Pricing American Style Stock Options Using Machine Learning

Author: Noah Kriss

Supervisor: Ryan Martin

Introduction

Options trading provides unique opportunities in the finance industry to speculate on market movements and can give unique strategic opportunities to a portfolio. Among the many different types of options, American-style options stand out due to their flexibility, being executable at any point up to their expiration date. This flexibility, gives traders further opportunities for hedging, speculation, and arbitrage.

While there are many proven pricing methods for European-style stock options such as the Black Scholes Method, there are no clearly proven pricing methods for the more flexible American-style options. Current methods such as binomial tree, and Monte-Carlo simulations are computationally intensive and can only be used by large companies with access to massive computational power. This thesis proposes using machine learning in an attempt to create a more efficient pricing method to compete with these larger financial players. The proposed method is to use a Transformer based neural network architecture to find underlying nuances in the options contract behavior.

Background

Financial Concepts

Stock Option: a contract that gives the holder the right, but not the obligation, to trade an underlying asset at a specified price on or up until and a specified date.

Call Option: If the contract allows the holder to make purchases on the underlying asset at the specified price it is called a "call option".

Put Option: If the holder is allowed to sell the underlying asset for the specified price it is called a "put option".

European Style Stock Options: An option contract where the holder can only execute the contract trades on the expiry date.

American Style Stock Options: An option where the holder can execute the contract trades at any point from purchase up until and including the expiry date.

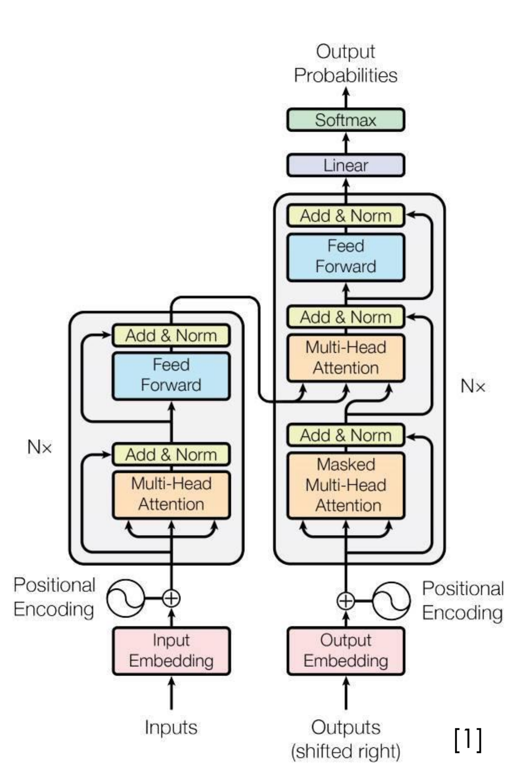
Strike Price: The price specified on the contract that the underlying stock can be bought or sold at.

Interest Rate: The interest rate is the rate money can be borrowed at and the price of not investing money.

Volatility: A metric of how much a trading price varies with time. Higher volatility indicates there will be higher fluctuations in the underlying stock price.

Machine Learning Concepts

A transformer is a type of neural network architecture introduced in the paper "Attention is All You Need" by Vaswani et al. in 2017 [1]. It is the architecture that is responsible for the recent breakthroughs in deep learning power, particularly in natural language processing with models such as ChatGPT and BERT. The Transformer uses a new self-attention mechanism that allows it to capture the importance of different parts of an input, and capture dependencies, contexts, and relationships of each part of the sequence. In addition, each layer in the transformer processes the input data in parallel allowing for faster and more efficient training.



[1]

The Transformer introduced a new attention layer called the "multi-head attention" layer. This layer learns "attention weights" that represent the importance or relevance of different parts of the input sequence in making predictions. These weights are computed in each step of the model's training, allowing it to isolate specific elements or patterns of each set of input data. The attention mechanism learns and applies these weights multiple times in parallel, allowing the model to capture diverse patterns and relationships in the input data while maintaining training efficiency.

Methodology

To price American-style options, this thesis proposes using a multistage neural network based off the Transformer. The network accepts inputs of both sequential stock data and the contract data to find important parameters and underlying nuances in the data.

Data Preprocessing

The input data used for this project is taken from a server created by Professor Martin that scrapes market data daily and saves the prices of options for roughly 40 tickers across a range of strike prices. It also saves other contract data including the days until expiry, current stock price, volatility, open interest, etc. The data from the server is processed into two parts, time sequential data and contract data.

X-Train Sequential Stock Data

The time sequential data contains the last ten end of day stock prices and option prices arranged into a 2x10 array. A rolling window normalization technique is applied to this data, with each row being scaled individually from 0 to 1. This allows the network to find patterns and derivatives in the stock and option price behavior, while being dynamic to any range of stock prices or future market conditions.

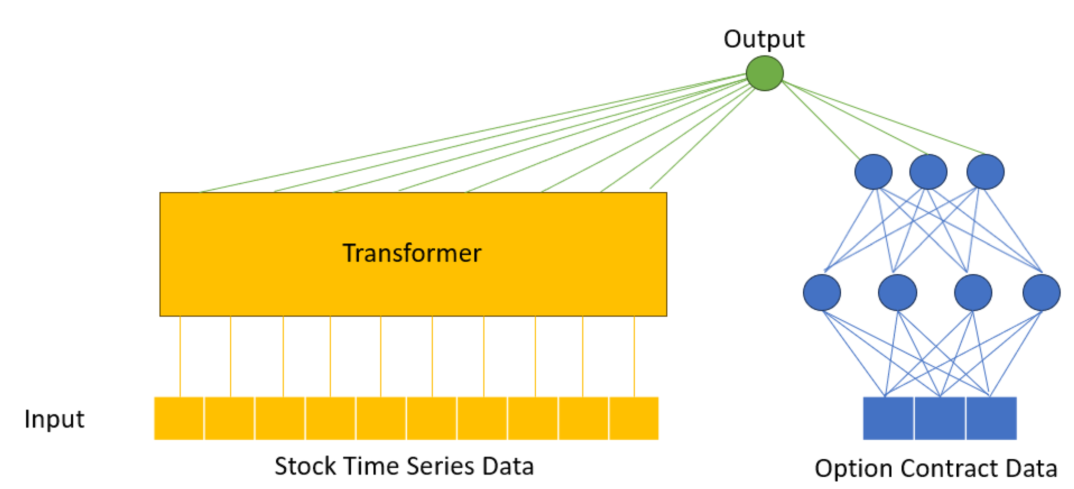
X-Train Option Contract Data

The contract data contains the five inputs to the Black-Scholes model. These five features are added to the array and are un-normalized to preserve relative scale between vectors. This proved to be important for accuracy. This data is unscaled.

Y-Train Output Contract Price

The output of the network is the predicted options price. The y-train and y-test datasets are the average of the bid and ask prices for the day. The "true price" of each contract was estimated to be the average of the bid and ask prices for the day. It was found that the model preformed very poorly at pricing options whose value was zero or near zero. Since these contracts are less interesting to price anyway, all input vectors with options prices below one dollar were removed from the training set.

Network Architecture



Sequential Transformer Network

The time sequential data is fed into a Transformer encoder. The structure of one transformer encoder is as follows:

1. The sequential data is first passed through a normalization layer
2. Next, the normalized data is fed through a multi-head attention layer to learn dependencies and discover patterns.
3. The next layer is a dropout layer to reduce chances of overfitting.
4. The output of this section is added back into the original inputs and is fed into a feedforward section.
5. The feedforward section consists of a normal convolutional layer, a dropout layer, then another convolutional layer, then another dropout layer.

This feedforward section learns more about the patterns and combines them for the final encoding. The above architecture is one transformer encoder, two of these are layered on top of each other to finish the sequential data processing.

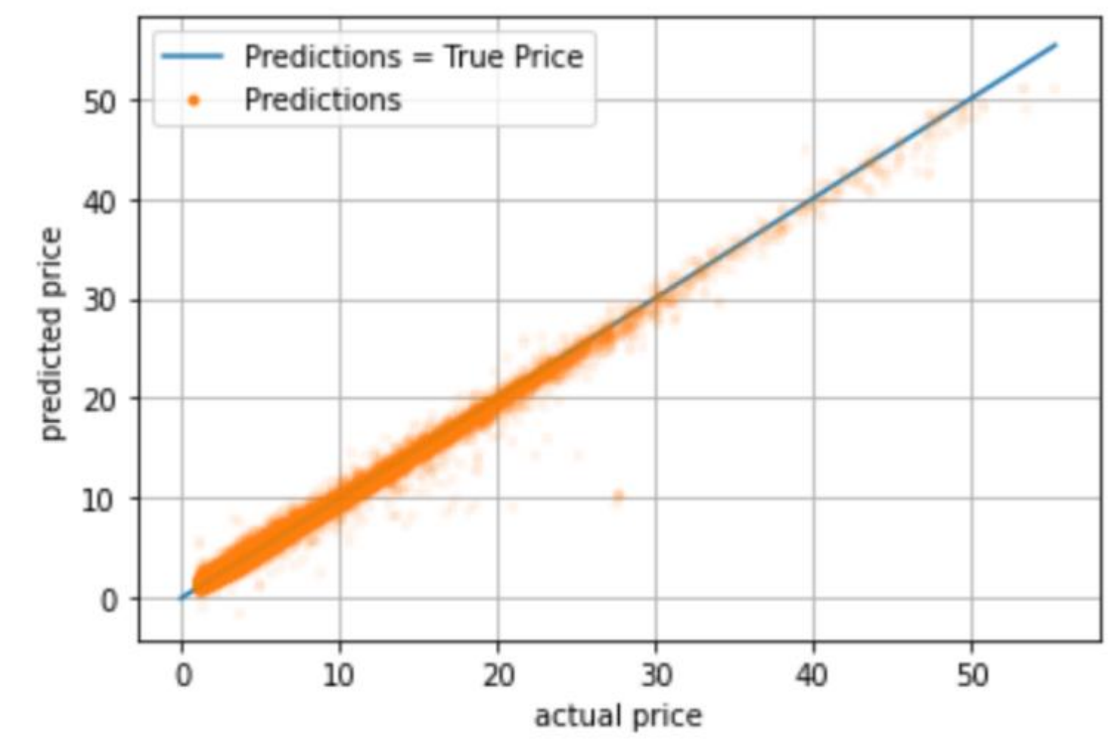
Contract Data Dense Network

The contract data is processed by a dense, feedforward network. The isolated processing of this data is only one layer with 20 nodes. This is sufficient because the processing of this data should only need to be as computationally complex as the Black-Scholes model.

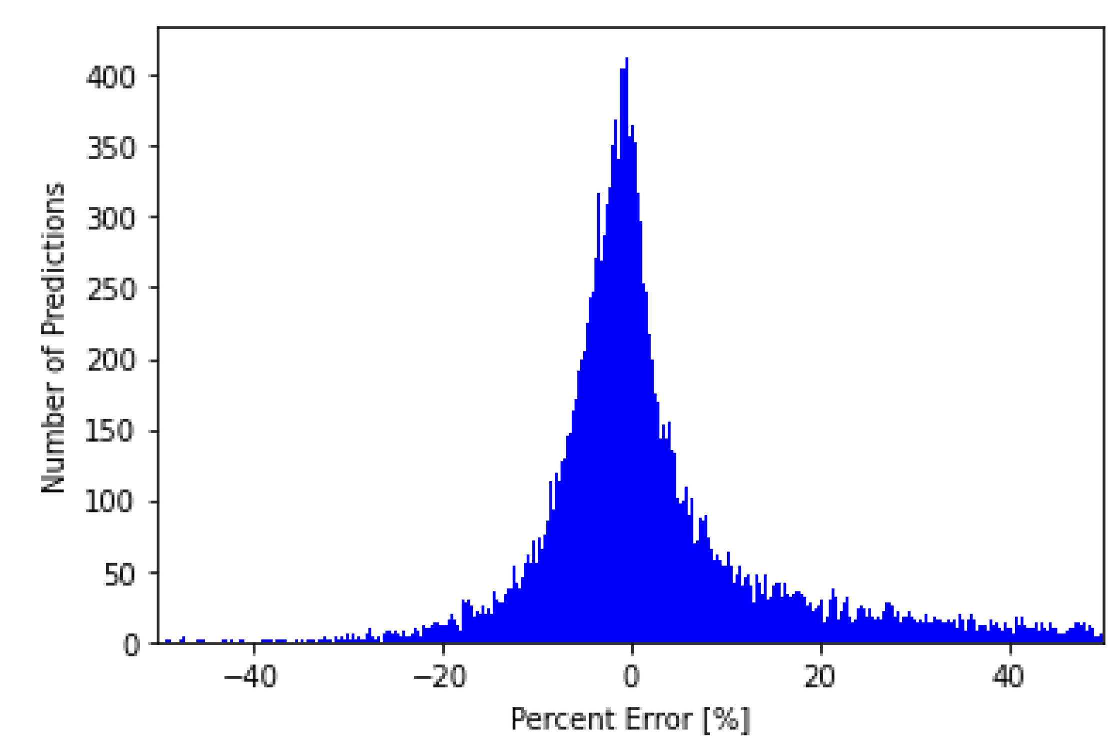
Combining Dense Network

The output of each of these sections contain 20 nodes. It was assumed that the importance of the contract data was more important than the sequential data so to assist the model in understanding this importance a dense layer was inserted on the output of the sequential data to reduce its nodes while maintaining important information. After some testing and iteration, the best balance was found to be half the nodes of the contract data, 10 nodes. These nodes are flattened and concatenated to the 20 nodes from the contract data. The data is then run through another dense feed forward network to produce a final output.

Results



The correlation plot shows that the model is learned patterns between the input data and the output prices. The predictions closely fit the line of perfect predictions with little outliers.

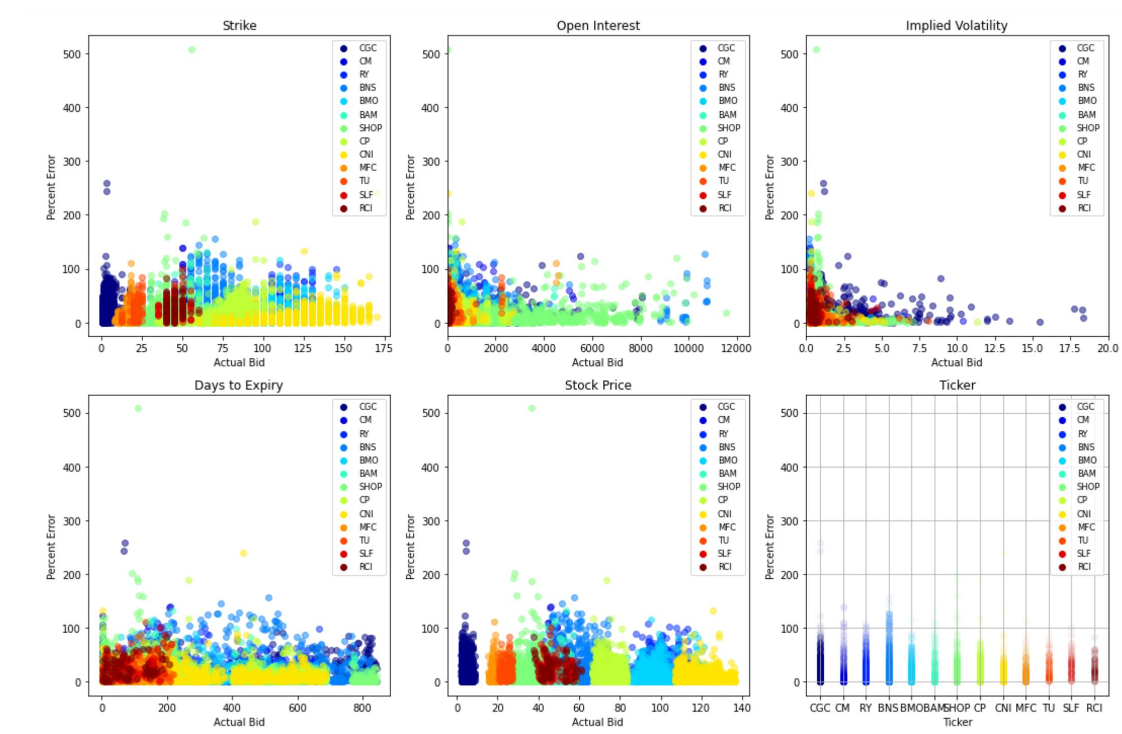


A histogram of the percent error better displays how the model is performing against the test data. The peak is close to center at zero. Overall, the distribution fairly centered with a slight bias towards larger errors being overestimates. The histogram is sharp with most of the predictions falling within ±10%.

The table below gives a summary of the performance metrics. The first column shows the model evaluated on the standard test set and the other column is a generality assessment, evaluating it on the TD options, isolated from the model during training.

Metric	Standard Test Set	TD Test Set
Mean Absolute Error	0.4232	0.6407
Median Absolute Error	0.2757	0.4952
Max Absolute Error	17.8682	4.5652
Mean Percent Error	9.75%	17.49%
Median Percent Error	4.57%	6.75%
Max Percent Error	389.70%	168.67%
Below 10% Error	72.98%	58.95%
RMSE	0.7372	0.8545

The model performs worse against the new contract vectors however it is still making close correlations. More training data on a wider range of tickers and sectors would make the model more general, performing better on a range of tickers and new conditions



By analyzing the behavior of error with respect to each feature, we may be able to understand why the model is falling short. Five of the six categories appear to be independent of model accuracy. For these categories, there does not seem to be a trend in percent error when varying each parameter. The implied volatility does seem to have an error bias as it approaches zero. The price of options tend to decrease with lower volatility because there is less of a chance the underlying stock will fluctuate, becoming in the money. If the volatility is low, the price of the option should be fairly steady because there is less uncertainty in the market.

An interesting observation is that the highest errors are in contracts with low volatility, but tickers that tend to have higher volatility like Canopy Growth Corp (CGC) colored in purple. It is likely that despite the low volatility on this trading day, the market expects it to go back up and their pricing calculations are being adjusted. These instances are rare, so letting the model train off more data with more of these examples could help it learn to price these outliers more effectively.

Pricing Method	Evaluation Speed [s]
Transformer Neural Network	0.000257
Finite Difference PDE	0.61
Binomial Method (Constant Volatility)	7.3
Longstaff-Schwartz Method	139.2

[2]

One of the other key metrics for the model is its computational efficiency. The calculation time for the model to predict a given price is orders of magnitude faster than the other traditional methods. Below is a table comparing the speed to popular pricing methods. It is clear that, should this method become competitively accurate to the other methods, it would be quickly adapted due to opportunities for arbitrage.

Conclusions

This thesis proposed a Transformer based neural network architecture as a method to price American-style stock options and compete with more computationally demanding methods only possible for large financial companies. The network proved that it could make consistent correlations within 10% of the true price at a rate of 73%. The Transformer portion of the network proved to significantly improve accuracy boosting the predictions within 10% by 22.3%. One of the model's biggest successes was that it can make predictions on the price over 2000 times faster than other pricing strategies.

While the model is not yet accurate enough to be used in an industry setting, the correlation and efficiency indicate that the model could be a very powerful pricing method. Once the accuracy of the model is competitive with other pricing strategies it would excel against competition due to its drastic prediction efficiency. The analysis indicates that the major next step to improve the model accuracy would be increasing the training dataset size allow the transformers to learn deeper correlations. A larger dataset would also allow for more transformer encoders, which proved to increase accuracy, and make the model more general, able to predict across broader range of sectors and market conditions.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in 2017 Neural Information Processing Systems (NIPS), 2017.

[2] D. R. Anderson and U. Ulrych, "Accelerated american option pricing with deep neural networks," Quantitative Finance and Economics, vol. 7, no. 2, pp. 207–228, 2023.