operations over time-series stock data. The performance can be measured based on prediction vs actual price for the stock.

Benchmark Model

I am using Linear Regression as the benchmark model for this project. This model will make use same preprocessed input features which will be consumed by LSTM model and thus it reflects a benchmark performance for LSTM-RNN model. Though linearity of it makes it inefficient for non-linear time-series data but still it lays down an important benchmarking for other effective model for this problem.

Evaluation Metrics

Since it's a clear cut regression problem, so the important metrics for such scenario would be R-square and root-mean-squared-error. R-square can help us in determining how much variation in the dependent variable is explained by the variation in the independent variables. While, root-mean-squared-error can provide what is the average deviation of the prediction from the true value, and it can be compared with the mean of the true value to see whether the deviation is large or small.

Project Design

We will use following steps to accomplish the implementation for this project:

1) We will use iPython notebook and python important modules/libraries like pandas, numpy, sklearn(for preprocessing and benchmarking), keras for LSTM model and matplotlib for visualization of figures.
2) Data collection: Download last five years data from Yahoo Finance for the mentioned S&P 500 stocks.
3) Data Preprocessing: We will use pandas to load data, normalize it and split it into training/testing data points.
4) Build benchmark model: We will use Linear Regression for benchmarking results
5) Build simple LSTM model: Single layered LSTM model.
6) Build improved LSTM model: Multi-layered LSTM complex network with dropout to handle over overfitting.
7) Visualize Results.
8) Check the robustness of the improved model.