AI & ML
Project Report

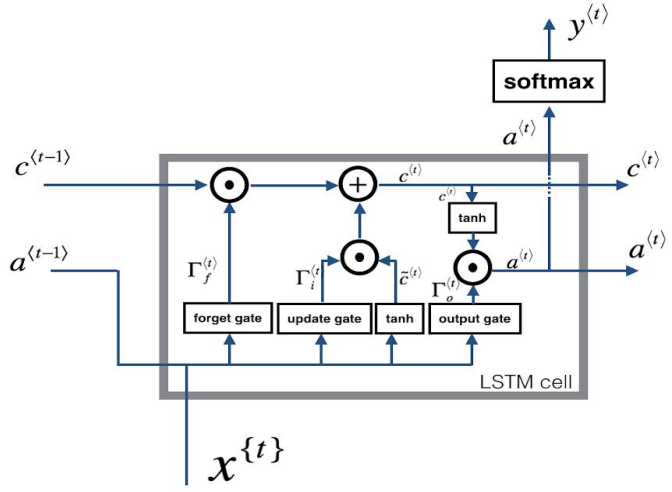
170050004 Yash Parmar
170050046 Nirmal Rajput
170050046 Devki Nandan Malav

Introduction

Predicting the Stock Market has been the bane and goal of investors since its existence. Everyday billions of dollars are traded on the exchange, and behind each dollar is an investor hoping to profit in one way or another. Entire companies rise and fall daily based on the behaviour of the market. Should an investor be able to accurately predict market movements, it offers a tantalizing promises of wealth and influence. It is no wonder then that the Stock Market and its associated challenges find their way into the public imagination every time it misbehaves. We have used deep learning forecasting techniques to predict future stock closing prices based on past stock prices to construct a portfolio of multiple stocks in order to diversify the risk. We did this via LSTM Recurrent Neural Network.

LSTM

LSTMs are very powerful in sequence prediction problems because they are able to store past information, this is important in our case because the past price of stock is crucial in predicting future's price. A typical LSTM unit consists of a **cell**, an **input gate**, an **output gate** and a **forget gate**. The cell remembers values over arbitrary time intervals and the three *gates* regulate the flow of information into and out of the cell.



$$\Gamma_f^{\langle t \rangle} = \sigma(W_f[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_f)$$
$$\Gamma_u^{\langle t \rangle} = \sigma(W_u[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$$
$$\tilde{c}^{\langle t \rangle} = \tanh(W_C[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_C)$$
$$c^{\langle t \rangle} = \Gamma_f^{\langle t \rangle} \circ c^{\langle t-1 \rangle} + \Gamma_u^{\langle t \rangle} \circ \tilde{c}^{\langle t \rangle}$$
$$\Gamma_o^{\langle t \rangle} = \sigma(W_o[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_o)$$
$$a^{\langle t \rangle} = \Gamma_o^{\langle t \rangle} \circ \tanh(c^{\langle t \rangle})$$

Setup

We have used following libraries in our project -

- Tensorflow
- Seaborn
- Numpy
- Matplotlib
- Pandas
- Sklearn
- Tqdm
- Datetime

You have to install them in your computer device before running the project. This can be done via pip3 install <library_name> where <library_name> can be among mentioned above.

Methodology

- Data Normalization
- Creation of LSTM model as our RNN architecture
- Forecasting Simulation
- Sanity Check
- Forecast Visualization

Data Normalization

We have normalized closing price of the training data to be in range of (0,1). For this, we have used MinMaxScaler module from scikit-learn.

Creation of LSTM model

We have built Recurrent Neural Network consisting of LSTM units. We have used only single layer. In that layer, there are 128 LSTM cells used.

Forecasting Simulation

We have simulated above model 10 times to predict future stock prices for the next 30 days. To ensure that, our model does not depend heavily on some specific LSTM units, we have used dropout rate of 0.8. Along with the mentioned parameters, we have used the following parameters while training our model - Epoch = 300, Learning Rate = 0.01, Batch Size = 5. To ensure faster Convergence, we have used AdamOptimizer in our model.

Sanity Check

Some of our models might not have a stable gradient, so the forecasted trend might really hangwired. We have used the following methods to filter out unstable models.

1. If one of the elements in forecasted trend lower than min(original trend).
2. If one of the elements in forecasted trend bigger than twice of max(original trend).

If both are true, reject that trend.

Forecasting Visualization

Predicted stock closing prices are plotted against datetime for each of the Simulated models. You can find the average accuracy of all simulated models to be 97.97%