Part B: Fast price approximator to time dependent derivatives

I – Introduction

1) Objective

Pricing complex derivatives such as Asian Options or Chooser Options can be complicated and can require a lot of time. Indeed, some of complex derivatives cannot be priced analytically and could require numerical approaches such as the Monte Carlo simulation. However, pricing with such models can be extremely time-consuming and therefore are not adapted to some financial institutions where time is key. Hence, building alternatives models to price such complex derivatives are interesting.

This is in this sense that we are going to try to build a Deep Learning model to price Asian Options.

2) Idea

To do so, we are going to generate a large set of data with various parameters (volatility, rates, stock prices, strikes and maturities). Then we are going to build a neural network and train it on our data to fit on Monte Carlo Prices and finally compare our results and draw conclusion on the effectiveness of our model.

II – Construction of the model

1) Data Generation

To have data to work on, we generated more than one million Monte Carlo Simulation with 1,000 paths with various volatilities, rates, maturities, initial stock prices and strikes and then compute Asian Option Call prices. And then stored the data into a csv (for next users) and into a Pandas Data Frame which will allow us to train our neural network. In our cases we generated exactly 1,152,000 sample prices of Asian Call Options with stock prices varying from 1 to 200 with a step of 4, the interest rates ranged from 0 to 0.08 with a step of 0.01, and the volatilities ranged from 0 to 0.9 with a step of 0.05, the strike prices varying from 1 to 400 with a step of 5, and maturity set to 6 months and 1.

Then in order to train on our model on the data, we standardized and split randomly the data into three parts (training set, validation set and test set) with the following ratio: 70/15/15.

2) Model Construction

We aimed to perform a regression on our data to get Asian Option Call prices with a neural network. The dependent and independent variables are set as follow:

$$X = [\,Stock\ Prices,\quad Interest\ Rates,\quad Volatilities,\quad Maturities,\quad Strike\ Prices\,],\quad Y = [MC\ Prices]$$

During the construction using TensorFlow, we focused on building a Neural Network to facilitate regression analysis. Throughout the process, we extensively experimented various architectural configurations, including multiple hidden layers and diverse activation functions in our case we chose the ReLu activator and the SoftMax activator. Then, to monitor the model's performance, we tracked accuracy across epochs utilizing the Mean Squared Error (MSE) metric, allowing us to gauge the effectiveness of the model's predictions. Also, to optimize the training process, we implemented the Adam optimizer which dynamically adjusts the learning rate to accelerate convergence while preventing overshooting of the minimum. Additionally, we incorporated a variable learning rate schedule with the Keras' Exponential Decay function, ensuring adaptive adjustments to the learning rate over the course of training, leading to improved convergence and enhanced overall performance.

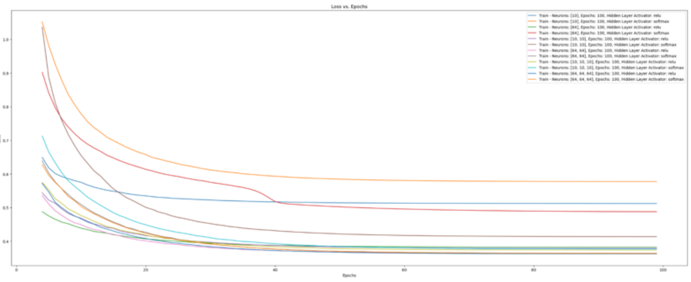
Notably, to ensure non-negative predictions, a crucial requirement in the context of derivative pricing, we opted for the softmax activation function in the output layer, owing to challenges we strangely encountered with convergence when using the ReLU activator.

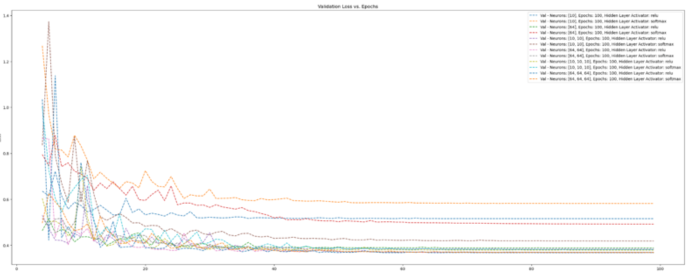
3) Hyperparameter tuning

During the development of our model, we dedicated strong efforts to the critical process of hyperparameter tuning. Our focus was primarily around the selection of appropriate activation functions for the hidden layers through the ReLU and SoftMax functions. Additionally, we explored various layer architectures, experimenting with different configurations such as single-layer structures with 10 and 64 neurons, as well as multi-layer structures like 10x10, 64x64, 10x10x10, and 64x64x64. Then, to evaluate the efficacy of these models, we tracked and compared the Mean Squared Error (MSE) values across multiple epochs, on both the Training Set and the Validation Set. This systematic comparison enabled us to identify the most efficient model architecture while preventing underfitting, providing us with valuable insights into the optimal hyperparameter configuration necessary for accurate and reliable predictions in the context of Asian option call price regression.

In our extensive exploration of model architectures and activation functions, we made some noteworthy discoveries. We observed that single-layer models performed relatively poorly when compared to multi-layer structures, indicating the importance of depth in our network. Interestingly, in specific architectures like 64x64, the SoftMax activation function yielded promising results, although, on average, the ReLU activation function outperformed others. Across all tested configurations, our models consistently achieved a Mean Squared Error (MSE) of approximately 0.37.

The initial tests on both the Training Set and the Validation Set led us to identify the 64x64x64 architecture with ReLU activations on hidden layers as the best-performing model. Subsequently, during the final evaluation on the test dataset, this same model continued to demonstrate its superior predictive power with a MSE of 0.364 on the Test Set. As a result, we have decided to adopt the 64x64x64 architecture with ReLU activations for computing Asian Call Options, and we look forward to comparing its performance to the Monte Carlo method. This selection was made based on the model's consistent and robust performance, and we believe it will serve as a dependable tool for our pricing analysis.

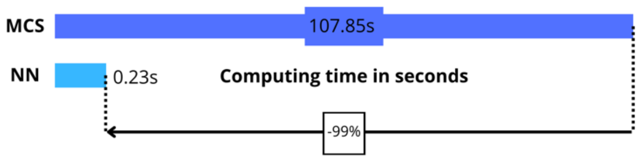


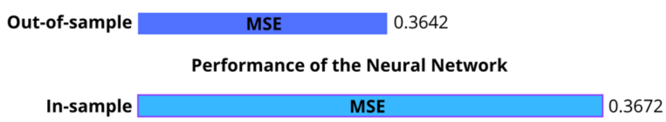


III – Conclusion

1) Comparison with Monte-Carlo Pricing

Following our rigorous analysis, we undertook a thorough comparison between our chosen 64x64x64 model with ReLU activations and the traditional Monte Carlo pricing method for Asian Call Options. Our investigation revealed a significant disparity in computation times, with the neural network requiring a mere 0.23 seconds to compute 10,000 derivative prices, as opposed to the 107.85 seconds needed by the Monte Carlo method. This reduction of 99% in pricing time signifies a substantial improvement in efficiency and highlights the computational prowess of our neural network model. Additionally, our thorough examination unveiled that our model consistently produces predictions that fall within the 95% confidence interval of the Monte Carlo prices, underlining its accuracy and reliability. Lastly, the model's R-Square value of 0.99971 further emphasizes its relevance and robust predictive capabilities, solidifying its position as a powerful and effective tool for Asian Call Option pricing. This extensive validation process reaffirms the strength and utility of our developed neural network model.





2) Uses and Limits

Our developed neural network model holds immense potential and practical applications, particularly fulfilling the needs of front office users (sales and trading professionals). Its capability to instantly generate indicative prices for clients and facilitate rapid pricing of exotic derivatives positions makes it a valuable asset in the fast-paced and dynamic of financial markets.

Nonetheless, it is imperative to acknowledge the inherent limitations of the model. The lack of transparency regarding the precise mechanics of the model, coupled with the availability of only feature importance, renders it a 'Black Box' for risk managers and other users, which could potentially pose challenges and raise concerns in the future. Furthermore, despite its robust predictive power, it is crucial to note that the model's application for trading purposes may not be advisable, as it could inadvertently lead to the emergence of arbitrage opportunities in the market. Thus, while our model represents a significant advancement in pricing efficiency, it is essential to proceed with caution and maintain a comprehensive understanding of its limitations and potential implications, especially concerning risk management and market dynamics.