Project 3: Deep Hedging

(First discussion: Nov 5; Last questions: Nov 19; Deadline: Nov 26)

Responsible: Gabriele Visentin

The goal of this project is to implement from scratch[1] the deep hedging model by Buehler et al. (2019) and to test it on simulated data from the Black–Scholes and Heston models. Submit your solution using the template provided in `template3.ipynb`. One submission per team is enough.

1. Consider the Black–Scholes model, in which the evolution of the stock price S is

$$dS_t = \sigma S_t dW_t, \quad S_0 = s_0 \in \mathbb{R}_+, \tag{1}$$

where W is a Brownian motion under a risk-neutral measure $\mathbb{Q}$ and σ is the annualized volatility.

Given an option with payoff $g(S_T)$ and maturity T, we look for its hedging strategy (i.e. a trading strategy whose value at maturity is exactly equal to the option payoff), which is the solution[2] to the following optimization problem:

$$\inf_{H \in \mathcal{H}} \mathbb{E}\left[\left(g(S_T) - p - \int_0^T H_u dS_u\right)^2\right],$$

where $\mathcal{H}$ is the set of all predictable processes and p is the risk-neutral option price.

We can solve this problem numerically on a uniform time grid $0 = t_0 < t_1 < \ldots < t_N = T$ by approximating the Itô integral with the discrete stochastic integral $\sum_{j=0}^{N-1} H_{t_j} \cdot (S_{t_{j+1}} - S_{t_j})$, where $H_{t_0}, \ldots, H_{t_{N-1}}$ are N neural networks jointly trained by minimizing the following empirical loss

$$\frac{1}{m}\sum_{i=1}^{m}\left(g\left(s_T^{(i)}\right) - p - \sum_{j=0}^{N-1} H_{t_j} \cdot \left(s_{t_{j+1}}^{(i)} - s_{t_j}^{(i)}\right)\right)^2 \tag{2}$$

on a training set $D = \left((s_{t_0}^{(i)}, s_{t_1}^{(i)}, \ldots, s_{t_N}^{(i)}), 0 \le i \le m\right)$ of m simulated paths of S.

Implement and test the model following the steps below:

(a) Use Itô's formula to check that $S_t = s_0 \exp\left(\sigma W_t - \frac{1}{2}\sigma^2 t\right)$ is a solution of (1).

(b) Simulate a training set of 10^5 paths and a test set of 10^4 paths for the asset S with parameters $N = 30$, $S_{t_0} = s_0 = 1$, $T = 1$ month $= 30/365$, $\sigma = 0.5$.

The process S can be simulated exactly on a finite grid by setting

$$S_{t_{j+1}} = S_{t_j} \exp\left(-\frac{\sigma^2}{2}\frac{T}{N} + \sigma\sqrt{\frac{T}{N}} Z_{j+1}\right),$$

where $Z_1, \ldots, Z_N$ are N iid standard Gaussian random variables.

[1] You can implement your model either in PyTorch (by adapting the code in `demo.ipynb`) or TensorFlow. In either case your model must be implemented from scratch: you are not allowed to use third-party repositories with ready-made implementations of deep hedging.

[2] The existence of a solution is guaranteed by the fact that the Black-Scholes model is complete.

(c) Implement the model by defining each $H_{t_j} : \mathbb{R} \to \mathbb{R}$ as a neural network with input S_{t_j}.

(d) Train the deep hedging model for a call option with payoff $g(S_T) := (S_T - K)^+$, with strike $K = 1$ and maturity $T = 1$ month $= 30/365$, by minimizing the loss (2) on the training set. To compute the risk neutral price p, recall that in the Black–Scholes model, the value of a European call option at time t, denoted by $C(S_t, t)$, can be computed explicitly and is a function of the value of the risky asset S_t and of time t:

$$C(S_t, t) = \Phi(d_+)S_t - \Phi(d_-)Ke^{-r(T-t)}, \tag{3}$$

where

$$d_+ = \frac{1}{\sigma\sqrt{T-t}} \left(\log\left(\frac{S_t}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t) \right),$$

Φ is the standard Gaussian cumulative distribution function, and $d_- = d_+ - \sigma\sqrt{T-t}$. You can compute the risk-neutral price $p := C(S_0, 0)$ using (3).

(e) Evaluate the hedging portfolio losses at maturity, i.e. $g(S_T) - p - \sum_{j=0}^{N-1} H_{t_j} \cdot \left(S_{t_{j+1}} - S_{t_j}\right)$, on the test set. Plot their histogram and print their empirical mean and standard deviation.

(f) In the Black–Scholes model, the hedging problem admits an analytical solution given by

$$H_t^{\mathrm{BS}}(s) = \frac{\partial C(s,t)}{\partial s}.$$

i) Derive a closed-form formula for $H_t^{\mathrm{BS}}(s)$ by computing the partial derivative above.

ii) Evaluate the hedging portfolio losses on the test set when using the analytical hedging strategy H^{BS} to rebalance the hedging portfolio at the trading dates $t_0, t_1, \ldots, t_{N-1}$. Plot their histogram and print their empirical mean and standard deviation. Compare with the results obtained in Ex.1(e). (Hint: make sure your deep hedging model performs similarly to the analytical hedging strategy)

(g) For $j \in \{0, 10, 20, 29\}$, compare in a plot the neural network function $s \mapsto H_{t_j}(s)$ and the analytical solution $s \mapsto H_{t_j}^{\mathrm{BS}}(s)$ for $s \in [0.5, 1.5]$. For what times t_j are the two functions most similar? Why?

2. Consider now the Heston model, in which the evolution of the stock price S is

$$\begin{cases} dS_t &= \sqrt{V_t} S_t dW_t, \quad S_0 = s_0 \in \mathbb{R}_+, \\ dV_t &= \tilde{\alpha}(b - V_t)dt + \sigma\sqrt{V_t}dW_t', \quad V_0 = v_0 \in \mathbb{R}_+ \end{cases} \tag{4}$$

where W and W' are two Brownian motions under a risk-neutral measure $\mathbb{Q}$ with instantaneous correlation $\rho \in [-1, 1]$.

In an incomplete market model, such as the Heston model, it is in general impossible to hedge a payoff $g(S_T)$ perfectly. Instead, we look for a hedging strategy that minimizes the hedging losses under a risk measure π of our choice. Mathematically, this requires solving the following minimization problem:

$$p = \inf_{H \in \mathcal{H}} \pi \left(g(S_T) - \int_0^T H_u dS_u \right), \tag{5}$$

where p is the minimum amount of money needed to erase the hedging risk under the risk measure π and is therefore the price that we would charge for selling the option.

As a particular risk measure, consider $\pi = \mathrm{CVaR}_\alpha$, the expected shortfall at level $\alpha \in (0, 1)$, defined as:

$$\mathrm{CVaR}_\alpha(X) = \frac{1}{1-\alpha} \int_\alpha^1 q_u(X) du,$$

where $q_u(X)$ is the u-quantile of X under $\mathbb{Q}$. Since CVaR_α admits the representation

$$\mathrm{CVaR}_\alpha(X) = \inf_{w\in\mathbb{R}}(w + \frac{1}{1-\alpha}\mathbb{E}\left[(X-w)^+\right]),$$

we can reformulate problem (5) as an empirical loss minimization problem for the following loss:

$$\frac{1}{m}\sum_{i=1}^{m}\left(w + \frac{1}{1-\alpha}\left(g(s_T^{(i)}) - \sum_{j=0}^{N-1} H_{t_j}\cdot\left(s_{t_{j+1}}^{(i)} - s_{t_j}^{(i)}\right) - w\right)^+\right), \tag{6}$$

where we minimize jointly over $w \in \mathbb{R}$ and the N neural networks $H_{t_0}, \ldots, H_{t_{N-1}}$.

Implement and test the model following the steps below:

(a) Simulate a training set of 10^5 paths and a test set of 10^4 paths for the asset S and the volatility V with parameters $N = 30$, $s_0 = 1$, $T = 1$ month $= 30/365$, $\rho = -0.3, \tilde{\alpha} = 4, b = 0.5, v_0 = 0.5, \sigma = 1$.

  i) First simulate the process V. The transition density of the process V in (4) is known explicitly (Glasserman, 2004, Chapter 3.4) and can be simulated exactly on a finite grid by setting $V_{t_{j+1}} = c\cdot C_j$, where $c = \frac{\sigma^2}{4\tilde{\alpha}}(1-e^{-\tilde{\alpha}T/N})$ and each C_j is a non-central chi-square random variable with $\frac{4b\tilde{\alpha}}{\sigma^2}$ degrees of freedom and $e^{-\tilde{\alpha}T/N}(V_{t_j}/c)$ non-centrality parameter.
  (Hint: the non-central chi-square distribution is implemented in `scipy.stats.ncx2`)

  ii) Now simulate S using the simplified Broadie–Kaya scheme (Andersen et al., 2010):

$$S_{t_{j+1}} = S_{t_j}\exp\left(\frac{\rho}{\sigma}\left(\left(V_{t_{j+1}} - V_{t_j}\right) - \tilde{\alpha}b\frac{T}{N}\right) + \left(\frac{\tilde{\alpha}\rho}{\sigma} - \frac{1}{2}\right)V_{t_j}\frac{T}{N} + \sqrt{(1-\rho^2)V_{t_j}\frac{T}{N}}Z_{j+1}\right)$$

  where $Z_1, \ldots, Z_N$ are iid standard Gaussian random variables.

(b) Implement the model by defining each H_{t_j} as a neural network with input (S_{t_j}, V_{t_j}).

(c) For the same call option considered in Ex. 1, train the model by minimizing the loss (6) on the training set for two different CVaR levels, $\alpha = 0.5$ and $\alpha = 0.99$.

(d) Compare the prices p for both values of α by evaluating the loss (6) on the test set for each trained model. Which is higher? Why?

(e) Compare the hedging portofolio losses, i.e. $g(S_T) - p - \sum_{j=0}^{N-1} H_{t_j}\left(S_{t_{j+1}} - S_{t_j}\right)$, on the test set, for both values of α. Plot their histograms and print their empirical mean and standard deviation.

References

Andersen, L. B., Jäckel, P., and Kahl, C. (2010). Simulation of square-root processes. *Encyclopedia of Quantitative Finance*, pages 1642–1649.

Buehler, H., Gonon, L., Teichmann, J., and Wood, B. (2019). Deep hedging. *Quantitative Finance*, 19(8):1271–1291.

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*, volume 53. Springer.