Calibration Techniques of Fractional Black Scholes Model

Athul AR[1], Richard Obonyo[2]

[1]*Worldquant University*
[2]*Worldquant University*
[1]athular7@gmail.com , [2]richardobonyo12@gmail.com

Abstract

The Black Scholes model, despite being the dominant derivative pricing model in the financial analyst's toolkit, has failed in numerous aspects. Fractional alternatives, based on the Fractal Market Hypothesis has been gaining foothold. However, calibration techniques are still at a primitive stage in this domain. We aim to leverage the power of neural networks and other advancements in the calibration techniques to make the model more effective and efficient to use

Keywords: *Fractional Brownian Motion, Neural Networks, Invertibility, Calibration*

1. Introduction

1.1 Problem Statement

The most widely used approach to model price movement is the Black Scholes model. Although quite easy to use, the model has been determined to be inadequate in predicting extreme market variations due to its numerous simplifying assumptions and choices. The Black Scholes model failed to predict black swan events like the economic crisis of 2008. Despite these shortfalls, the Black Scholes model is widely used as the primary modelling tool in finance, notably for pricing financial derivatives.

Benoit Mandelbrot, a French mathematician, noticed that option prices were not normally distributed and did not follow a Gaussian distribution as assumed by Black and Scholes but actually reflected access kurtosis with flat tail tendencies and the price distributions exhibited both long and short memory of price trends. Price movements assume α-stable and fractional brownian motion distributions with memory and could therefore be used to better predict extreme variations and movements of prices in the market.

The effectiveness of the fractional Black Scholes model based on Mandelbrot's theory requires accurately determining model parameters like volatility and Hurst exponent; which indicates the memory trends. These parameters are presently not well modelled as it is determined from time series data of financial instruments and this affects the predictive accuracy of the model. Correctly modelling and determining these parameters will allow the better calibration of the model which will improve price prediction, reduce losses incurred from wrongful pricing of derivatives, enable better anticipation and prediction of market crises and black swan events and help generate profitable trading strategies based on financial derivatives.

1.2 Goals and Objectives

1.2.1 Goal

To study existing estimation and calibration techniques, design and implement tractable approaches for calibration of heuristic fractal models.

1.2.2 Objectives

- To study and review existing literature on fractal based price modelling methods for pricing of options.
- To design techniques and methods to help us better understand and model price movements in pricing models based on fractal Brownian motion as proposed by Mendelbrot and other researchers.
- To study the parameter(s) of the fractal brownian motion based pricing models and determine how to better predict the parameter(s) using models like Artificial neural networks, linear models and machine learning.
- To test the effectiveness and reliability of the designed model in predicting option prices by comparing our predictions with real market option price data sets.
- To compare, validate and calibrate the fractal Brownian motion pricing model with the Black Scholes pricing model and market prices so as to determine and improve the effectiveness in modeling option prices.

1.3 Literature Review

1.3.1 Fractional Black Scholes Model And Pricing Options Under The Fractional Brownian Motion

The dominant economic theory in the field of financial modelling is Black Scholes model backed up by ideas like Efficient Market Hypothesis. It was based on numerous simplifying but restrictive assumptions like efficient markets, perfect liquidity, rational agents, normal price distribution etc. Fractals, a brainchild of Benoit Mandelbrot came as a radical alternative, relaxing many erroneous assumptions, and using fundamental principles of behavioral economics. In particular, this approach allowed non-efficient markets, illiquidity, irrational agents, and general price distributions among many others. [1] These advances were frowned upon by academia for a long period due to loss of properties like finite volatility, but have found its way back to the mainstream lately.

The Fractional Black Scholes model is based on fractional Brownian motion which was originally introduced by Kolmogorov in 1940. Fractional Brownian motion is also known as a stochastic two sided Brownian motion process or generalized Gaussian processes defined mathematical as

$B_t^H,\ t \in R^+$ and $H \in [0, 1]$ is the Hurst exponent.

Consider the fractal differential equation

$$dS_t = \mu S_t dt + \sigma S_t dB_t^H \qquad \rightarrow (1)$$

where, under the risk neutral measure

$$dS_t = rS_t dt + \sigma S_t dB_t^H \qquad \rightarrow (2)$$

Equation 1 can be solved using the fractional Ito calculus to obtain our stock price propagation based on the fractional Brownian motion as below:

$$S_t = S_0 exp\left(\sigma B_t^H + \mu t - \tfrac{1}{2}\sigma^2 t^{2H}\right) \qquad \rightarrow (3)$$

where S_0 is the current/initial stock price and r is the risk free interest rate

Hu and Oksendal [11,12]were able to improve the classical Black Scholes model by using Fractal Ito calculus and a fractal Geometric Brownian motion. We are able to derive the following closed form solution for a European call option.

$$\pi_c = S_t\, \phi(d_1) - Ke^{-r(T-t)}\phi(d_2) \qquad \rightarrow (4)$$

$$d_1 = \frac{\ln ln\left(\frac{S_t}{k}\right) + r(T-t) + \frac{\sigma^2}{2}(T^{2H}-t^{2H})}{\sigma\sqrt{(T^{2H}-t^{2H})}}$$

$$d_2 = \frac{\ln ln\left(\frac{S_t}{k}\right) + r(T-t) - \frac{\sigma^2}{2}(T^{2H}-t^{2H})}{\sigma\sqrt{(T^{2H}-t^{2H})}} \qquad \rightarrow (5)$$

Where t is the start period, T is the end period, S_t is the initial price at the start of the period, K is the strike price, ϕ is the cumulative standard normal distribution, r is the risk free rate, and H is the Hurst exponent.[11]

Put options can be priced using the put-call parity.

$$\pi_c - \pi_p = Ke^{-r(T-t)} \qquad \rightarrow (6)$$

Therefore based on equations 1 to 5 if we are able to determine the Hurst exponent and volatility as accurately as possible then we can price our options based on the fractal Black Scholes model.

Traditionally, the idea of implied volatility is borrowed from Black Scholes model to compute volatility and Hurst exponent is estimated separately. We will therefore look at various methods for estimating the Hurst exponent and see how to improve them so that we can have more accurate pricing of European options. We will look at the rescale range (R/S) analysis method and the minimum covered area method of estimation of fractal index.

1.3.2 Rescale Range (R/S) Analysis Method

This approach for estimating Hurst exponent was developed by Mandelbrot and is a very widely used method to determine Hurst exponent.

The process for estimating the Hurst exponent using R/S analysis is as below.

1. Obtain the logarithm of the ratio of the stock price from the i^{th} period to the $(i-1)^{th}$ period where *for i= 1,2,3,………..,n*

$$X_t = log\left(\frac{X_i}{X_{i-1}}\right) \qquad \rightarrow (7)$$

This helps to remove trends from our data.

2. Compute the standard deviation on the sample size n of our data $s(n)$

3. We compute the quantity

$$\left(\frac{R}{S}\right)n = \frac{1}{S(n)}\left[sup\left(X_t - X_t^*\right) - inf\left(X_t - X_t^*\right)\right] \quad \rightarrow(8)$$

Where $X_t^* = \frac{1}{n}\sum_{i=1}^{n} X_i$ is the sample mean. →(9)

4.We repeat 2 and 3 for various values of n and then plot a graph of $log((\frac{R}{S})_n)$ against $log(n)$

5.Determine the slope of the line of best fit and that is the Hurst exponent for our data which can then be used in the fractal black Scholes model to price our options.

Please note that the Hurst exponent can also be more easily obtained by using the Ordinary least squares (OLS) method. [1]

1.3.3 Box Counting Method For Estimating Fractal Dimension

The approach in this section was first suggested in 1919 by Hausdorff it basically involves plotting the time series data on a graph and the surface of the time series data on the graph is divided into small cells(boxes) of size h the number of boxes $N(h)$ is then counted[8][13]..

Note that; $h = t_i - t_{i-1} = \frac{b-a}{m}$ →(10)

where $i = 1, 2, 3, ..., m$ and $N(h) \sim \left(\frac{1}{h}\right)^D$ →(11)

The computation of $N(h)$ is repeated for various values of h and a graph of $log(N(h))$ against $log(h)$ is plotted with the slope being the fractal dimension D. The hurst exponent H is then determined from $H = 2 - D$.

It is to be noted that OLS estimation can be applied to obtain the fractal dimension here as well.[13].

1.3.4 Calibration: Machine Learning and Artificial Neural Network Approaches To Option Pricing

Model calibration is a complementary problem of option pricing, since any pricer involves parameters based on theoretical models, which needs to be decided prior. There has been a lot of advances in financial model calibration techniques. However, many of them were, understandably, to improve Black Scholes model. The initial techniques included the calibration of the implied volatility complying with the assumption of constant volatility. But these were inadequate primarily due to the non-constant relationship of the volatility with the option term.[2] Many subsequent approaches tried to account for this mismatch by improving the model. The notable ones include smile-consistent pricing, stochastic volatility models, local volatility models, jump diffusion models etc.

However, relatively fewer advances have been made for models set in the Mandelbrotian world. The noteworthy ones are Mare et. al[5] and Li and Chen[6] These have focused on exploiting the theoretical and heuristic advantages that fractal market hypothesis has over the efficient market counterpart. The results are arguably superior to the vanilla models, since the volatility term structure is a power function rather than a constant. [5] However, for calibration they employ techniques which might be too simplistic, like linear regression or which need global optimization and run the risk of getting stuck in local minima.

A different approach which is based on machine learning has recently gained prominence, especially with increase in availability of powerful computational resources. In particular, there have been works on using Artificial Neural Networks for Black Scholes option pricing and calibration. Notable among those are Horwaith et. al [3] and Itkin [4] . These have laid the theoretical foundations for proving important conditions of the neural network based pricers like no-arbitrage, stability etc, as well as compiling the tractable training approaches. But, the usage of machine learning techniques in fractal space for calibration, is little to none. This is a potentially rewarding approach.

The most attractive quality of such an approach is that most of the intensive computations are a one time activity. Adapting a Neural network based pricing model, and an inverse map, in the context of fractional derivative pricing [4] could make daily calibration of the model parameters, quicker and more efficient. While it needs to be explored and adapted to the fractal setting, the generalised version involves training an appropriately deep neural network based pricer with a sophisticated cost function having soft penalties to ensure no-arbitrage conditions are not (approximately) violated. This is done prior and stored. Later, an inverse map can be made use to obtain the calibrated parameters at once. In addition, invertible neural networks have had success in different domains including medicine and astronomy[7] , but haven't been used widely in the financial domain. It could also be successfully used for effective calibrations. However, more design analysis is required to confirm its plausibility.

In conclusion, academia has started giving importance to Black Scholes model's alternatives after realizing its numerous shortcomings. And the fractal approach is one of the most theoretically backed candidates. Tractable calibration techniques are imperative for any model's success. So, it would be fruitful to understand the existing estimation techniques in depth, and try to apply more advanced approaches such as Neural Networks to make the blooming field of Mandelbrotian derivative pricing practical and accurate.

1.4 Competitor Analysis

Option trading and modelling is an area which sees numerous continuous and fast improvements. While it is extremely difficult to compile all the important work done in this field, we have attempted to survey the existing work, in the light of the emergence of fractal market hypothesis and models based on it. The prime proposal is to examine the existing techniques and explore the use of advanced estimation techniques in the fractional Black Scholes model. The market viability of improving the calibration of this fractal model can be better grasped by making use of a SWOT analysis of the existing work. Here, an attempt is to elucidate the advantages and gains of the current work and approaches, along with what could be improved and what opportunities are present.

SWOT: Existing Research

Strengths	Weaknesses
• The option pricing based on Black-Scholes models are the current industry standard. • There have been various improvements to improve the shortcomings of the BS model. • Numerous analytical and numerical implementation strategies are available. • Calibration techniques are really mature and can be chosen according to specific needs. • Multiple theoretical advances, including strong proofs were made in the Fractal Market domain.	• Black Scholes model have been demonstrated to multiple theoretical limitations. • It has failed to predict Black Swan events. • Improvements of the model often come at a cost of less intuitiveness & computational complexity • While Fractional models are starting to be used, calibration techniques are still at an early stage.
Opportunities	**Threats**
• Fractional model based option pricing offers a heuristic approach, while being free of numerous erroneous assumptions. • Works have demonstrated superior performance to BS model and comparable performance to SABR model.[1] • If calibration techniques are improved for Mandelbrotian models, it could provide a practical yet more accurate forecasts. • Being based on heuristic techniques, it offers better feedback.	• Using machine learning models is more intense computationally. • Numerical techniques often require stability and bias analysis. • While more complex, SABR model offers slightly better fit and is more widely adopted.

From the above analysis, few conclusions can be drawn. Firstly, the Black Scholes model has matured and has been a foundation for most of the option trading activities. But, many of its assumptions have been questioned and invalidated, especially after the black swan event of the 2008 global crisis. Fractal models, its viability and heuristic strengths have been discussed at some length. But, apart from basic estimation techniques for Hurst exponent and using the prevalent ideas of implied volatility, calibration of the Fractal models aren't given much attention, as clearly pointed out in recent works[9]. While there are clear benefits to doing this, it is to be noted that numerical or approximate techniques like Neural networks shouldn't undermine the heuristic and intuitive strength of the model by introducing instability or bias.

2. Theoretical Framework

2.1 Inverse Map/ Invertible Neural Networks Approach

This approach is inspired by Deep Learning Inverse Map Approaches employed by Horvath. et. al (2019)[3] and Itkin (2019) [4] in the context of Vanilla Black Scholes model.

Neural network based pricer plays an auxiliary role in calibration, but is often an effective primary step. So, that is discussed first.

2.1.1 Neural Network based Pricer

Analytical approaches are often time consuming and significantly slow down the calibration and estimation steps. For numerical advantages and tractability, many prefer approximation pricing models over the analytical models. The primary advantage of one of the more important approximation techniques, neural network based pricers is that the numerical approximation training of the pricing model becomes a one time preprocessing step which can be done offline. However, this is often the most computationally demanding step. Approximating the pricing functional using neural networks has numerous challenges like choosing a suitably rich model to learn its characteristics, curating big enough datasets, designing proper loss function to ensure essential market properties, and employing proper optimizers to avoid getting stuck at local minimas.

In the context of Fractal models, following in the line of Itkin at. al[4] , a straightforward approach could be to learn the characteristics by using random vectors generated by sampling the model. It should also be noted that preprocessing of such generated vectors are necessary for a quality dataset. One such step would be to avoid unnaturally large or small prices. While the network architecture is yet to be ascertained during training and finetuning, the number of layers should be at least 2 or more, to learn the analytical formula functional surface.

The 2 most critical pieces are that of cost function and training optimizer. All popular deep learning frameworks including Tensorflow and Pytorch provide efficient training optimizers. Cost function, on the other hand, is much more closer to the domain and hence, needs to be specifically designed. The primary component of the cost function is the mean squared error of the output price, but equally important is to ensure no arbitrage conditions are satisfied. The conditions for call prices as explained in Carr and Madan[8], are as follows:

$$\frac{\partial C}{\partial T} > 0, \quad \frac{\partial C}{\partial K} < 0, \quad \frac{\partial^2 C}{\partial K^2} > 0 \qquad \rightarrow (12)$$

But, as shown in Marquez-Neila et. al[7], in the field of Deep Learning, imposing hard constraints often adversely impacts the training effectiveness. Soft regularising constraints are more effective, when combined with modern backpropagation based optimizers. In short, the regularised loss function would be roughly of the form :

$$L_c = \arg\min_w \sum_i \sum_j \left\{ \left[C\left(\boldsymbol{\xi}_i\right) - C_{\text{ANN}}\left(\boldsymbol{\xi}_i, w_{i,j}\right)\right]^2 + \Phi_{\lambda_1,m_1}\left(-\frac{\partial^2 C_{\text{ANN}}\left(\boldsymbol{\xi}_i, w_{i,j}\right)}{\partial K^2}\right) \right.$$
$$\left. + \Phi_{\lambda_2,m_2}\left(-\frac{\partial C_{\text{ANN}}\left(\boldsymbol{\xi}_i, w_{i,j}\right)}{\partial T}\right) + \Phi_{\lambda_3,m_3}\left(\frac{\partial C_{\text{ANN}}\left(\boldsymbol{\xi}_i, w_{i,j}\right)}{\partial K}\right) \right\} \qquad \rightarrow (13)$$

where $\Phi_{\lambda,m}(x)$ is a penalty function. A widely used penalty is :

$$\Phi_{\lambda,m}(x) = \begin{cases} 0, & x < 0 \\ \lambda x^m, & x \geq 0 \end{cases} \quad \rightarrow (14)$$

2.1.2 Calibration techniques approaches

Calibration is an inverse problem of the above. Given the observable quantities and market prices, we wish to calculate the implicit parameters of the model. More often than not, calibration by inverting the analytical formula is infeasible. So, a data driven numerical approach is suited to this problem. The practitioners use model calibration with the recent data as often as possible to get best heuristics and parameter estimations. For this, price computation needs to be extremely efficient, which we can achieve using Neural networks based pricing, discussed above. The next and more important component is to find techniques for solving this optimization problem, expressed mathematically as:

$$\underset{p}{\arg\min} \sum_{i=1}^{n} \sum_{j=1}^{m_i} \omega_{i,j} \left\| C_M\left(\boldsymbol{\theta}_{i,j}\right) - C\left(\boldsymbol{\theta}_{i,j}, \boldsymbol{p}\right) \right\| \quad \rightarrow (15)$$

where θ refers to the observable quantities, p to the parameters, C_M to the market price and C to the model under consideration. That is, we aim to get single values for the model parameters, H and σ.

Note that this optimization can be solved analytically under L^2 norm, with the parameters obtained using below formula:

$$p_k = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m_i} \bar{\omega}_{i,j,k} p_{k,\text{model}}\left(\boldsymbol{\xi}_{i,j}\right)}{\sum_{i=1}^{n} \sum_{j=1}^{m_i} \bar{\omega}_{i,j,k}} \quad \rightarrow (16)$$

The only portion left is to compute $p_{k,model}$. For obtaining $p_{k,model}$, we plan to use 2 related techniques. The first one is training of an inverse map which is a neural network on the lines of the pricing model analysed in the previous section. It will be trained with an inverse dataset, formed by inputs and outputs of the pricer. The simplifying feature is that since the price incorporates the market requirements, this model can be trained in a relatively straightforward manner. A more challenging, but intuitive approach would be to use a different architecture called Invertible Neural Networks, introduced by Ardizzone et. al (2018)[9] for pricer. Although the model complexity increases, it gains invertibility property. So, the model can be inverted to obtain the model parameters without the need to train another network for the inverse.

3. Methodology

Python programming language shall be used to do all components of implementation, including processing the data, implementing and training networks and to carry out algorithmic processes on our processed data.

3.1 Data Collection and Handling

Stock price data will be obtained from Yahoo finance and option price data shall be obtained from the Chicago Board options exchange. Any data that can not be obtained we will apply Monte Carlo methods on simulated data. This data will be the primary source for testing the effectiveness of the calibration procedures.

In addition, we hope to create a module for generating random vectors of observables, parameters and call prices sampling the fractal model, for effective offline training of Neural network prices.

3.1 Data Analysis and Modeling

3.1.1 Modeling Hurst exponent using R/S analysis

The techniques explained in the literature review on modeling Hurst exponent using R/S method shall be used. We shall use the Ordinary least Squares method to determine the hurst exponent for each of the sections of our time series price data.

In the literature the hurst exponent is obtained by getting an average of all the values of the hurst exponent for the various sections of the data. In our approach we will try a normal average, a moving average approach and exponential moving average approach to estimate the hurst exponent of our data.

Moving averages help us filter out noise in our time series data, exponential moving averages give more weight to recent data and therefore will provide us with a hurst exponent that is more representative of the current trend.

Implied volatility shall be computed from the hurst exponent and the option prices shall be computed using the closed form solution.

3.1.2 Calibration using Neural Networks

Generated random stock parameters shall be used for offline calibration of our neural networks. The methods to be used are as per explanation in the theoretical framework. For implementation, the primary options are PyTorch/Tensorflow for vanilla NN designs and FrEIA(Framework for Easily Invertible Architectures) for designing Invertible Neural Networks.

The methodology to be used in calibration of the option pricer is based on the classical Black Scholes model and we will therefore try to use it in pricing options under the fractal Black Scholes model.

3.1.3 Comparison of Classical solution and Neural network pricer

The option prices obtained from the two methods shall be compared with actual market data option prices to determine the effectiveness of each of the models and to notice any areas of improvement that can be made on each of the models.

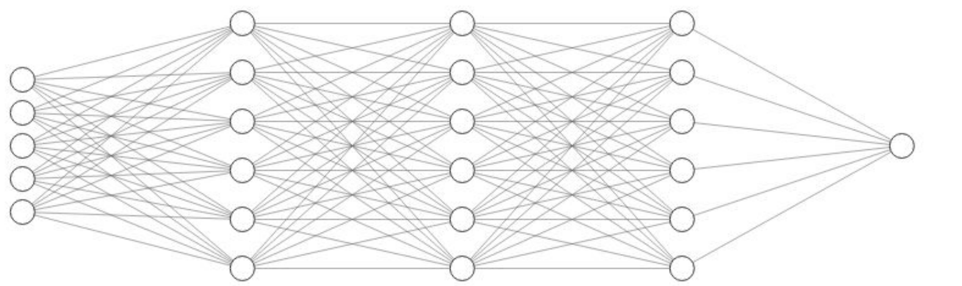
4. Results

4.1 Neural Networks

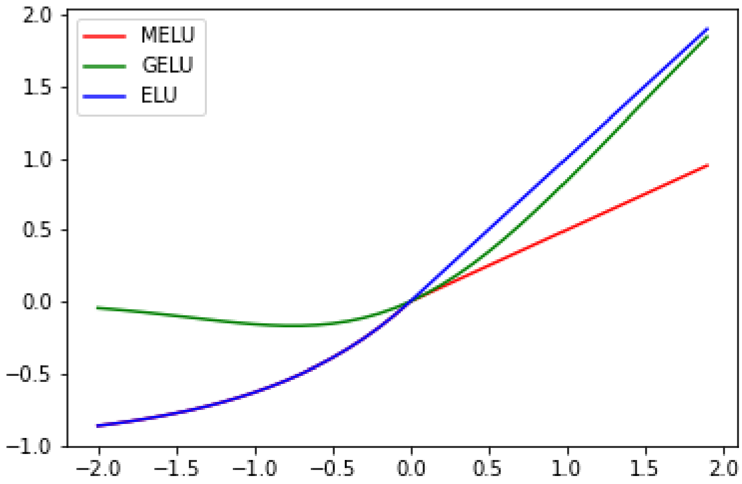
Generated random stock parameters shall be used for offline calibration of our neural networks. The methods to be used are as per explanation in the theoretical section. All the implementation was done in Python, using the PyTorch deep learning framework.

4.1.1 Option Pricer

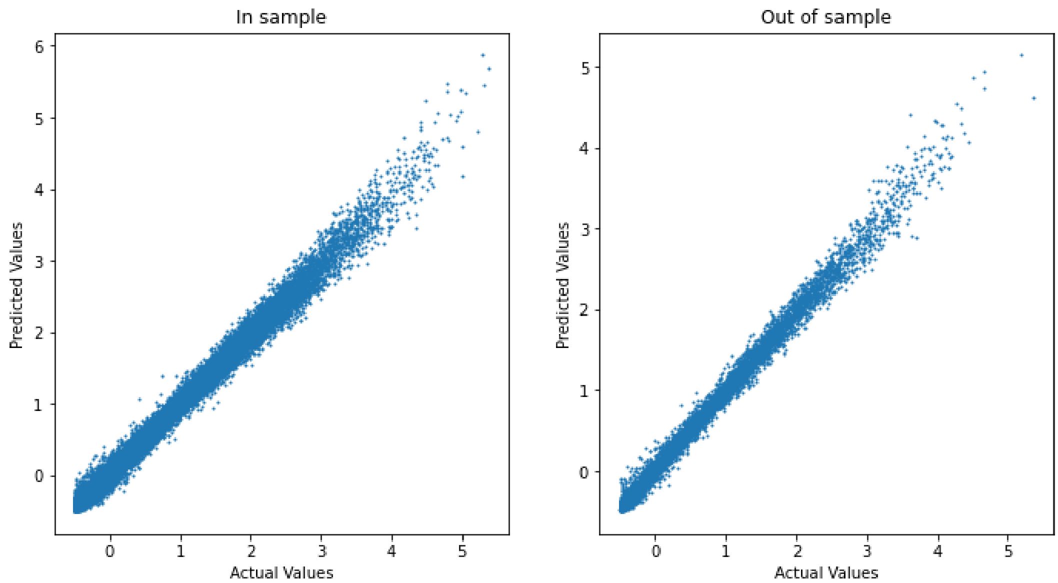
While designing the neural network, the items of consideration were the architecture, the loss function and training optimiser.
A depth of 3 was chosen to utilise the approximation power of deep Neural Networks. The architecture is as follows:



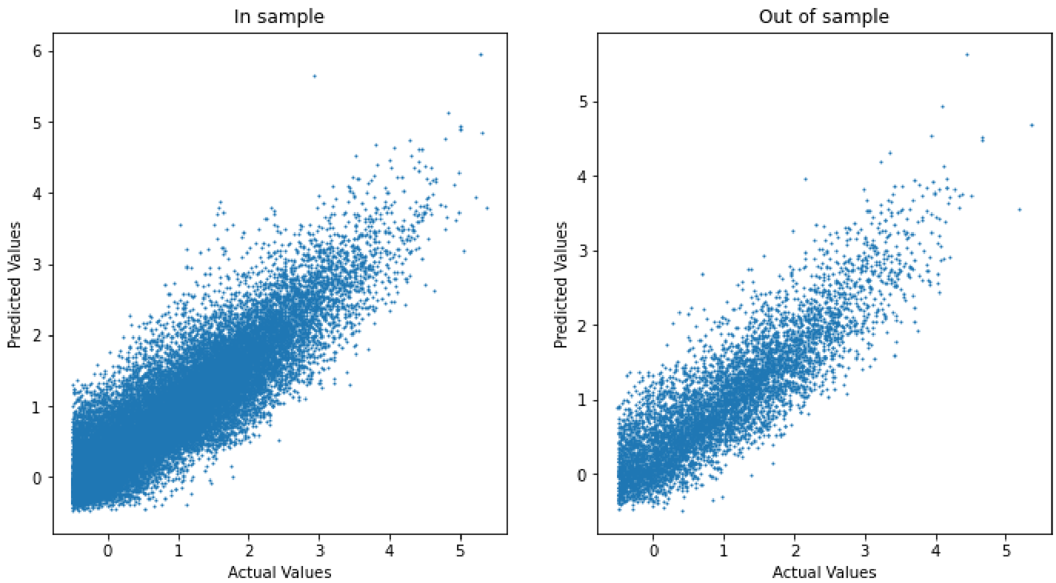
The activation function which is essentially for the neural networks to approximate a nonlinear function needs to be carefully chosen since the additional no-arbitrage conditions which can be imposed requires the functional to be C^2 i.e it should be twice continuously differentiable. So, activation functions like RELU (Rectilinear Linear Unit) and ELU (Exponential Linear Unit) , while proven to be effective in vanilla networks, don't qualify. Ikin suggests using a modified ELU (MELU) which was tested, but the training was unstable. A relatively new C^2 activation function called Gaussian Error Linear Units (GELU) was used in conjunction with the softplus function which resulted in stabilised training.



The loss function would be Mean Squared Loss. Initial experiments could be simplified as follows: Since the call pricer is linear homogenous in the stock price S and strike price K, K could be eliminated by scaling the dataset: $S \to S/K$, $C \to C/K - \min(C/K) - 0.5$ along the lines of Itkin's approach. Essentially, this ANN is a function f: $(S,T,r,H,\sigma) \mapsto C_f$. The initial experiments used just this vanilla loss function with no penalty and had a good fit as shown below.



But, as Carr and Madan (2005) proves, any call pricing model should satisfy the three conditions mentioned in Equation (12). In almost all deep learning models, additional constraints like these will be better achieved by using soft constraints like penalisation as opposed to hard constraints. It is demonstrated in Marquez-Neila et al (2017). But, it is to be noted that this does not eliminate arbitrage, rather made negligible. However to enforce this constraints involving derivatives and double derivatives w.r.to K, we need to increase the input dimension back to 6. This results in a noticeably inferior fit after an initial round of training as shown below. *This needs to be improved further.* However, as expected, the penalty component of L_c in equation (13) is very less.



4.1.2 Inverse Maps

An important problem, which hasn't been actively considered in the fractional Brown Scholes model based pricing, is the calibration of its parameters Hurst parameter (H) and fractional volatility (σ). Often, the computations are a time consuming step in this. But the neural network approaches can significantly reduce the time involved due to quick inference.

Using Neural networks to directly return the calibrated parameters for Black Scholes model has been cited by few authors as difficult to train. (Hernandez, Tomas) In our initial experiments too, the results were not satisfactory.

Using the pricing maps outputs as opposed to the actual prices has been shown to be more robust (Horvath), wherein the first steps involves approximating the neural network pricer and secondly using its outputs to solve the optimization problem of finding the calibrated parameters. This approach results in inaccurate but promising fits. It is inaccurate possibly due to the below reasons which needs to be rectified.

1. The pricer needs to be improved further.
2. Predicting 2 quantities (H,σ) without correlation is affecting the performance.

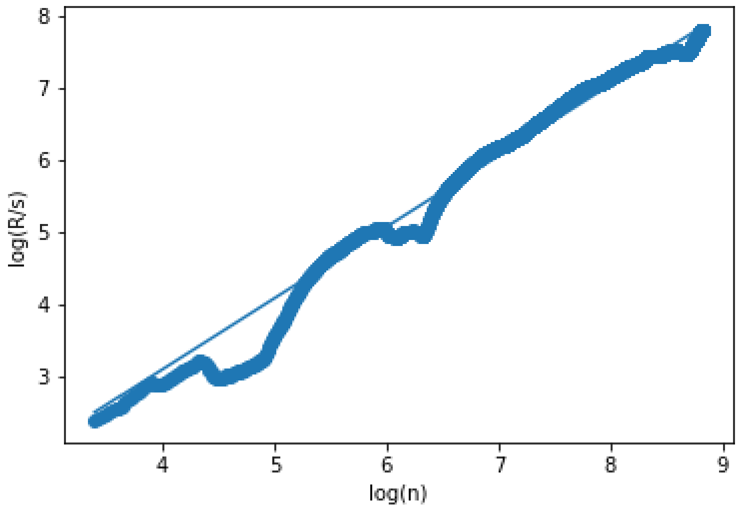
An enhancement under progress is to create the network to predict a single quantity - $\sigma\tau^{H}$. The rationale being the 2 model parameters are utilised only in this parameterised manner. This should result in better performance. Another approach, albeit a bit ambitious, which we plan if time permits, is to utilise a different type of Neural Network called Invertible Neural Networks, which has the intrinsic property of calculating the parameter.[9] And they could be developed using FrEIA framework built on PyTorch. But, they are less understood, difficult to train and have less framework support especially for penalised loss functions.

4.2 Hurst Exponent techniques using Rescale analysis methods

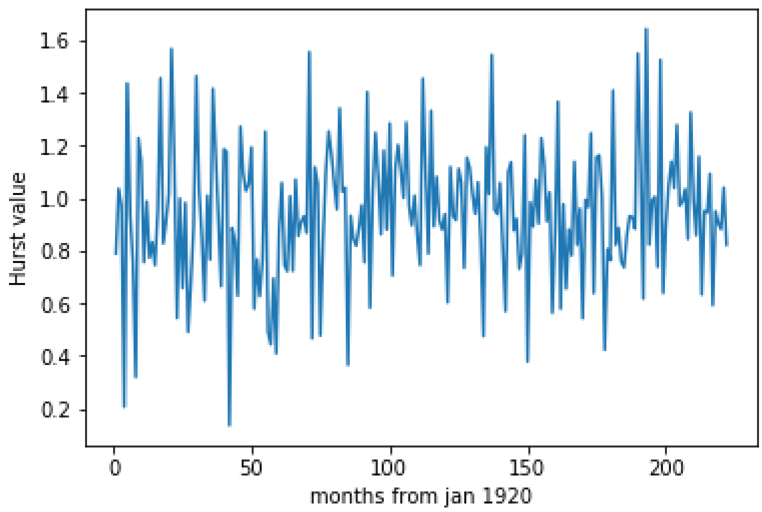
In this section, we attempt a preliminary comparison of 4 different models for determine hurst exponent and the implied/fractional volatility, these methods are:

1) Using Rescaled Analysis for all the data period from 1993 to 2019

We computed computed $log((\frac{R}{S})_n)$ against $log(n)$ and used train_test split to create both training data and test data. The training data is fit into a linear regression model in python Scikit-learn and a scatter plot and line of best fit is obtained as below:



2) Compute Hurst exponent for each of the months from 1993 to 2019. We computed hurst exponent values for each month using rescaled analysis, the data is plotted as below.



One approach was computing the average for all the months and using that to price the options.

Looking at the plot the graph exhibits characteristics of stationarity as it appears to have a constant value through the months with periods of volatility, we therefore used an autoregressive model to evolution of the hurst exponent over the months model.

Below is the summary of the model:

Out[222]:

ARMA Model Results

Dep. Variable:	y	No. Observations:	222
Model:	ARMA(2, 0)	Log Likelihood	-12.645
Method:	css-mle	S.D. of innovations	0.256
Date:	Tue, 28 Jul 2020	AIC	33.290
Time:	22:20:46	BIC	46.901
Sample:	0	HQIC	38.785

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.9428	0.015	64.165	0.000	0.914	0.972
ar.L1.y	-0.1404	0.067	-2.095	0.037	-0.272	-0.009
ar.L2.y	-0.0305	0.067	-0.456	0.649	-0.162	0.101

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	-2.2995	-5.2413j	5.7236	-0.3158
AR.2	-2.2995	+5.2413j	5.7236	0.3158

The model is well fitting as the explanatory variables are greater than twice the standard error.

Below is a summary of the performance of each of the models:

Model	Hurst exponent	Volatility	MSE of prices
Linear model for R/S	0.987848851	0.011453107	82660.05756
Monthlyhurst computation-avg	0.942634636	0.011453107	82668.39844
Monthly hurst with AR model	0.956535222	0.011453107	82639.62633
Standard BS model	N/A	0.011453107	82804.44884

5. Conclusion

From the table we do notice that the Models do reduce the error between the actual price and the calculated price compared to the standard BS model. And there has been significant progress in the implementing and training of option pricing neural network models, as well as the implementation of pricing maps. The offline training steps have been documented in the codebase and can be easily reused for variations of other fractal based models. The inferences as well as the analytical computation of calibration are extremely quick, which is an important advantage in finance.

However few important steps still remain. The performance of the inverse mapping needs to be greatly improved to get accurate calibration. The market data for testing the effectiveness of the calibration needs to be standardised for better comparison with other standard techniques. As explained in the results section, we are actively working on both of these items.

6. References

Various Journals, Books and Softwares used for the fulfillment of this project.

6.1. Journal Articles

[1] Oleksii Romanko (2018). Option pricing with fractals: An empirical analysis. Economic Analysis MA Thesis. Kyiv School of Economics

[2] Dupire, B (2006). Model free results on volatility derivatives. Working paper. Bloomberg, NY: SAMSI.

[3] Horvath, Muguruza, Tomas (2019). Deep Learning Volatility A deep neural network perspective on pricing and calibration in (rough) volatility models. arXiv:1901.09647

[4] A. Itkin (2019). Deep learning calibration of option pricing models: some pitfalls and solutions. arXiv:1906.03507.

[5] Flint, Emlyn & Mare, Eben (2017). Fractional Black–Scholes option pricing, volatility calibration and implied Hurst exponents in South African context. South African Journal of Economic and management Sciences. 20. 10.4102/sajems.v20i1.1532.

[6] Li, Kinrey & Chen (2014). Implied Hurst Exponent and Fractional Implied Volatility: A Variance Term Structure Model. Available at SSRN: https://ssrn.com/abstract=2383618

[7] Marquez-Neila, P., Salzmann, M., Fua, P., 2017. Imposing hard constraints on deep networks: Promises and limitations. URL: https://arxiv.org/pdf/1706.02025.pdf

[8] Carr, P., Madan, D., 2005. A note on sufficient conditions for no arbitrage. Finance Research Letters 2

[9] L. Ardizzone, J. Kruse, S. Wirkert, D. Rahner, E. W. Pellegrini, R. S. Klessen, L. Maier-Hein, C. Rother and U. Kothe (2018), Analyzing inverse problems with invertible neural networks. arXiv:1808.04730

[10] Angkola, F. (2016). A generalized fractal dynamics option pricing model with transaction costs. PhD thesis. Curtin University

[11] Teneng, Dean. (2011). Limitations of the Black-Scholes model. International Research Journal of Finance and Economics. 99-102.

[12] Hu, Y. and B. Oksendal, (2000), Fractional white noise calculus and application to Finance, Preprint, University of Oslo

[13] Hu, Y., B. Oksendal, and A. Sulem, (2000), Optimal consumption and portfolio in a Black-Scholes market driven by fractional Brownian motion, Preprint 23/2000, University of Oslo

[14] Lianga, Z., Fengb, Z., and Guangxianga, X., (2012),Comparison of Fractal Dimension Calculation Methods for Channel Bed Profiles, International Conference on Modern Hydraulic Engineering

6.2. Softwares

[15] Codecogs plugin for rendering LaTeX equations. Found here: https://www.codecogs.com

6.3. Books

[16] Marcos Lopez de Prado, Author, “Advances in Financial Machine Learning”, Wiley Publishing. (2018).