Predicting Option prices using Deep Learning Networks

Viraj Thakkar

Contents

1 Introduction

Derivatives are financial instruments i.e a contract or an agreement which "derives" its value based on the performace of an underlying asset for instance, a stock. Derivatives are important for the purpose of risk management as they offer protection against an adverse change in the market condition e.g. the COVID-19 pandemic. Many investors and traders use derivativies to reduce their exposure to sudden market movements(hedging) and to avoid big losses. Hence, it is important to study derivaties and how they are priced based on the market conditions and asset pirce.

2 Problem overview and Model

In this project, we will work under the assumptions of the Black-Scholes model [1], [2]. It is a widely used mathematical model in the financial markets with some modern adjustments. We will be using the model to estimate the price of European-style call options. We will simulate data for various input features of the Black-Scholes formula and the target variable of option prices. Finally, we use a deep learning network architecture to learn the option prices using the simulated data and predict the option prices on the test data set i.e. essentially a regression analysis [3], [4].

2.1 Black-Scholes Model

The Black-Scholes Model assumptions that are relevant to this project are:

- The stock price follows a Geometric Brownian motion path.
- The stock does not pay any dividend
- The risk-free interest rate is known and constant.
- The volatility is known and is constant.

The Black-Scholes second order partial differential equation is of the form

$$\mathbf{\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial^2 S} + rS\frac{\partial V}{\partial S} - rV = 0}$$

where $\mathbf{V(S,T) = max(S - K, 0)}$ which basically governs the dynamics of the value function $\mathbf{V(S,T)}$. This is similar in nature to the heat equation used in physics. The famous Black-Scholes formula to obtain option price is the solution of this partial differential equation and is stated below:

$$\mathbf{C = SN(d_1) - Ke^{-rT}N(d_2)}$$
$$\mathbf{d_1 = \frac{ln(S/K) + (r - 0.5\sigma^2)T}{\sigma T}}$$
$$\mathbf{d_2 = d_1 - \sigma\sqrt{T}}$$

where $\mathbf{C}$ = **Call option price**, $\mathbf{S}$ = **Stock price**, $\mathbf{K}$ = **Strike price**, $\mathbf{r}$ = **Risk-free interest rate**, $\mathbf{T}$ = **Time to maturity**, σ = **Volatility**, and $\mathbf{N}$ = **Standard normal cdf**.

3 Methodology

3.1 Monte-Carlo simulation of Stock prices

Let $\mathbf{S_t}$ denote the price of the stock at time $\mathbf{t}$. For a stock following Brownian motion, we have:

$$\mathbf{S_{t+\triangle t} - S_t \sim \mathcal{N}(0, \triangle t) = \epsilon\sqrt{\triangle t}}$$

For a stock following Geometric Brownian Motion(GBM), the fractional change in stock price is given by:

$$\mathbf{\frac{\triangle S}{S} = \mu \triangle t + \sigma\epsilon\sqrt{\triangle t}}$$

where $\epsilon = \mathcal{N}(\mathbf{0}, \mathbf{1})$, μ is drift (risk-free rate) and σ is volatility. See Fig.1. Starting with an initial stock price of $S_0 = \$100$, we add incremental changes of $\triangle S$ to S_t for each $\triangle t$ time step of size=1 day. We run many such simulations with r and σ as constant parameters. See Fig.2.

3.2 Dataset simulation

For various values of r and σ, we simulated GBM stock prices as descibed in the previous section. More combinations were taken for various values of Strike price S and time to maturity T. The full range of values taken for these variables is shown in the table below. A total of 15 million such samples were simulated and call option prices(target variable) was calculated for all the samples using the Black-Scholes formula. As all the conditions of the Black-Scholes model were

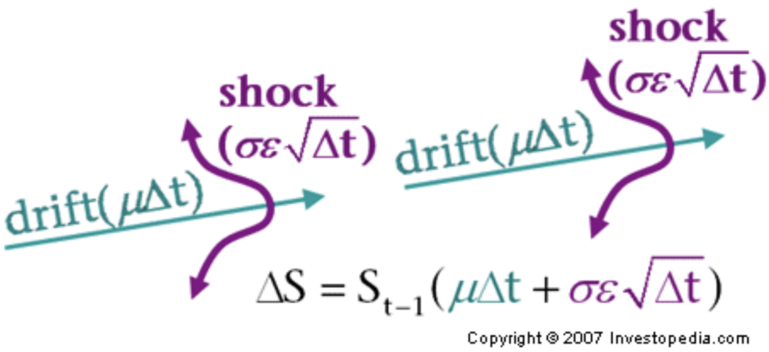

Figure 1: Representation of stock price changes through time following GBM. Figure source [5].

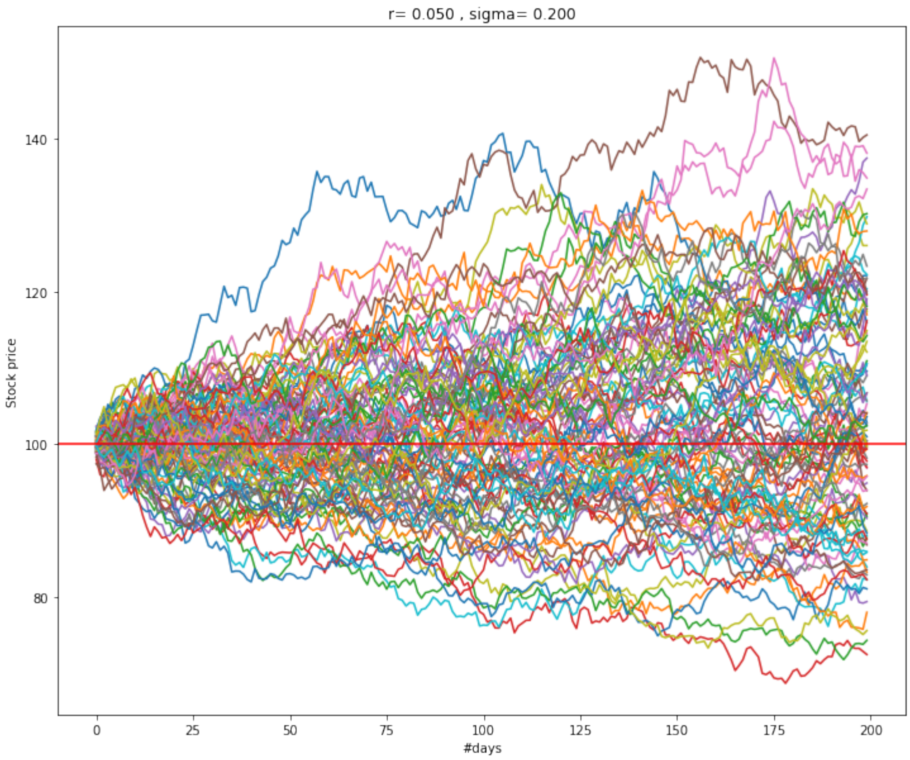

Figure 2: Monte-Carlo simulation of stock price following GBM starting at $S_0 = \$100$ for 100 such simulations.

satisfied while simulating the dataset, these call option prices calculated represent the true values of call option in our project.

Variable	Range
r (Risk-free rate)	[0.01-0.1]
σ (Volatility)	[0.05-0.6]
T (Time to maturity)	10 days- 1.5 years
K (Strike price)	[60-150]

3.3 Deep Neural Network Implementation

1.5 million data points were randomly selected (f10% of the population) for machine learning purposes with train, validation and test split ratio of 60:20:20. The 5 different features namely S, K, r, σ and T can be reduced to 4 features by using the following property of the Black-Scholes formula.

$$\mathbf{C}(\alpha\mathbf{S}, \alpha\mathbf{K}) = \alpha.\mathbf{C}(\mathbf{S}, \mathbf{K})$$
$$\mathbf{C}(\mathbf{S}, \mathbf{K})/\mathbf{K} = \mathbf{C}(\mathbf{S}/\mathbf{K}, \mathbf{1})$$

where α is a constant. Hence, after making this feature transformation, we have our 4 features as S/K, r, σ and T with the target variable as C(S/K,1).
A dense neural network architecture was implemented using keras[6] with the following structure: 1 Input Layer: 4 nodes, 3 Hidden Layers: 250 nodes each and 1 Output Layer: 1 node(continuous output ranging from 0 to ∞). ReLU and Leaky ReLU activation functions were used to fit the highly non-linear dataset [7], [8]. Droupout rate was set to 0.2 to avoid overfitting of data [9]. The loss function used was Mean Square Error(MSE).

4 Results

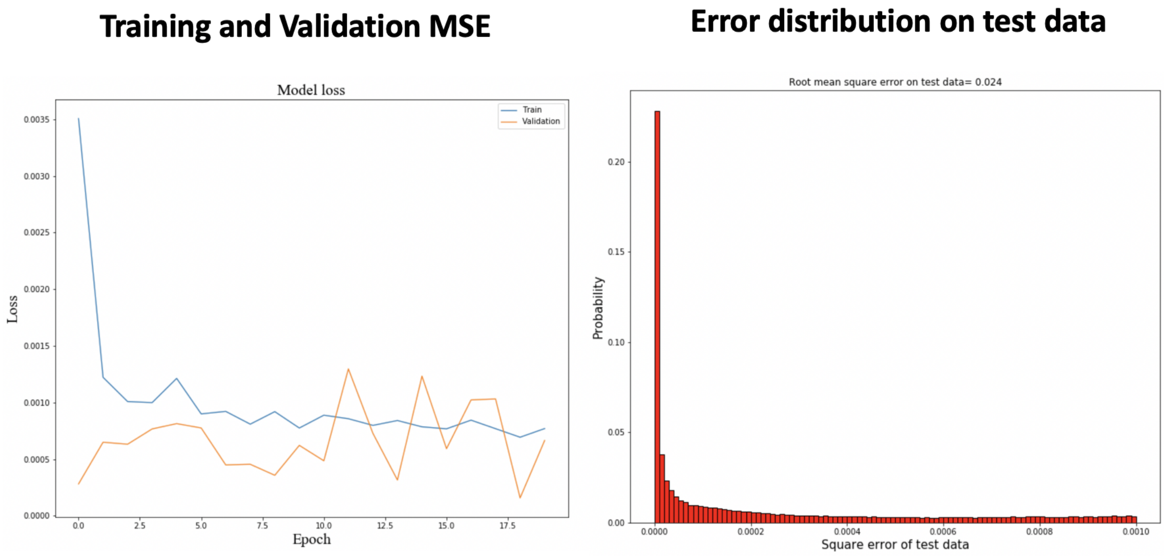

Figure 3: Left: Learning curve for MSE on train and validation data set. Both seems to converge after 15 epochs. Right: Squared error distribution on the test dataset. As seen in the figure, most of the error is very small. The y-axis is normalized to show probability.

The call option prices can be recovered back from the target variable C(S/K,1) by simply multiplying it by its corresponding strike price K. We can make a plot of predicted option price(by deep learning networks) vs true option price(obtained by Black-Scholes) to check how well the algorithm has performed. The result is a nearly perfect straight line of y=x as seen in Fig.4 (Left y-axis). The root mean square error of the target variable on the test dataset= 0.024. This means that for a Strike price of \$1, the error in prediction of the option price is around 2.4 cents.

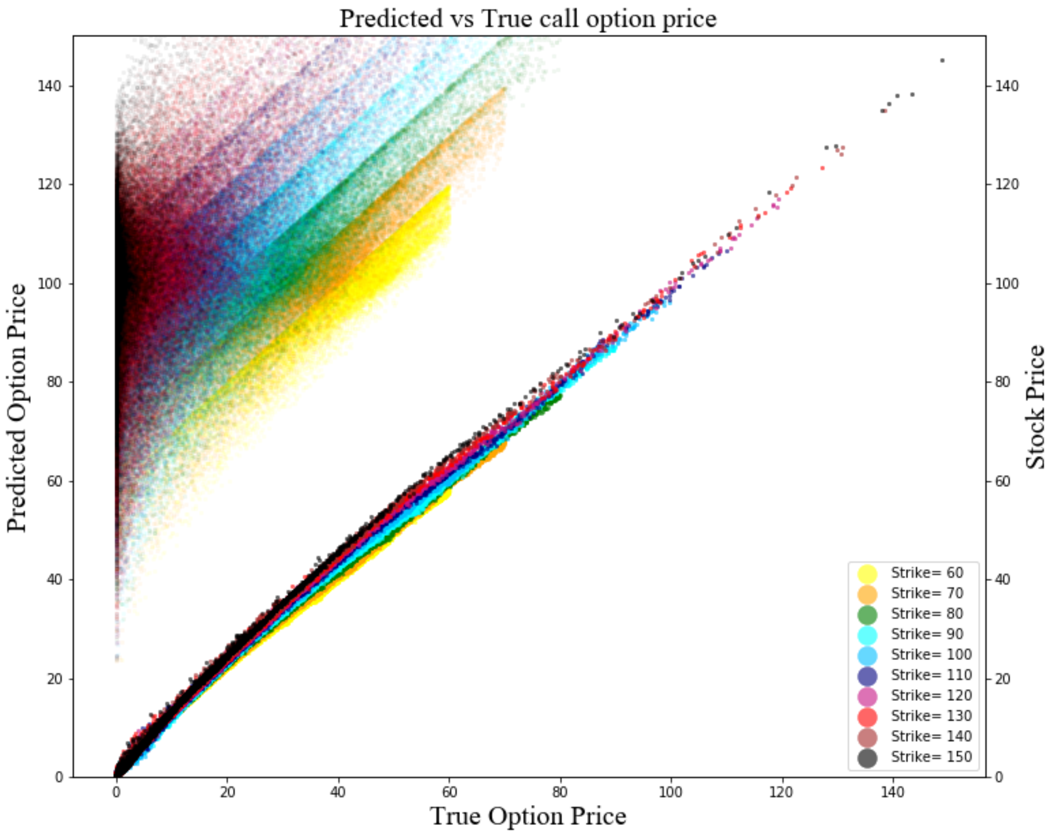

Figure 4: The figure consists of two y-axis and x-axis is common. As explained earlier the plot of predicted vs true option price is almost a straight line y=x. The upper left corner color shaded part of the graph represents the Stock price(right y-axis) at that given option price. All the data points are color coded with thier corresponding Strike prices which is shown in the legend. **Intuition:** →Let's have a look at the black points which have the highest value of strike price =150. As the stock prices shown in the plot are below 150 and the stike price for black points is 150, the option value will be priced very low because they are out of the money. Hence most of the black points are concentrated near the low option price region. → Let's have a look at the yellow points which have the lowest strike price=60. As most of the stock prices in the figure are above 60, the option value will be priced higher because they are in the money. Hence, most of the yellow points are spread out in the higher option prices range.

5 Discussion and Outlook

From this project we demonstrated that deep neural networks can be used effectively in financial markers for tasks like option pricing. Although the predictions were obtained on the simulated dataset which followed the Black-Scholes model, the deep learning model was able to capture the non-linearity in the date and make successful predictions. Hence, the deep learning model can be further trained and implement on real option prices market data which do not strictly follow those ideal assumptions in the Black-Scholes model.

6 Appendix

- Call Option: It gives the buyer the "option" to buy a stock on a future date at a specific price (strike price). Hence, we are hoping for the Stock price to rise. Example: Stock price is \$110 at present. Call option costs \$12 at a Strike price of \$100. If Stock price rises to \$150 by the time of expiration, we can earn a profit of 150-100-12=\$38
- Stirke Price: The price at which the asset in the contract is borought or sold.
- European Style: The right to exercise the option is at the expiry date.
- Volatality σ: The standard deviation of the logarithmic returns on the stock. It measures how quickly the market changes over time.
- Risk-free rate r: Rate of hypothetical return on an investment with no risk of loss.
- In the Money: Strike price is less than stock price for call option. We can make money directly by exercising the option.
- Out of the money: Strike price is more than stock price for call option.

Github: https://github.com/virajthakkar/Option-prices-deep-learning

References

[1] https://www.investopedia.com/terms/b/blackscholes.asp

[2] https://en.wikipedia.org/wiki/Black\OT1\textendashScholes_model

[3] Machine Learning in Finance: The Case of Deep Learning for Option Pricing Robert Culkin & Sanjiv R. Das Santa Clara University August 2, 2017

[4] Pricing Options with an Artificial Neural Network: A Reinforcement Learning Approach, Haakon A. Trønnes, Department of Economics, Norwegian University of Science and Technology, June 2018

[5] https://www.investopedia.com/articles/07/montecarlo.asp

[6] https://keras.io/

[7] https://www.machinecurve.com/index.php/2019/09/04/relu-sigmoid-and\penalty-\@M-tanh-todays-most-used-activation-functions/#rectified-linear-unit-relu

[8] https://towardsdatascience.com/exploring-activation-functions-for-neural-networks-73498da59b02

[9] https://machinelearningmastery.com/dropout-regularization-deep-learning-models-keras/