Project 3: Deep Hedging

(First discussion: Nov 15; Last questions: Nov 29; Deadline: Dec 6)

The goal of this project is to implement from scratch the deep hedging model introduced in [Buehler et al., 2019] by adapting the code[1] shown in the notebook `demo.ipynb` and to test it on simulated data from the Black–Scholes and Heston models. Submit your solution using the template provided in `template3.ipynb`

1. Consider the Black–Scholes model in which the risky asset S follows a risk-neutral dynamics given by:

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

where W is a Brownian motion under the unique risk-neutral measure $\mathbb{Q}$, σ is the annualized volatility, and where we have assumed that the risk-free interest rate is zero[2].

Given an option with payoff $g(S_T)$ and maturity T, the hedging problem consists in finding a self-financing trading strategy H with initial value equal to the risk-neutral price of the option and such that its value at maturity is exactly equal to the option payoff.

In a complete market model, such as the Black–Scholes model, every option admits a hedging strategy, which can therefore be represented as the solution of 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 $\sum_{j=0}^{N-1} H_{t_j} \cdot \left(S_{t_{j+1}} - S_{t_j}\right)$, 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 $D = \left((s_{t_0}^{(i)}, s_{t_1}^{(i)}, \ldots, s_{t_N}^{(i)}), 0 \leq i \leq m\right)$ 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 for the SDE (1).

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

[2] If the risk-free interest rate is non-zero, one can reduce to the zero interest rate case by working with discounted prices.

(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 $Z_1, \dots, Z_N$ are N iid standard gaussian random variables.

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

(d) Let us assume we sell a European call option with strike $K = 1$ and maturity T for its Black–Scholes price p. Train the model by minimizing the custom loss (2) on the training set for this particular payoff.

(Hint: compute the risk-neutral price p using the Black–Scholes formula)

(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

$$H_t^{\mathrm{BS}}(s) = \frac{\partial V(s)}{\partial s}$$

where $V(s) = \mathbb{E}^{\mathbb{Q}}\left[g(S_T) \mid S_t = s\right]$ is the risk-neutral price of the option at time t conditional on $S_t = s$.

i) Find a closed-form formula for $H_t^{\mathrm{BS}}(s)$ in the case of our call option.

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 $t_0, t_1, \dots, t_{N-1}$. Plot their histogram and print their empirical mean and standard deviation.

(g) For $j \in \{0, 10, 20, 29\}$, create a plot in which you draw both 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 following stochastic volatility model (known as the Heston model) with risk-neutral dynamics given by:

$$\begin{cases} dS_t &= \sqrt{V_t} S_t dW_t, \quad S_0 = s_0 \in \mathbb{R}_+, \\ dV_t &= \tilde{\alpha}(b - V_t)dt + \sigma\sqrt{V_t} dW_t', \quad V_0 = v_0 \in \mathbb{R}_+ \end{cases} \tag{3}$$

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 by trading in the risky asset only. Instead we can look for a hedging strategy that minimizes a given monetary risk measure ρ of the hedging losses and we charge the option buyer a price $\pi(g(S_T))$, which is the minimum amount of money needed to erase this risk. This requires solving the following minimization problem:

$$\pi_\rho(g(S_T)) = \inf_{H \in \mathcal{H}} \rho\left(f(S_T) - \int_0^T H_u dS_u\right). \tag{4}$$

As a particular risk measure consider $\rho = \mathrm{CVaR}_\alpha$, the expected shortfall at level α, defined as:

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

where $q_u(X) = \inf\{x \in \mathbb{R} : \mathbb{Q}(X \le x) \ge u\}$ is the u-quantile of X under $\mathbb{Q}$.

Since CVaR_α admits the representation $\mathrm{CVaR}_\alpha(X) = \inf_{w\in\mathbb{R}}(w + \mathbb{E}\left[\ell_\alpha(X-w)\right])$ with $\ell_\alpha(x) = \frac{1}{1-\alpha}x^+$, we can reformulate problem (4) as an empirical loss minimization problem for the following loss:

$$\frac{1}{m}\sum_{i=1}^{m}\left(w + \ell_\alpha\left(g(s_T^{(i)}) - \sum_{j=0}^{N-1} H_{t_j}\cdot\left(s_{t_{j+1}}^{(i)} - s_{t_j}^{(i)}\right) - w\right)\right), \tag{5}$$

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 process V in (3) is a Cox-Ingersoll-Ross process, whose transition density is known explicitly [Glasserman, 2004, Chapter 3.4], and can therefore 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]:

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

   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) As in the Black–Scholes case, consider a European call option with payoff $g(s) = (s-K)^+$ and strike $K = 1$. Train the model by minimizing the custom loss (5) on the training set for two different CVaR levels, $\alpha = 0.5$ and $\alpha = 0.99$.

(d) Compare $\pi_{\mathrm{CVaR}_{0.5}}(g(S_T))$, $\pi_{\mathrm{CVaR}_{0.99}}(g(S_T))$ and the risk-neutral price p given by the expected payoff under the risk-neutral measure $\mathbb{Q}$.

(e) Consider an agent that sells the call option at $t = 0$ for the risk-neutral price p. Evaluate on the test set the losses at maturity of the hedging portfolios, i.e. $g(S_T) - p - \sum_{j=0}^{N-1} H_{t_j}\left(S_{t_{j+1}} - S_{t_j}\right)$, for both values of α.

   i) Plot their histograms and print their empirical mean and standard deviation.

   ii) For both loss distributions, compute their empirical CVaR_α for $\alpha = 0.5$ and $\alpha = 0.99$. Which strategy has lower $\mathrm{CVaR}_{0.5}$ risk? Which strategy has lower $\mathrm{CVaR}_{0.99}$ risk?

References

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

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

[Glasserman, 2004] Glasserman, P. (2004). *Monte Carlo methods in financial engineering*, volume 53. Springer.