Neural Black-Scholes Model

Aaditya Bhashyam

July 25, 2025

1 Introduction

This report introduces a Neural Black-Scholes Model, a hybrid, grey-box modelling approach, through which we aim to overcome the shortcomings faced by the classical Black-Scholes model. The goal is to incorporate a Neural Network corrective term into the Black Scholes partial differential equation (PDE), solve it backwards in time, to show that it can improve upon the predictions of the classical Black-Scholes model.

2 Background

2.1 Black-Scholes

The Black-Scholes Model, first introduced by Fischer Black and Myron Scholes, is a mathematical framework for pricing options. It starts with the assumption that the underlying stock follows a Geometric Brownian motion, and by making theoretical asssumptions and enforcing a no-arbitrage condition, arrives at a closed form solution which allows you to fairly price options.

The Black-Scholes formula for a European call option is given by:

$$C(S,t) = S\Phi(d_1) - Ke^{-r(T-t)}\Phi(d_2)$$

where

$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)}{\sigma\sqrt{T-t}}, \quad d_2 = d_1 - \sigma\sqrt{T-t}$$

and the parameters are:

- S: Current price of the underlying asset
- K: Strike price of the option
- T: Time to maturity
- t: Current time
- r: Risk-free interest rate
- σ: Volatility of the underlying asset
- $\Phi(\cdot)$: Cumulative distribution function of the standard normal distribution

2.2 Heston Model

The Heston Model is an improvement upon the Black-Scholes model, which assumes that the volatility in the Geometric Brownian Motion term itself follows its own stochastic differential equation, as follows.

$$dS_t = \mu S_t \, dt + \sqrt{v_t} S_t \, dW_t^{(1)}$$

$$dv_t = \kappa(\theta - v_t) \, dt + \xi \sqrt{v_t} \, dW_t^{(2)}$$

where:

- μ: Drift rate of the asset
- κ: Rate of mean reversion of the variance
- θ: Long-term mean of the variance
- ξ: Volatility of the variance ("vol of vol")
- $W_t^{(1)}$, $W_t^{(2)}$: Correlated Brownian motions with correlation ρ

3 Methodology

3.1 Data Generation.

The Black-Scholes PDE is solved backward in time for a single option, i.e., over a grid of (S, t) values. The resulting data is a grid of potential option values between the present time ($t = 0$) and the time of maturity ($t = T$).

To demonstrate that the Neural Black-Scholes Model can compete with the classical Black-Scholes Model, the training data must exhibit complexities that the classical model cannot capture.

Therefore, the data is generated using the Heston Model. For each point on the grid, we perform Monte Carlo simulations forward in time using the Euler–Maruyama discretization scheme and discount the outcomes backward to compute the option price. This yields a grid of option values. The simulated data inherently contains more complexities than what the classical Black-Scholes Model can account for.

3.2 Application of Method of Lines to Black Scholes PDE.

In employing the neural ODE structure, we use `odeint` of `torchdiffeq`. The paradigm of the package is such that it takes in a vector of initial conditions and integrates forward in time, and returns a vector of solutions at a given time point.

Thus, we must apply the concept of Method of Lines to the Black-Scholes PDE in order to solve it using the `odeint` package. The Method of Lines deals with the discretisation of all variable dimensions except one, i.e time. Furthermore, we replace the discretisation process across the stock dimension by obtaining the gradients, i.e $\frac{\partial V}{\partial S}$ and $\frac{\partial^2 V}{\partial S^2}$, from the grid of points we previously generated. It is important to note that the gradient values change with both time and stock price. Subsequently, the solver deals with the discretisation across time.

The discretisation equation for a single point in the grid is :

$$\frac{dV_i(t)}{dt} = \frac{1}{2}\sigma^2 S_i^2 \left.\frac{\partial^2 V}{\partial S^2}\right|_{i,t} + rS_i \left.\frac{\partial V}{\partial S}\right|_{i,t} - rV_i(t) = 0$$

The discretisation equation for a single time slice is:

$$\frac{d\mathbf{V}_t}{dt} = \frac{1}{2}\sigma^2 S_i^2 \left.\frac{\partial^2 \mathbf{V}}{\partial S^2}\right|_{t} + rS_i \left.\frac{\partial \mathbf{V}}{\partial S}\right|_{t} - r\mathbf{V}_i = 0$$

Where, if the grid size of stock price and time is $n \times n$, then $\mathbf{V}$, $\left.\frac{\partial^2 \mathbf{V}}{\partial S^2}\right|_t$, $\left.\frac{\partial \mathbf{V}}{\partial S}\right|_t$ are matrices of size $n \times 1$.

The initial conditions provided to the solver are the undiscounted payoffs at the time of maturity, or in the case of having to split the data, the option values at the time slice at which the split occurs.

We first solve the Black-Scholes PDE backwards in time, without the neural network corrective term and compare the solutions with the predictions of the classical Black- Scholes, to ensure that the scheme of the solving the Black-Scholes PDE is correct. To corroborate the validity of the scheme, for each grid of solutions that are constructed, we construct a heatmap of the residuals i.e the error between the solutions as per the Method of Lines scheme and the solutions as per the classical Black-Scholes model. An instance of which is given in the image below.

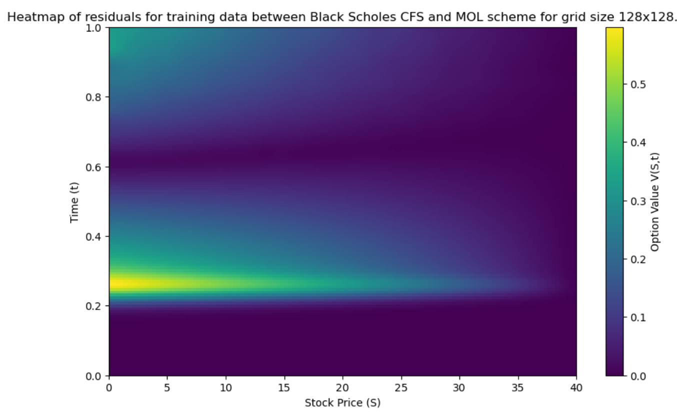

Figure 1: A heatmap of the residuals between the classical Black Scholes and MOL scheme.

Having ensured that the scheme is correct, we move on to the adding the neural network corrective term like so:

$$\frac{d\mathbf{V}_t}{dt} = \frac{1}{2}\sigma^2 S_i^2 \left.\frac{\partial^2 \mathbf{V}}{\partial S^2}\right|_{t} + rS_i \left.\frac{\partial \mathbf{V}}{\partial S}\right|_{t} - r\mathbf{V}_i = \mathcal{N}_\theta(\sigma, S_i, t)$$

3.3 Model Construction - Training, Validation and Testing

The neural network in the performed experiments, is a simple Artificial Neural Network. The training scheme is simple. For the purposes of training the model, we generate the data across grids of various resolutions from 16x16 to 256x256. Each grid is further split into an 80:20 ratio for training and testing. Across resolutions, the stock prices vary from [0, 40], the time varies from [0,1] and the strike price is 10.

Lastly, another grid is created with resolution 512x512, however the stock prices vary from [5, 50], and the strike price is 15 and the time takes on the same range. This serves as the validation set, to prove that the model is not overfitting.

4 Results and Analysis

The results of the experiment are summarised in Table 1.

Grid Size	Train Data Verification	Validation Data Verification	Train Loss	Validation Loss	Test Loss	Black-Scholes CFS Test Loss
16x16	MSE FDM–Heston: 22.1651 MSE CFS–Heston: 21.8112 MSE CFS–FDM: 0.1444	MSE FDM–Heston: 33.4818 MSE CFS–Heston: 34.4964 MSE CFS–FDM: 0.0555	0.5725	2.8715	0.2254	10.4432
32x32	MSE FDM–Heston: 21.4108 MSE CFS–Heston: 21.0995 MSE CFS–FDM: 0.1440	MSE FDM–Heston: 33.4818 MSE CFS–Heston: 34.4964 MSE CFS–FDM: 0.0555	0.4772	2.3221	0.1539	10.6708
64x64	MSE FDM–Heston: 21.4265 MSE CFS–Heston: 21.1251 MSE CFS–FDM: 0.1434	MSE FDM–Heston: 33.4774 MSE CFS–Heston: 34.4974 MSE CFS–FDM: 0.0555	0.2604	1.3420	0.0965	10.7623
128x128	MSE FDM–Heston: 20.9261 MSE CFS–Heston: 20.6579 MSE CFS–FDM: 0.1438	MSE FDM–Heston: 33.4774 MSE CFS–Heston: 34.4974 MSE CFS–FDM: 0.0555	0.3737	1.8341	0.1228	10.6152
256x256	MSE FDM–Heston: 20.8468 MSE CFS–Heston: 20.5826 MSE CFS–FDM: 0.1443	MSE FDM–Heston: 33.4774 MSE CFS–Heston: 34.4974 MSE CFS–FDM: 0.0555	0.3103	1.6349	0.1133	10.5445

Table 1: Summary of Results

There are two significant observations to be noted:

1. The Neural Black-Scholes Model outperforms the classical model (after training) across all grid sizes, given the constraints and specific conditions of the experiment.

2. Despite training on coarser grids, the Neural Black-Scholes model (after training) tends to perform exceptionally well on finer grids (as evidenced by the validation loss.)

5 Conclusion

The concept of introducing a neural network corrective term into the Black-Scholes PDE has worked well and sparks interest into whether the same idea can be extended to other PDEs, better yet SDEs, of finance.