Stock Prediction using Machine Learning and AI

LaTeX template adapted from:
European Conference on Artificial Intelligence

Iustin-Andrei Moisa-Tudor[1]

Other group members:
Vitalie Stinca[2], **Ansad Parayil**[3], **Habeeb Rahman Nanath** [4]

Abstract. For our project, we have chosen to create a couple of Machine Learning algorithms in order to properly predict future stock prices based on past data from the past 10 years. In our project we chose to create 4 different AI models which we can then compare with eachother and see which is the most efficient. We chose to use the google dataset of stocks from a wide range of dates. Then we tested the AI models with recent data to get a prediction. In our code we used different models, such as Linear Regression model, Convolutional Neural Network Model, RNN Model and LSTM Model in order to check and compare which has the best accuracy and which model performs faster.

1 Introduction

I have chosen to create a Convolutional Neural Network in order to predict the future stock prices based on past data, which is an excelent alternative compared to Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), and linear regression models. The main reason why I chose to do a CNN compared to the other AI models is because it has certain strenghts such as CNNs being able to learn spatial relationships from input data. When talking about stock prediction using a CNN, it can easily and accurately identify features between historical price sequences. In my case I used a Convolutional Neural Network to determine future stock prices. A CNN can also extract hierarchical representations of data using convolutional layers which can represent crucial features in stoick prediction. Another benefit of using a CNN for stock price prediction is that it is resource light, meaning it won't need a lot of computational power in order to train the AI model, and it will take less time to train using a CNN model with a large data set since it doesn't take as long as other alternatives to train. In order to process the training and test data from the datasets we used, I used the Pandas library which allows me to analyse and manipulate the data inside the dataset. I used it to extract the data from the column named Close in the dataset since it shows the final prices for the google stocks which is what we all used for training on certain dates. The data from the Close column was converted to numerical values and I had to modify the dataset so that rows that had NaN/empty values to be removed from the dataset itself. I also normalised the Close column between 0 and 1 using MInMaxScaller which helps in model training. For my CNN, I initialised a Sequential model from the Keras Python library and created a couple of layers for training such as 2 Conv1D Convolutional neural network layers that have RELU activation function which might help with finding patterns that are more complex in the time series data. And I added some dropout layers to prevent the model from overfitting during training. I trained the model with 60 time steps, and I used the widely known "Adam" optimiser and MSE (Mean Squared Error) which are used as poss functions to minimise any errors during predictions. The epochs used for training is 50 bebica gave me good results, however I'd like to mention that using more epochs will allow the loss to be minimised a lot and will make the overall predictions more accurate for the AI model. Regarding the loss, I used a library called matplotlb to display a graph with the loss of the AI model and assess its performance. For the test data I had to preprocess the test data in a similar way I did for the training data X Train. I still converted the "close" column to numerical values and still scaled the data using MinMaxSCaler once again then I prepared the data sequence for prediction. The model will predict the test stock prices for the test dataset then inverse transformation is applied to the predicted test data in order to revert the scalled predictions to their original values which allows for a easier comparison between the stock prices. In the next step I created a plot graph using the library matplotlib which I talked about earlier in the document when I plotted the training loss into a graph, but this time I plotted the actual stock prices and the predicted stock prices I got from the ai model using the test dataset containing the predicted stock prices for Google. The plotting on a graph allows the user reading the plot to easily see the difference in between the actual prices of the google stocks and the predicted values or prices generated by the AI model during prediction phase. Which in fact offers insights into the performance and accuracy of the Convolutional Neural Network (CNN) model that I created in order to predict the Google Stock Prices.

2 Background

The other members of my group, like mentioned earlier have done 3 other AI models. A linear regression model, a RNN and an LSTM model. The Linear Regression Model will predict the direction where the overall price will go, the LSTM Model does not have any functions to avoid overfitting but the RNN which is also an LSTM Model has 4 LSTM layers and functions that avoid overfitting in order to

[1] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: im3904v@gre.ac.uk
[2] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: vs6390s@greenwich.ac.uk
[3] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: ap1650o@greenwich.ac.uk
[4] School of Computing and Mathematical Sciences, University of Greenwich, London SE10 9LS, UK, email: hn5462v@greenwich.ac.uk

get more acurate results from the AI model. Some of the advantages of linear regression are that it has a fast implementation, it requires small datasets and small data requirements and it's easy to analyse the data, with LSTM, it is better at handling long term dependencies and it is efficient at modelling complex sequential data however it is more resource hungry.

3 Experiments and results

For my model I used the Google Stock Prices dataset as Google train data and Google test data , It contains the historical Google stock data downlaoded from yahoo finance. And I split the dataset in 2 different files, one being the train dataset using the Close column for training for older data, and the most recent data has been added to the test dataset where I used it to test the model's accuracy in predicting stock prices. My code is integrating Convolutional Neural Network (CNN) layers that uses keras to help find features inside the google train dataset which is using a .csv format. It is using Conv1D and ReLU activation to extract spatial patterns from the inputs representing historical stock prices data for Google. I have also chosen to use Dropout layers which allowed me to stop the model from overfitting. I decided to utilise a higher number of epochs in order to get more accurate results and minimise the loss of the AI model after training and it also allows the model to capture more complext data patterns and improve the final model. By using highjer epoch count it makes sure that the model will have greater prediction rate and lower loss rate. I also decided to use the ADAM optimiser since it is widely used and has also been used by my group members in order to optimise AI models and accelerating convergence during training.

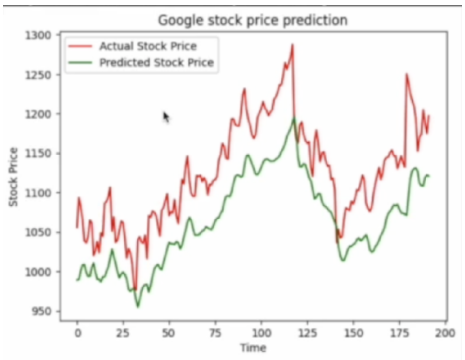

Figure 1.

In the final part after I trained the model, I used it in order to plot in tnto a graph the actual stock value and the predicted stock value for the final model using the test data of the google historical stock prices dataset containing more recent data and the model worked with no errors and gave me a good result as seen below in the figure 1.

4 Discussion

My CNN approach on the Google Stock Prices dataset focused on capturing spatial patterns within historical stock data. The CNN model, compared to Vitalie's RNN LSTM, Ansad's LSTM, and Habeeb's linear regression, was better suited for this task. CNNs excel in identifying intricate spatial features within the data, offering an advantage in capturing complex patterns that might influence stock prices. However, RNNs and LSTMs are strong at modeling sequential dependencies, while linear regression handles linear relationships effectively. While each approach has its strengths, my CNN model's ability to discern spatial features within the historical stock data contributed to more accurate predictions, as evident in the obtained results, without compromising its ability to generalize well to new data.

5 Conclusion and future work

This coursework has helped me with learning how to use, manipulate and adapt a dataset to be used with a convolutional neural network in order to create a accurate stock prediction program, in the future, if I am encountered with this kind of work again, I know how to deal with it and how to bring good results in the end. Overall, the Convolutional Neural Network (CNN) model I created showed accurate results during training and testing phases of building the AI model using past stock prices from our datasets. It it able to capture spatial patterns in Google historical stock prices data like any other Convolutional neural network and it proved that it can acuratete predict the results I wanted. Which can also be reused for other datasets as well as long as they have the same format as our datasets containing a date and "close" column. But my CNN also has limitations, it is excelent in capturing spatial data within the dataset I trained it on but in my opinion and from my research, a Convolutional neural network architecture might not fully exploit the temporal dependencies in the sequential stock prices data in the dataset. For example, in comparison with my other colleagues who did a Long Short-Term Memory (LSTM) model, Recurrent Neural Network (RNN) and linear regression model, my CNN showed robust performance especially in identidying spatial patterns that influence the stock prices in the data I trained it on which is more complex. The Convolutional neural network model I created gave me accurate results and it showed why this architecture might be accurate for stock prediction tasks. Despite the strengths of my CNN model, in the future I could also create a combined model that uses multiple model architectures for better and more efficient stock prediction such as combining the architecture of a Long Short Term Memory LSTM with a Convolutional Neural Network like mine which could use both spatial and temporal dependencies which might increase the accuracy of the stock prediction model created. All of our solutions were good and fast but each had it's own advantages and disadvantages, in my opinion by testing (trian and error) I have gained insights on possible errors that might encounter using the CNN in creating a newural network capable of stock prediction and it helped me train myself in the feld of Artificial Inteligence. All our solutions gave us good results and I think my Convolutional Neural Network was the most appropriate solution for our stock prediction problem.

...
...
...

ACKNOWLEDGEMENTS

I have gained insights on Convolutional Neural Networks and how to manipulate datasets to create an accurate AI model. I worked with a great team which allowed us to efficiently complete our designated tasks in time and in a proper manner. I have worked with a great team and gained a lot of skills during this coursework. Regarding team collaboration, all of the other members have put equal effort in completing their parts of the coursework. We all helped eachother out when we had problems with the dataset or when we got syntax

errors since all of us have different knowledge and one knows what the other person doesn't, this is helpful especially when you want to built close relationships with people and work efficiently in getting the designated task done which helps out the team by a lot. We had a team leader who always prepared presentations during each meeting we had and it helped us a lot with getting organised and completing our tasks in time. In the last minutes we changed the topic for out ai coursework to stock prediction and we were working as a team in order to complete these tasks before the deadline and on time for submission which makes me say that I am glad I worked with my team members since we are all eager to learn more about the field of artificial intelligence and this has driven us to work more efficiently and in a smarter manner. We had a great teacher which made it very easy for us to also understand what tasks had to be done in the coursework as well as helped us remember what was taught in lectures and labs with easy due to the way the content was explained, which was easier than we thought. When we had any problems we always went to the teacher and solved it as well as gained insights on what we were doing wrong so we can prevent that from happening into the future when developing more AI models like our ones. I am glad that we had a lot of resources to inspire ourselves from, I personally picked up a book for Convolutional Neural Network from the university's library which helped me gain insights on more uses of CNN's as well as where to find resources and solutions to my problems, since we all did stock prediction, it was also relatively easy for us to find datasets from websites such as Yahoo finance since we can input the dates from where we want to download a CSV document containing favirous features related to the dates and stock prices we need, in our case, Google Finance Stocks. The libraries we worked with in class were very helpful in the coursework as well as a course I am studying in Tensorflow and Machine Learning taught by Google themselves which gave me more insights into AI and ML. In my free time I also did a course from IBM which allowed me to gain processing in Explanatory data analysis for machine learning and it helped me find different techniques of how to preprocess and read data as well as manipulate it to my liking.

REFERENCES