Machine Learning Engineer Nanodegree

Capstone Proposal

Rohit Gupta
January 8th, 2018

Stock Price predictor with Recurrent neural nets

Domain Background

The transition from the traditional auction in the 1970s to computerized transactions was fueled by a need for an efficient access to the market. This set a stage for algorithmic and high-frequency trading. An estimated 70% of US equities in 2013 accounted for by automated trading[1]. Investment firms and hedge funds have used modeling to predict the market. They aim toward maximizing the return and spreading their risk over different financial assets. Traditional approaches can be classified as 'Follow-the-Winner', 'Follow-the-Looser', 'Pattern-Matching' and 'Meta-learning'. Though these models are efficient, their performance is dependent on the validity in different markets.

Quantitive strategies of hedge funds have received considerable returns for investors over the decades. The application of growing computing power and availability of big data has allowed models to identify and harvest on market inefficiencies[2]. These companies are actively looking for AI solution to outperform the benchmark indexes. In recent times many startups have come up with the aim of providing AI solutions for investing.

This work is towards utilizing Recurrent Neural Nets to predict the stock prices (for a 7 day period). The aim here is to leverage modeling abilities of neural networks for time series forecasting[3]. Time series forecasting is a difficult type of predictive modeling problem, as it has added complexity of sequence dependence among the input variables[7]. Recurrent neural networks have been quite successful in modeling time series data. LSTM is a type of RNN which can model short-term memory over a long period of time. Using LSTM enables us to model sequence dependence as short-term memory and incorporate it as a feature in our model.

Problem Statement

Project's aim is to give a 7 day prediction (forecast) for stock prices. This is a regression problem. The model should satisfy,

Category	Description
Input	Daily trade data having attributes such as open, close, high and adjusted close price. This input is provided over a date range for a stock.
Output	Forecast of Adjusted Close prices for query dates for pre-selected stock.

Expected Solution

The solution will be 7 predicted prices for each trading day within 7 trading days after the last date in the input date range. These prices will be compared with the actual price.

Datasets and Inputs

All of the necessary data for the project will come from Yahoo Finance website and dataset will be included in the project for ease of reproducibility. The input to the program will be a CSV file having historical data for a stock ticker symbol. The data will be of range 2007-2010 and around one trading year of data will be used for training.

Attribute (Target Stock)	Description
Adjusted closing price	value we are trying to predict
Volume	Reflects trade activity and market momentum for a stock
n day Rolling mean	n=20, to smooth out short-term fluctuations and highlight longer-term trends or cycles[5].

To capture the market trend and macro-micro indicators, we are going to use data of SPY and target stock ETF. Following are the attributes we will use,

Attribute	Description
n day Rolling mean	n=20, to smooth out short-term fluctuations and highlight longer-term trends or cycles[5].

Solution Statement

The expected solution is a model having the capability to accurately predict prices for a 7 days range, of a stock. We will compare the 7 predicted prices with the actual adjusted close prices and evaluate the model using R^2 metric. The code will be implemented in python. For LSTM we are going to use Keras. Keras is a high-level neural networks API, written in Python andTime Series Forecasting Based on Augmented Long Short-Term MemoryTime Series Forecasting Based on Augmented Long Short-Term Memory capable of running on top of TensorFlow[6]. Additionally, we will use pandas to load stock data into dataframe. For exploratory analysis, we will use numpy and matplotlib.

Benchmark Model

For the benchmark, we will use a simple Linear Regression model. We will use Scikit-Learn for training and optimizing this model.

Evaluation Metrics

It is important to measure the quality of a model by quantifying its performance. For this, we are going to use a R^2 metric which is also known as the coefficient of determination. We will use R^2 to predict the performance of our model. RMSE measures the average deviation of the predictions from the actual price. R^2 value reflects how well is a model performing[8].
The values for R2 range from 0 to 1, which captures the percentage of squared correlation between the predicted and actual values of the target variable. A model with an R2 of 0 is no better than a model that

always predicts the mean of the target variable, whereas a model with an R2 of 1 perfectly predicts the target variable. Any value between 0 and 1 indicates what percentage of the target variable, using this model, can be explained by the features. A model can be given a negative R^2 as well, which indicates that the model is arbitrarily worse than one that always predicts the mean of the target variable[4].

Project Design

Following are the proposed steps to achieve above

Setup the project (required installations)

- Anaconda
- IPython notebook
- Essential libraries (tensorflow, keras, pandas, matplotlib etc)
- setup git project

Load and transform data

- Download target stock and benchmark indices historical data from yahoo finance (API request are no longer supported).
- Load and process the data into pandas dataframe.
- Filter data for a given range of date.
- Normalize the data, and append to the main dataframe.
- As our data is a time series, we do a train test split using TimeSeriesSplit of sklearn. This cross validation method is a variation of kFold, it provides k fold as train set and (k+1)th fold as test set.

Exploratory analysis

Do an exploratory analysis of data to understand and reflect upon the important statistics of stock data.

Develop benchmark model

- Develop a benchmark Linear regression model using Scikit-Learn.
- Optimize the parameters.

Develop and improve deep learning model

- Setup a basic RNN using keras.
- Optimize hyperparameters.
- Reflect upon and log parameters and results.

Document and visualize results

- Plot and compare predictions of benchmark model, RNN model, and actual data.
- Plot deep learning model's and benchmark model's performance.
- Analyze and summarize the results.

Citations

1. Efficient market hypothesis
2. Artificial Intelligence: The new frontier for hedge funds
3. Financial Market Time Series Prediction with Recurrent Neural Networks, Armando Bernal, Sam Fok, Rohit Pidaparthi
4. Wikipedia article for R^2
5. Wikipedia article for rolling mean
6. Keras documentation
7. Time Series Forecasting Based on Augmented Long Short-Term Memory
8. R^2 definition, Boston housing price prediction