Deep learning the Hurst parameter of fractional processes; its reliability and effect on option pricing

László Márkus, Dániel Boros
partially joint work with
Bálint Csanád, Iván Ivkovic and András Lukács

Department of Probability Theory and Statistics, Eötvös Loránd University

November 23, 2023

Motivation

- Probabilists and statisticians created **sophisticated financial asset price models** matching many features of market prices. (Heston's stochastic volatility, rough Bergomi, SABR, variance gamma, normal inverse Gaussian, Merton, Kou, Bates, to name a few)
- Large investment banks do not often use those in their daily operation
- The computational requirements of parameter estimation or model calibration is the critical problem of those models
- To make the sophisticated models usable **fast and accurate** estimations are needed, and **AI** is an obvious candidate for providing precisely that
- Here, however, the little known **reliability of the AI** estimation is the problem

Motivation

- We focus on the class of linear fractional processes and, within that class, on the **fractional Brownian motion (fBm)** the **fractional Ornstein-Uhlenbeck process (fOU)** (of the first kind) and the **fractional Lévy stable motion (fLsm)**
- We estimate the **Hurst parameter** of those processes by deep learning techniques
- In the financial literature, the fBm is reported as a successful model for S&P500, while the fOU process is suggested for modeling, e.g., instantaneous interest rates, specific index prices, and foreign exchange rates.
- The fOU serves as the basis for modeling rough stochastic correlation between asset prices or between the volatility and asset price, e.g., in the Heston model
- **Current** deep-learning-based research predominantly concentrates on **predicting** risks or losses.
- To enhance financial stability, an ambient risk prevention framework is essential. **Deep learning** may serve to that end to unravel the dynamic **parameters** of stochastic processes that underlie or stimulate **risk propagation**.

Fractional Brownian motion

Definition (fBm)

The process $(B_H(t))_{t\geq 0}$ is called the ***fractional Brownian motion*** *(fBM), which is a zero-mean Gaussian process having the following property:*

$$E(B_H(t+\Delta)-B_H(t))^2=\Delta^{2H}$$

where $0<H<1$ is the ***Hurst*** *parameter or Hurst exponent. The autocovariance function can be expressed as:*

$$cov(B_H(t),B_H(s))=\frac{1}{2}\left(|t|^{2H}+|s|^{2H}-|t-s|^{2H}\right).$$

- *The fBm has* ***stationary*** *and* ***dependent*** *increments.*
- *It is* ***self-similar****: upon rescaling time by $a>0$, $B_H(at)\sim a^H B_H(t)$.*
- *It has* ***fractal*** *paths with fractal dimension: $D=2-H$.*

Fractional Ornstein-Uhlenbeck Process

Definition (Fractional Ornstein-Uhlenbeck Process)

A mean-reverting ***fractional Ornstein-Uhlenbeck (fOU)*** *process is given by a fractional stochastic Langevin differential equation:*

$$dX(t) = -\alpha(\mu - X(t))dt + \sigma dB^H(t)$$

with the initial condition

$$X(0) = X_0$$

where $\alpha, \sigma, \mu \in \mathbb{R}$ *are the process parameters,* $\alpha, \sigma > 0$ *and* X_0 *is a random variable in general.*
$X(t)$ *has a* ***stationary*** *regime when suitably initialized.*
The solution of the fOU equation exists, unique – up to initial condition –, and can be given by an explicit formula. For 0 initialization and expectation it reads as

$$X(t) = -\sigma \int_0^t e^{-\kappa(t-s)} dB_H(s).$$

Generating the Training Set

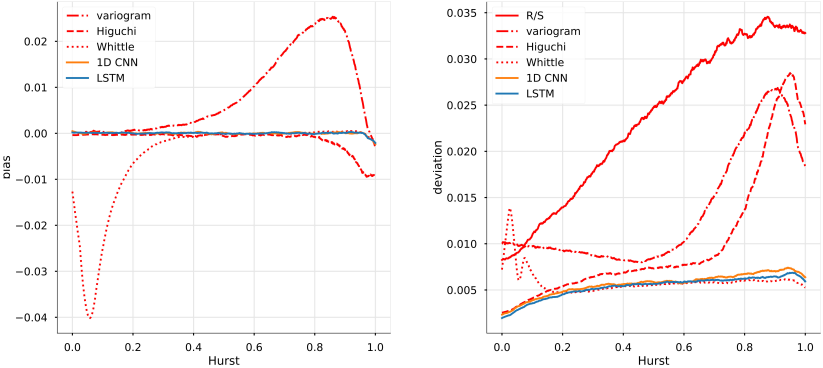
- To train the network properly, we need a **large and quality-controlled simulated sample** from fBm and fOU processes.
- We created an efficient **Python** implementation of Kroese's method[1] belonging to the Davies-Harte procedure family leveraging the FFT with a computational complexity of $O(n \log n)$.
- **Speed** is of paramount importance here as we use ten million trajectories of length 2-6000 for training the network. That's **twenty to sixty billion data** for each training configuration.

[1] Dirk P Kroese and Zdravko I Botev. Spatial process generation. arXiv preprint arXiv:1308.0399,2013

Neural Network Architecture and Training

- **Model Design:** A **shift and scale invariant LSTM model** is employed. The input data is based on the increments of the original series. The LSTM has an input size of 1, with 128 hidden units, and comprises 2 layers. The output layer is a multi-layer perceptron with the PReLU activation function.
- **Data Configuration:** Utilizes a Fractional Brownian motion (fBm) time series. The Hurst parameter is uniformly generated between 0 and 1. Each epoch consists of **100,000 trajectories of length 1600, 3200, 6400**.
- **Training Strategy:** The model uses a learning rate of 0.0001 with the AdamW optimizer. The **MSE and L1Loss (MAE)** are the chosen loss functions. Training spans 25, 50 and 100 epochs, with batch sizes of 32 for training and 128 for validation.

Metrics (Bias and Variance)



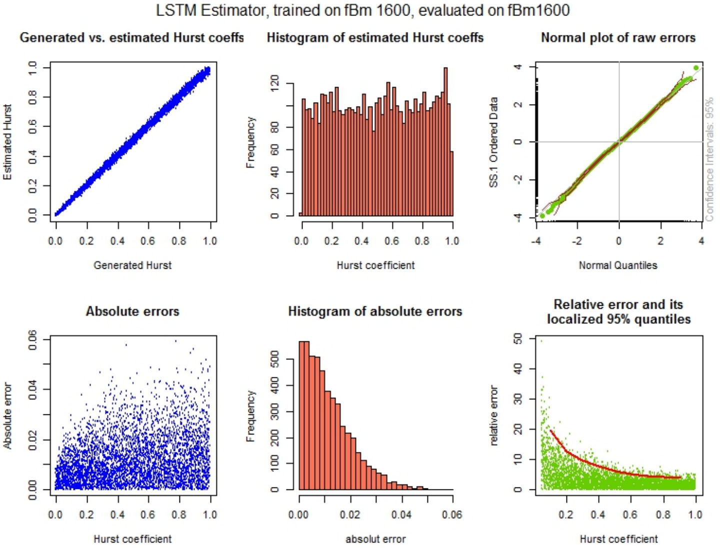
AI vs Higuchi

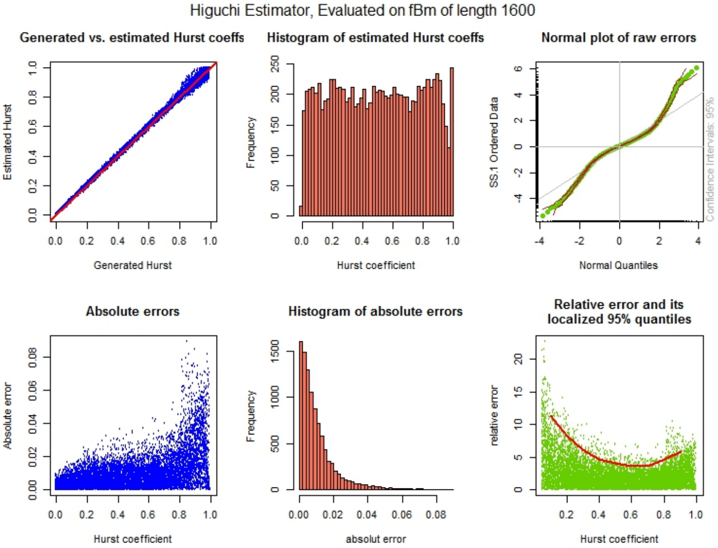
Table: Estimation of fBms' Hurst parameter

	MSQErr	Abs Err (max)	Abs Err (95%)	Rel Err% (95%)
AI 400	0.00097	0.14820	0.06304	24.44510
AI 1600	0.00022	0.05915	0.02999	11.72535
AI 6400	0.00007	0.03893	0.01651	9.80169
Higu400	0.00198	0.25071	0.09025	33.69157
Higu1600	0.00061	0.12567	0.05150	14.34120
Higu6400	0.00023	0.08953	0.03327	7.50449

Table: Estimation of fOUs' Hurst parameter

	MSQErr	Abs Err (max)	Abs Err (95%)	Rel Err% (95%)
AI 400	0.00094	0.12787	0.06196	22.59178
AI 1600	0.00024	0.07103	0.03156	11.94912
AI 6400	0.00006	0.02935	0.01566	6.53867
Higu400	0.00191	0.21441	0.08977	32.68264
Higu1600	0.00058	0.12947	0.04974	14.94272
Higu6400	0.00020	0.09212	0.03039	7.39990


LSTM Estimator, trained on fBm 1600, evaluated on fBm1600
Generated vs. estimated Hurst coeffs
Estimated Hurst
Generated Hurst
Histogram of estimated Hurst coeffs
Frequency
Hurst coefficient
Normal plot of raw errors
SS.1 Ordered Data
Normal Quantiles
Confidence Intervals: 95%
Absolute errors
Absolute error
Hurst coefficient
Histogram of absolute errors
Frequency
absolut error
Relative error and its localized 95% quantiles
relative error
Hurst coefficient

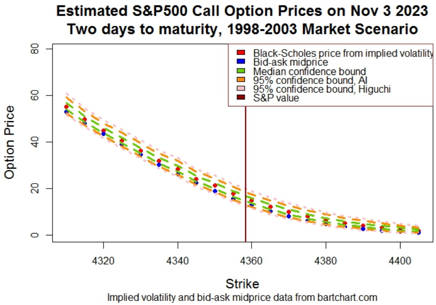

Higuchi Estimator, Evaluated on fBm of length 1600
Generated vs. estimated Hurst coeffs
Estimated Hurst
Generated Hurst
Histogram of estimated Hurst coeffs
Frequency
Hurst coefficient
Normal plot of raw errors
SS.1 Ordered Data
Normal Quantiles
Confidence Intervals: 95%
Absolute errors
Absolute error
Hurst coefficient
Histogram of absolute errors
Frequency
absolut error
Relative error and its localized 95% quantiles
relative error
Hurst coefficient

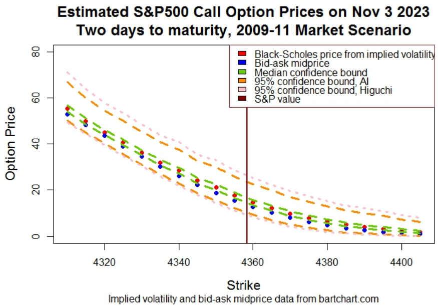
An Illustrative "Toy" Example in Option Pricing

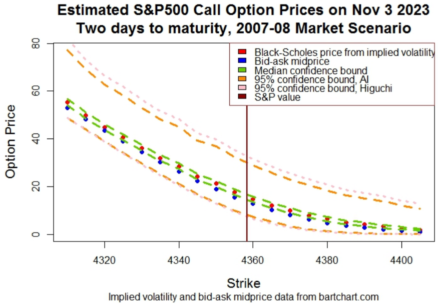
We wish to illustrate what may the inaccuracy of the Hurst parameter's estimation result in in the setting of a call price.
Some authors advocate for a fBm modeling of the SP&500 – accept it for the time being.
E.g., D.C. René, and Rivera-Solis claim that the Hurst parameter is time dependent and they estimate it as
$H = 0.61\ in\ 1998-02, \qquad H = 0.14\ in\ 2007-08 \qquad H = 0.28\ in\ 2009-11$

- Denote the implied volatility by Σ for a call with strike K and maturity T
- Denote the variance parameter of the fBm by σ, i.e. $D^2 B_H(t) = (\sigma \cdot t)^{2H}$.
- Given the SP&500 value as S and if we know the implied volatility, the variance of the implied geometric Brownian motion is $D^2(S_t) = S^2 e^{2rt}\left(e^{\Sigma^2 t} - 1\right)$ with the risk neutral interest rate r, that we take now for 0 for simplicity. Approximate $e^{\Sigma^2 t} - 1$ by $\Sigma^2 t$ and take t unity to obtain $D^2(S) \approx \Sigma^2 t$.
- It is reasonable to claim that the modeling fBm must have the same variance. Then $(S \cdot \Sigma)^2 = \sigma^{2H}$ from where $\sigma = (S \cdot \Sigma)^{\frac{1}{H}}$.

- Suppose now that we do not know the implied volatility but we ”know” or estimate from somewhere else σ – corresponding to the option – with high accuracy, and wish to compute the implied volatility.
- Compute first the variance with the estimated $\widehat{H}$ as $\sigma^{2\widehat{H}}$.
- Then $S \cdot \widehat{\Sigma} = \sigma^{\widehat{H}}$ from where $\widehat{\Sigma} = \dfrac{\sigma^{\widehat{H}}}{S} = \dfrac{(S \cdot \Sigma)^{\frac{\widehat{H}}{H}}}{S} = S^{\frac{\widehat{H}}{H}-1} \cdot \Sigma^{\frac{\widehat{H}}{H}}$.
- We come to the most important conclusion that it is the relative error that governs the error in the implied volatility.




Estimated S&P500 Call Option Prices on Nov 3 2023
Two days to maturity, 2009-11 Market Scenario
Black-Scholes price from implied volatility
Bid-ask midprice
Median confidence bound
95% confidence bound, AI
95% confidence bound, Higuchi
S&P value
Option Price
0
20
40
60
80
4320
4340
4360
4380
4400
Strike
Implied volatility and bid-ask midprice data from bartchart.com


Estimated S&P500 Call Option Prices on Nov 3 2023
Two days to maturity, 2007-08 Market Scenario
Black-Scholes price from implied volatility
Bid-ask midprice
Median confidence bound
95% confidence bound, AI
95% confidence bound, Higuchi
S&P value
Option Price
0
20
40
60
80
4320
4340
4360
4380
4400
Strike
Implied volatility and bid-ask midprice data from bartchart.com

Conclusions

- A NN with relatively standard architecture and a massive amount of training data is capable of more accurate in average and much faster estimation of process parameters than statistical estimators if taught on the proper type of processes.
- Skewness may degrade the accuracy of the DL estimator with rarely occuring but then unacceptable loss or risk.
- Relative errors may still be large at certain parameter setups.
- Although the training time may grow to a couple of hours, the estimation time remains an order or two smaller than that of the traditional methods.
- The fBm trained network performs also well for the fOU, but not so for fLsm.
- Transforming the evaluation data slows down the estimation process unacceptably; hence, signature transforms, or certain classes of transformers, do not represent promising alternatives

Further Study

- Further study may consider a newly invented transformer specifically for sequential data.
- Non-linear processes such as the CIR and compound models like Heston's are the subjects of our ongoing study, just like model calibration to option price data
- What feature of the process does the network learn?
- Developing a similarly efficient estimation for fLsm

Thank you very much for your attention!