Stock Options Prediction with Advanced Deep Learning

Jacky Chow

Jerry Huang

Jumana Nadir

Abstract

In the investment terms options is a derivative that is derived from the price of another security. Another security can be a stock, a currency, a rate or a commodity. This means that the price of options moves if the price of another security would move. Understanding how options are priced is important because there are a lot of variables that determine their value. In this project, we are creating a Deep Learning model for the prediction of the price of an option. Options prediction can help in determining the investment better and understanding the stock marketing better. In our model, we pick specific options for the stock. We need to consider the five important characteristics of the option stock-Underlying asset, Call vs. Put, Strike price, Expiration date, and American vs. European. These five features are the input to our model and the output of the model will be the price. As we know there are many different strategies for options trading depending on what we want to get and how much risk we are willing to expound, we will limit the features for the scope of this project. We are using the open-source dataset such as Robinhood API and Yahoo Financial API to get options data for training our model. For our model, we are using 'tsai' which is a deep learning package built on top of fastAI for state-of-the-art time-series classification and regression.

Introduction

Options market refers to the sum of all the options to buy or sell, can be described in the global or regional scale options trading market and the stock market is closely linked because options trading is one of the most widespread types of stock options can also be based on other financial instruments (such as futures, commodities, currencies and indices) to obtain the basic premise of options contracts is that by buying the contract to pay a premium and the buyers shall have the right at a predetermined price (called the strike price) to buy a specified number of underlying assets, but has no obligation to do so if a buyer decides to buy assets, this is known as the exercise of an option contract Depending on the style of the contract, the contract may be exercised at any time before or only at the due date; They are American style or European style. In our project, we are using the European style of the option.

We use the input to study the performance of deep learning models on pricing options the popular Black-Scholes model. By treating option prices as a function of contract terms and financial conditions, we can use neural networks to avoid assumptions Understand financial mechanisms and learn from historical data. model2 and model3. Taking 5-day, 10-day and 21day historical volatility as input, the LSTM model takes the base price of the last 20 days allows it to understand volatility. Because of U.S Unprecedented data availability, we have not ruled out any possible illiquid options from our data set, so that our model can be generalized to them. All models executed Far better than the Black-Scholes model, and we found the bid/ask price rather than the equilibrium price in model2 is the most successful. This implies Future work using historical data should consider forecasting bid/ask prices.

Related Work

We did some research on related papers and study for option data and found some interesting literature review. We cover them in the section below-

In a proposed study by Mezofi and Szabo [2], they characterized the final outputs of the neural network approaches in three categories: directly predicting the option premium, while possibly adding in implied volatility or other engineered features; predicting implied volatility and using it as an input to Black-Scholes to return the option premium [3]; and finding the ratio between the options premium and strike price. Garcia et al in their proposed method introduced the homogeneity hint to constrain the set of possible outputs such that the option pricing function is homogeneous in asset price and strike price with degree [4].

A Stanford University study by Alexander Ke and Andrew Young [1] illustrates a deep learning model for pricing options that learn from historical data. As there is no open-source dataset available for stock options, the authors obtained their options prices and security prices from Wharton Research Data Services and supplemented this data with the treasury yields rates from the US Treasury Resource Center [6]. In this study the authors developed three working deep learning models for multi-task learning and compared them using various hyperparameter tuning.

In a similar study by Malliaris and Salchenberger, they first developed a neural network approach to estimate the close price of S&P 100 options using transaction data from the first six months of 1990. They supplemented the contract data with the option premium and underlying price for the day prior, as well as historical realized volatility. They further divided their training set to separately forecast the price of in the money and out of the money options, which may lead to overfitting and disagrees with the practically continuous property of stock prices. They constructed a neural network with a hidden layer of four nodes and one output node. Using mean squared error, this network outperformed the Black-Scholes model in about half the test periods [5].

Dataset and Features

In this project, we have three different datasets. The first one is all the stocks dataset from NASDAQ. We collected 3 years data in more than 3000 company's stock data from yahoo finance. It has about 200MB of the data size and 1367420 rows of data. We use this dataset for picking the stock, the final format we used to run the clustering is the CSV format file.

The second dataset we use is we crawl the option datasets on yahoo finance. The option features we crawled are underlying price, exchange, option symbol, type, expiration date, start date, strike price, bid price, ask price, volume. And we picked 26 company's options from the result of the stock picking strategy. We collected around 150MB of the data from January 2018 to July of 2019. For stock picking strategy, we used clustering based on the highest Sharpe Ratio of stocks.

The third data we use is we collect the treasury rate data which is provided by the US government from 2018 to 2020. We do the data preprocessing in our stock dataset to get our Sharpe rate. And we do the preprocessing in option dataset to get the sigma dates, different dates, and calculate the Black Scholes score etc. And we do the many steps of the data preprocessing and we will mention it in the next paragraph.

Methods

At the beginning of the project, for any investment in option marketing. Picking a good option is necessary. Based on location and other factors there are different types of stock markets around the world. If a person has money and wishes to invest in a stock it is difficult to find the best prediction with human intuition. Therefore, for the brevity of this project, we center our study on the NASDAQ stock exchange-listed companies. We collected 3 years of data in more than 3000 company's stock data from Yahoo Finance. It has about 200MB of data size and 1367420 rows of data. We use this dataset for picking the best stocks. Our stock-picking strategy is explained below. Firstly, we apply K-means clustering to these 3000 companies based on their average yearly returns and yearly variance. And we further reduce this cluster of 3000 companies down to 29 companies based on their Sharpe Ratio. We calculated the average annual return and variance based on the below formula.

1. Gains of each year = Last trading day Close price - First trading day Open price.

2. Annual Return of each year = Gains / First trading day Open price.

3. Variance = (Each year's annual return - Average Annual Return).

For the K-means clustering, we selected optimal clusters based on the sum of squared errors (SSE) within clusters and the Silhouette score. Then we used the Sharpe Ratio metric to better

evaluate the cluster performance. Sharpe Ratio is used to help investors understand the return of an investment compared to its risk involved. It is derived using annual returns, variance and risk-free rate over a period of time. The ratio is the average return earned in excess of the risk-free rate per unit of volatility or total risk. A Sharpe Ratio of more than 1 is considered good while a Sharpe Ratio of more than 2 is considered very good.

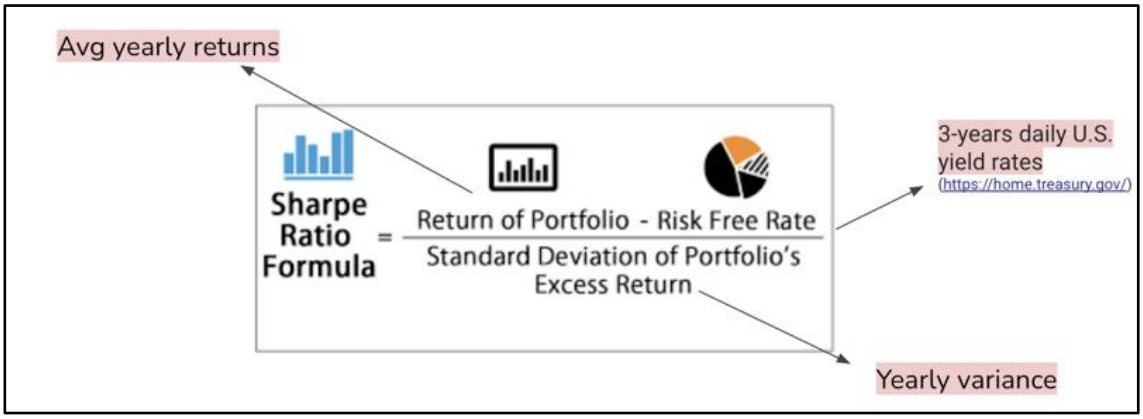

Figure 1.1

We once again applied K-means clustering based on the Sharpe Ratio, and we found a best cluster of 29 companies. After sorting the companies based on their Sharpe Ratio, we get a list of best stocks to invest in. We use these stocks to apply our Reinforcement learning approach to profit prediction.

In our project, we created two models, LSTM and regression model for model2 and model3. Our LSTM is built based on the RNN architectures. Because RNNs capture state information, we hope that this architecture can learn estimate volatility from recent observations to improve option pricing performance. And the reason why we use 8-unit LSTM is because we take a look at the closing price within 21 days but when we calculated the days in over 21 days. And our LSTM model does not have the dropout because we think not to add dropout in LSTM cells for one specific and clear reason. LSTMs are good for long terms but an important thing about them is that they are not very well at memorizing multiple things simultaneously. The logic of drop out is for adding noise to the neurons in order not to be dependent on any specific neuron. By adding drop out for LSTM cells, there is a chance for forgetting something that should not be forgotten.

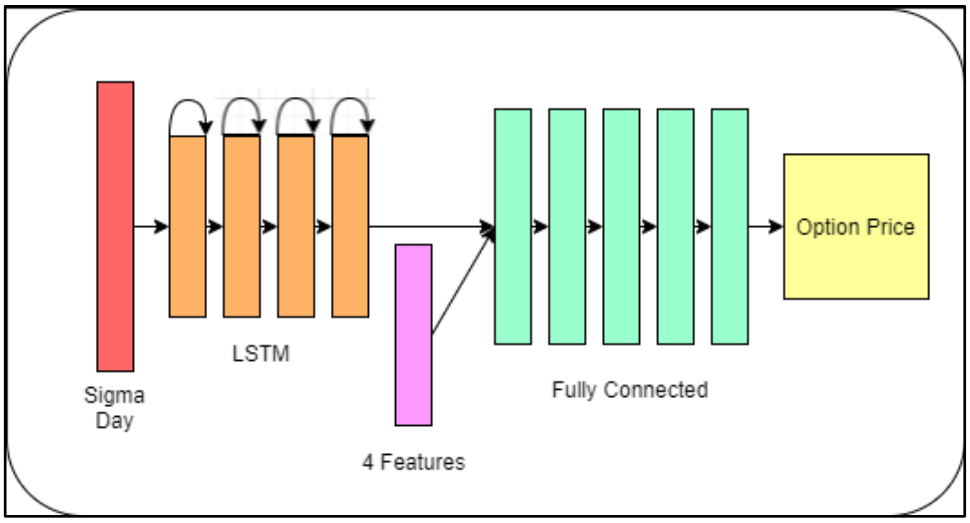

Figure 1.2

We extend the hyperparameter search in Liu et al., and Liu et al. performed a similar task on the network. The task is to calculate the implied volatility based on the data generated by BS. Our network contains four hidden layers: three layers, each with 400 neurons, and one output layer with one neuron. Liu et al. have not tried Leaky ReLU activation, but we found that using this method can make the network learn a bit faster, so our 400 neuron layers use Leaky ReLU. Our output node uses ReLU activation, which is appropriate because the option price is non-negative. As a result of these activation choices, the weights are initialized by Glorot initialization. Contrary to Liu et al., we found that batch normalization significantly improves the training speed and the loss during convergence, so we apply batch normalization after 400 neuron layers. We did not apply any regularization techniques, because option prices are sensitive to all inputs provided, so dropping out does not improve the accuracy of price predictions.

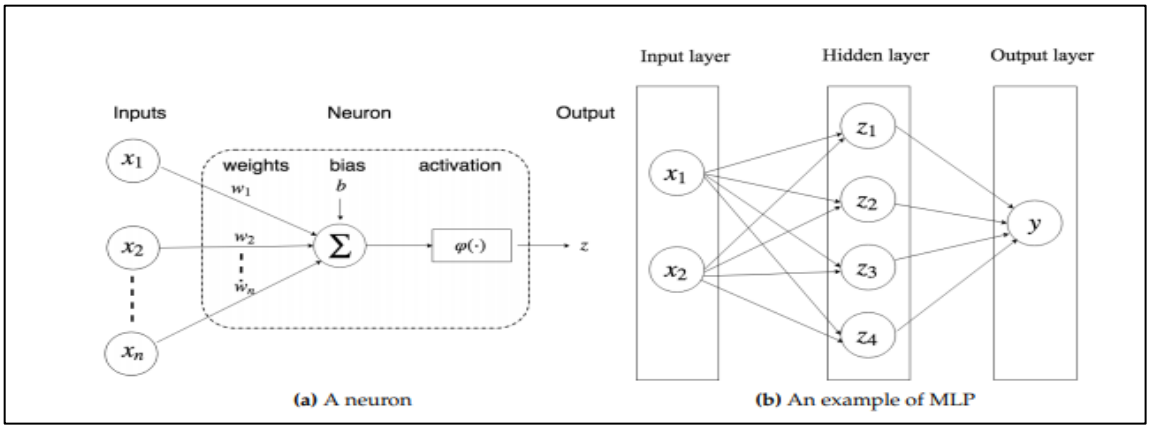

Figure 1.3 MLP Basic Idea

Experiments

First, we used the machine learning algorithm to cluster the best stock groups, then we will focus on the options from this group of the stock. Then we crawl option data from Yahoo finance from 2018 to the end of 2019, and then through data preprocessing, the data is divided into cities one by one, and we also divide the data into put dataset and call dataset. Then we used some features to calculate in addition to the date difference. At the same time, we captured the treasury rate data from the US government's website, compared the date difference, calculated the treasury rate of each data, and finally calculated the value of the BS model. At the same time, we use the sigma we calculated as an important input variable, and we created three models to predict the selling price and selling price of the option at a certain point.

We adjusted the hyperparameters to reduce the error rate and get the final result. Meanwhile we upload all the results to the Tensorboard. Finally, the conclusions calculated by the three models are compared with the BS model, and it is found that our model performs better than the BS model.

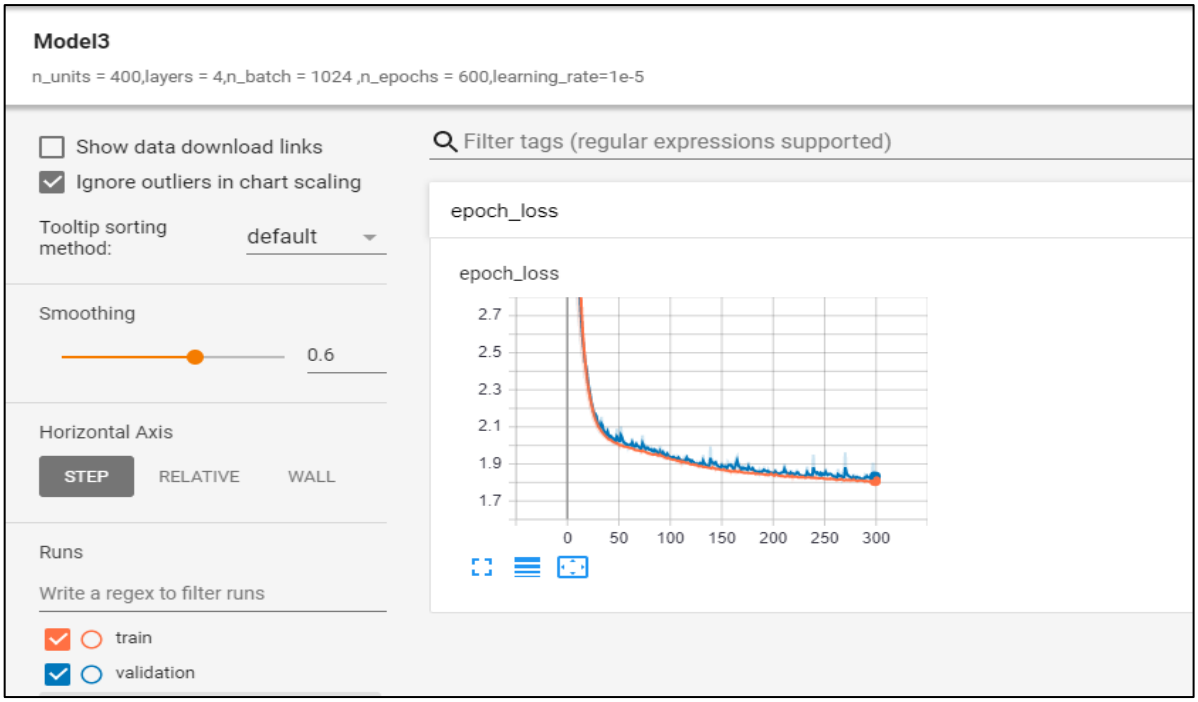

Figure 2.1 TensorBoard

Conclusion

best mse comparision	
model	MSE
LSTM call	0.58
LSTM put	0.6089
Model2 call	0.8284
Model2 put	0.832
Model3 call	0.4618
Model3 put	0.8204
BS Model call	23.2528235
Bs Model put	23.7103045

Fig. 2.2 Best MSE Model Comparison

After we trained many times on the training set, we found that we used the Adam optimizer with the batch size of 1024 and the learning rate at 0.00001 will come up with the best result. We used MSE as our metrics method and we found out that the loss value for model three is the best one.

For future work, we will use the stocks with better Sharpe Rate to find our target option. Meanwhile, we will set up an Options filter to not use the Options deep out of the money, also not using the options expired in 7 days. Also, we will use more different models to train our data and improve the training episode for the data.

References

[1] Alexander ke and Andrew Yang. Option pricing with deep learning. Stanford University, 2019
[2] S. Amornwattana, D. Enke, and C. H. Dagli. A hybrid option pricing model using a neural network for estimating volatility. International Journal of General Systems, 36(5):558–573, 2007.
[3] B. Mezofi and K. Szabo. Beyond black-scholes: A new option for options pricing, Feb 2019
[4] R. Garcia and R. Gençay. Pricing and hedging derivative securities with neural networks and a homogeneity hint. Journal of Econometrics, 94(1):93 – 115, 2000.
[5] M. Malliaris and L. Salchenberger. Beating the best: A neural network challenges the black-scholes formula. In Proceedings of 9th IEEE Conference on Artificial Intelligence for Applications, pages 445–449, March 1993.
[6] U.S. Department of the Treasury. Daily treasury yield curve rates. https://www.treasury.gov/ resource-center/data-chart-center/interest-rates/pages/textview.aspx? data=yield, 2019. Accessed: 2019-10-27
[7] Shuaiqiang Liu 1, Cornelis W. Oosterlee and Sander M.Bohte . Pricing options and computing implied volatilities using neural networks. https://arxiv.org/pdf/1901.08943.pdf .

Supplementary Material

Github Link- https://github.com/zjzsu2000/CMPE297_AdvanceDL_Project