Stock Market Predictions

Deep Learning Project, March 2021

Filipe Dias (m20200655)
John Ruiz (m20200540)
Mariana Byrne (m20200638)

Abstract

The nature of this study is a Time Series problem, forecasting the stock market closing price of Goldman Sachs. The Time Series are not easy to predict, we are aware of it, we tried with a simple model and got some results (Trial1 notebook).

After some reading about how to forecast timeseries with RNN we found out that using LSTM has some limitations and we dug into it. We found out that most models in real life situations used to forecast timeseries make use of extra tools like Time2Vec and the transformers/Attention mechanisms.

We ended up having good results with the application of those tools.

1. Introduction

The nature of our study is a time series problem, using stock market behavior to forecast the market closing price, for which we will be using the stock market information of Goldman Sachs. As we know Long-Short Term Memory networks (LSTMs) have become a common use for time series analysis, however there is some criticism to it. There are some papers published that say that it is in fact impossible to predict stock markets at all, because they are very random [1], and susceptible to too many uncontrollable factors. Other concern is that LSTM work better when looking at a short time window, a one-day prediction for example, but will struggle to predict based on years of data, like stock market behavior. While researching about the topic, we noticed that for time series prediction, people in the field are not only relying in LSTM alone but are also applying some new techniques. To try and prevent those LSTM memory problems we are applying some of those techniques: Time2Vec, Encoder-Decoder Model, Attention Mechanism (an upgrade to Encoder-Decoder Model).

2. Background

Time2Vec is used to improve the model notion of time, it will be implemented as a normal embedding layer. Because time is such a relevant feature in real life situations it is important that it is encoded the right way so that the model can learn the most information from it, especially in what concerns the temporal order of the stock prices. [4] This representation has two main properties: it considers non-periodic (linear) and periodic patterns; and is invariant to time rescaling. [4]

Encoder-Decoder Model is being used because contrary to the RNN, LSTM, GRU, that each input corresponds to an output for the same time step, we want to predict and output sequence from an input sequence of different length, without a correspondence between them, this is called sequence to sequence mapping and the Encoder-Decoder Model was introduced to address it. As the name suggests there are two parts to this model: The encoder: responsible for stepping through the input time steps and encode the sequence into a vector of fixed length, called context vector. The decoder: responsible for stepping through the output time steps while accessing the context vector.

The Attention Mechanism, also called *Transformer*, was developed by Google, is an evolution of the *Encoder-Decoder Model*, developed to improve the performance of long input sequences. The transformer works by ways of a self-attention mechanism comprised by two layers: the single-head attention, and the multi-head attention. From the single-head attention layer we can assign attention scores to each feature, which are then concatenated in the multi-head attention layer and passed through a dense layer, so that a non-linear transformation can be applied. [2, 3]

Global Average Pooling is a pooling operation designed to replace fully connected layers in classical CNNs. The idea is to generate one feature map for each corresponding category of the classification task in the last layer. Instead of adding fully connected layers on top of the feature maps, we take the average of each feature map, and the resulting vector is fed directly into the softmax layer. [6]

3. Data and Methodology

Pandas Data Reader was the tool used to directly get the data from *Yahoo finance*. The data acquired goes back to the year 2000 and the variables we are using are “High”, “Low”, “Open” and “Volume”. The closing price will be our target variable.

Before inputting the data into the model, some transformations were applied to the data. First, we also calculated the moving average of the price features, so that the data is smoothed. Furthermore, we calculated the percentage change of all the variables, to work with the daily stock returns and daily volume changes. This step helps to increase stationarity in the data, making it less vulnerable to time changes, and allowing our model to make predictions with a higher degree of validity. After this step, we applied a min-max normalization, and split the data into training, validation, and test sets. On the following graphs, we can see the transformed data. Finally, these three sets were separated into sequences of 128 days that will later serve as input for the *Time2Vec*.

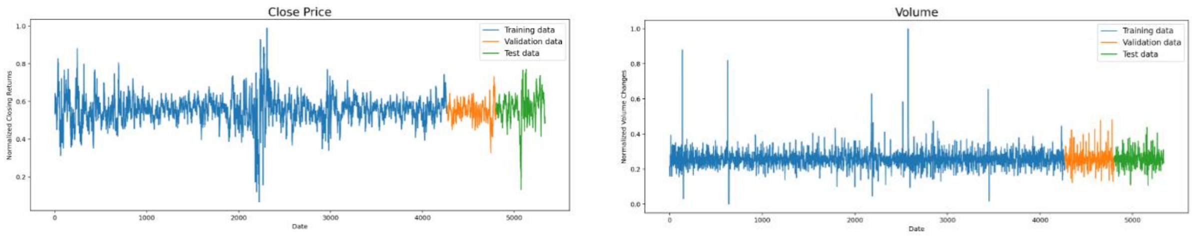

Figure 1. Training, validation, and test split of the normalized data.

For the model evaluation, the *Mean Squared Error* (loss function) and *Mean Absolute Error* were used as metrics.

4. Model

As a first attempt to forecast stock market values, we used a simple LSTM model, which is commonly used in this field. However, this model did not output the results that we pretended. We decided to go down the path where we would use the techniques mentioned before, so we could improve our model’s notion of time and space.

We found some limitations when we tried to add market indicators such as Simple Moving Average and Relative Strength Index. We expected that those indicators would bring more information to the model, but it failed. We did run for both Goldman Sachs and for Microsoft with the hope that the problem would be something related to the behavior of the time series but unfortunately, we could not forecast anything at all (notebook 3). We were also using PCA and XGBoost, for feature selection but since we could not achieve the desired results, we had to drop it.

The model is initialized with the Time2Vec a time embedding layer and one attention layer, followed by a global average pooling layer, which makes the network less complex, and finally two dense hidden layers. Additionally, there are two dropout layers before each dense layer, to avoid overfitting.

We first attempted to run the model with one attention layer, and the graphs below show that a single transformer is good enough to predict the closing price.

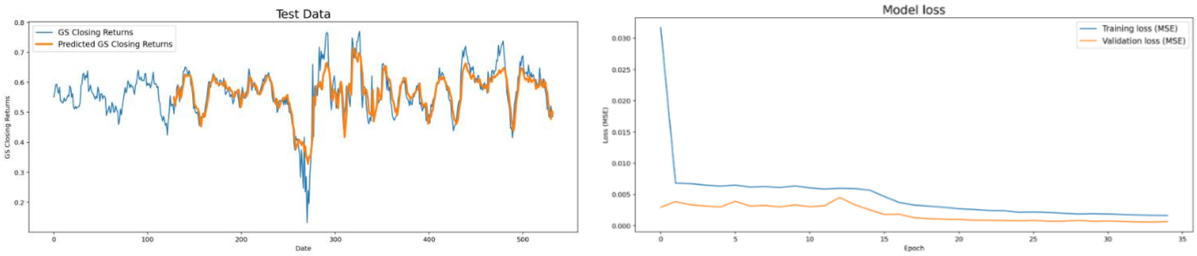

Figure 2. Predicted closing prices plotted against original closing prices (on the left). Graph showing the loss of the model with one attention layer (on the right).

Following that attempt, we ran the model with two, three and four attention layers, separately, and found that with the latter architecture we started to get bad results (flat prediction line). Therefore, we opted that for the architecture of the final model, we would have three attention layers.

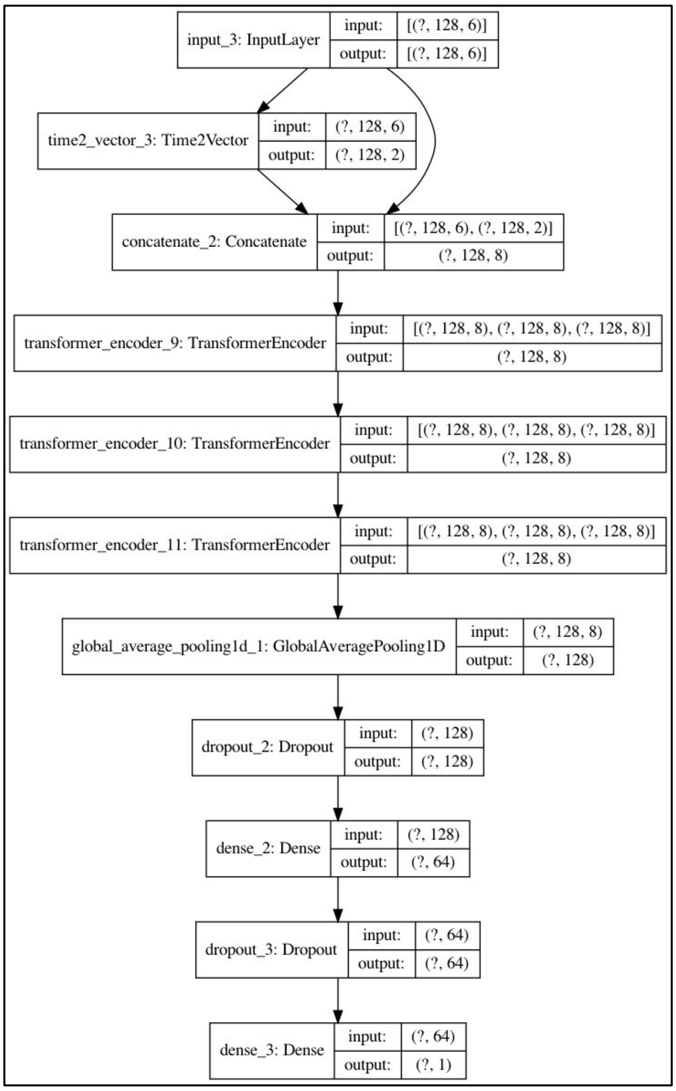

Figure 3. Final model.

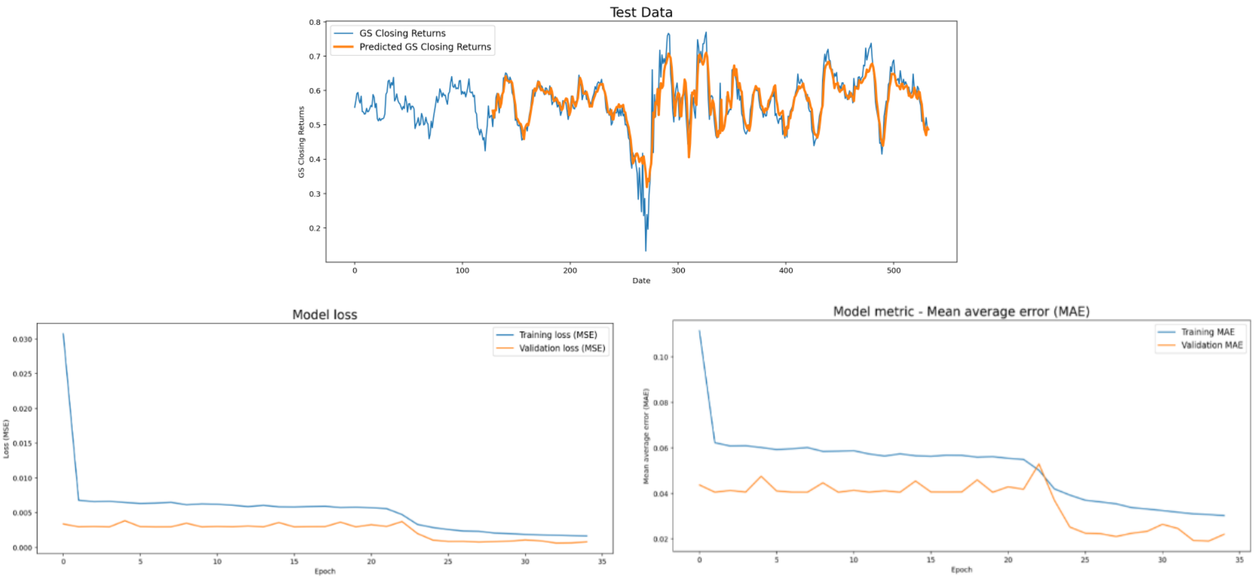

Figure 4. Predicted closing prices plotted against original closing prices (on top). Graphs showing the evaluation metrics of the final model.

Looking at the MAE graph we have to look for the place where the validation error is the minimum, by running 35 epochs we can see that it would be around epoch 32. It's not possible to say that if that is global minimum since we were not able to run it for more than 35 epochs due to computational/time issues. Although If it was the global mininum we should stop there so that our model can generalize well on new data. By looking at the MSE graph, the validation loss is lower than the training loss, it indicates that the validation dataset may be easier to predict than the training. It is also clear that the model is not overfitting.

5. Conclusions

If we look at our model without considering the tools used, Time2Vec and Attentions layers, the model is actual simple, two dense layers, which one of them is the output layer and two dropout layers. This means that the tools used to forecast time series are good. We could have added more layers and make more trials but unfortunately running the model with those two functions associated to it turns out to be very time consuming and computationally intensive/demanding.

As a continuation of this work, we would try to improve the model that uses the technical indicators, seeing as it seems logical that these would improve the results of the model. Furthermore, we would purpose to add sentiment analysis to news articles and tweets to also consider factors that are not controlled by the stock market, such as the opinion of the public regarding the company in question.

To conclude, neural networks are on the rise and being used on several areas, more and more tools like the ones implemented are coming out every day. It is a must-know tool for any data scientist.

REFERENCES

[1] Ariton, L., 2021. No, LSTMs Can't Predict Stock Prices. [online] Medium. Available at: <https://medium.com/analytics-vidhya/no-lstms-cant-predict-stock-prices-11f10dcb35d6> [Accessed March 2021].

[2] Ashish, V., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł. and Polosukhin, I., 2017. Attention is All You Need.

[3] Chen, Y., Lin, W. and Wang, J., 2019. A Dual-Attention-Based Stock Price Trend Prediction Model With Dual Features.

[4] Kazemi, S. M., Goel, R., Eghbali, S., Ramanan, J., Sahota, J., Thakur, S., Wu, S., Smyth, C., Poupart, P., M. Brubaker. 2019. Time2Vec: Learning a Vector Representation of Time. Borealis AI.

[5] Schmitz, J., 2020. Stock predictions with state-of-the-art Transformer and Time Embeddings. [online] Medium. Available at: <https://towardsdatascience.com/stock-predictions-with-state-of-the-art-transformer-and-time-embeddings-3a4485237de6> [Accessed March 2021].

[6] Versloot, C., 2020. What are Max Pooling, Average Pooling, Global Max Pooling and Global Average Pooling? [online] MachineCurve. Available at: <https://www.machinecurve.com/index.php/2020/01/30/what-are-max-pooling-average-pooling-global-max-pooling-and-global-average-pooling/> [Accessed March 2021].

[7] Wilcox, P., 2019. Stock Forecasting with Transformer Architecture & Attention Mechanism - Lucena Research. [online] Lucena Research. Available at: <https://lucenaresearch.com/2019/11/18/how-transformers-with-attention-networks-boost-time-series-forecasting/> [Accessed March 2021].