STOCK PRICE PREDICTION USING TENSORFLOW

Project Report

Protim Mondal

The goal of **Stock Market Prediction** is to forecast the future value of a company's financial stocks. The most recent development in market prediction technology is the application of machine learning, which generates predictions based on the values of current stock exchange indexes after training on their prior values. Machine learning applies a variety of models to make prediction more accurate and reliable. The research focuses on the application of **Tensorflow** and **LSTM**-based Machine Learning to forecast stock prices. Open, close, low, high, and volume are all elements to consider.

The major goal of this project is to educate readers about deep learning applications in time series forecasting using the example of stock market price prediction. This is just one of many use case scenarios that you can develop with neural networks (**stacked LSTMs** in this case). It is critical to note that before adopting these concepts in real-world circumstances, make sure you investigate the many tools at your disposal and determine which ideas work best for your application. Let us now proceed to a full breakdown of our project from the ground up.

Methodology:

A correct stock forecast will result in enormous earnings for both the vendor and the broker. We employed data mining and machine learning techniques to make predictions about the stock market.

- Preprocessing of data to ensure consistency.
- Model construction and algorithm selection.
- Learning from the model outcomes.
- Data visualization - converting data into a graph and cleaning the text:

Preparing the data:

We'll look at a random dataset from Kaggle that has historical stock data from **Apple.** We'll read the CSV file with Panda's library . Although there are other characteristics to examine for the stock market prediction model, we will only look at the Close/Last column because it provides a more average approach. It is also simpler to consider only one of the columns for training and validation.

Data Preprocessing:

The following step will be to prepare our dataset. We will use the **numpy** and scikit-learn tools to visualize our data as an array for better modelling. Because the data is somewhat large and the values are high, it is preferable to fit them between **0 and 1** for better computation. To carry out the operation, we will use the **Min Max Scaler** function from the **scikit-learn library.**

Visualization (EDA):

The **Matplotlib** package is one of the top visualization libraries in the Python programming language. It will enable you to visualize the dataset appropriately. Here I visualize the model using the data and their corresponding indices. The data comprises of the stock values at their corresponding intervals.

- I have conducted an in-depth analysis of Apple's stock price data, spanning a decade from 2013 to 2018. Open, high, low, and closing prices. These key price points were used to examine price dynamics, trends, volatility, and trading patterns in stocks and financial instruments.
- We learned from this that there are 2518 rows of data available, with 6 different attributes or columns for each row.
- The plot of **Apple** closing price demonstrates an upward trend, with the largest peak occurring in 2018. Although came down dramatically from 2015 to around August 2016, it increased rapidly again .
- We can see from our visualization that the plot of the **daily dollar rate** appears to have an increasing tendency.

Now that we have a basic understanding of the shapes produced by our testing and training datasets, as well as their respective outputs, the final critical step is to convert these existing dataset shapes into a format suited for Long short-term memory **(LSTM)** models. Because our existing structure is merely a **two-dimensional space**, we must convert it to a **three-dimensional space** in order to do **LSTM** operations and calculations.

<u>Building the Model:</u>

We will use stacked **LSTM** layers and our deep learning neural network to design an architecture to tackle this challenge. To acquire a more extensive approach to these specific issues, I highly recommend reading my prior two posts on the introduction to **TensorFlow** and **Keras.**

The **Dense** and **LSTM** layers are the only two layers required for processing our computation of the stock market price prediction problem. Because we only need one output or prediction for a certain set of parameters or pieces of data, our Dense layer will only have one output node. We have already gone over the subject of LSTM layers in greater depth.

We supply three critical parameters while building a model:

- **optimizer** - This is a method for optimizing the cost function using gradient descent.
- **loss** - The loss function is used to determine whether or not the model improves with training.
- **metrics** - This aids in model evaluation by predicting training and validation data.

<u>CONCLUSION:</u>

In this study, we conducted experiments on a unique approach to predicting stock prices using data from numerical analysis. The **LSTM** model with a sliding window was used for the numerical analysis. This yielded an MSE of 100.91. We can see that the root of these MSE values is not too bad and that they can yield useful results. They are congruent with the model that was developed and can be utilized to generate reasonable and suitable predictions.

<u>FUTURE WORK:</u>

Using other methodologies, numerical analysis results can be improved. A revolutionary deep learning system for stock price prediction that combines layered auto encoders, long-short term and wavelet transformations (WT). Better results can be obtained by decomposing the time series using wavelet transformations to remove noise to generate deep high-level features.