Project 2 – Deep hedging: the multi-asset case

Deadline : April 6, 5 pm. Please see the course Moodle page for detailed instructions on the project delivery.

The problem

- **Two tradable assets** with values S_t^1 and S_t^2. For example: two stocks in the same stock index (e.g. two components of the SP500).

- We want to **hedge an option with payoff** $g(S_T^1, S_T^2)$ depending on the values of the two assets. We delta-hedge over the time grid $t_j = j\,\frac{T}{N} = j\,\Delta t$, $j = 0, \dots, N-1$. We will consider an at-the-money (ATM) call option on the product:

$$g(S_T^1, S_T^2) = \left(\frac{S_T^1 S_T^2}{S_0^1 S_0^2} - 1 \right)^+$$

- The **hedging portfolio** now has a **delta component** δ_t^k **in each of the two assets**:

$$V_t = V_t^0 + \delta_t^1\, S_t^1 + \delta_t^2\, S_t^2 = \Big(V_t - \sum_{k=1}^{2} \delta_t^k\, S_t^k \Big) + \sum_{k=1}^{2} \delta_t^k\, S_t^k\,.$$

  The evolution in time of this self-financing portfolio is still made of the variation of the cash part (with interest rate r) plus the variation of the delta part:

$$V_{t_{j+1}} = V_{t_j} + r\Big(V_t - \sum_{k=1}^{2} \delta_t^k\, S_t^k \Big)\Delta t + \sum_{k=1}^{2} \delta_{t_j}^k (S_{t_{j+1}}^k - S_{t_j}^k), \qquad j = 0, \dots, N-1$$

- **Dynamical model**: a two-dimensional log-normal model

$$\begin{cases} \log S_{t_{j+1}}^1 = \log S_{t_j}^1 + \mu_1 \Delta t + \sigma_1 \sqrt{\Delta t}\, G_j^1 \\ \log S_{t_{j+1}}^2 = \log S_{t_j}^2 + \mu_2 \Delta t + \sigma_2 \sqrt{\Delta t}\, G_j^2\,, \end{cases}$$

  where each asset evolves in time with its own drift parameter μ_k and volatility parameter σ_k. For every j, the two increments (G_j^1, G_j^2) form a standard Gaussian vector in $\mathbb{R}^2$ with zero mean and correlation parameter ρ, that is

$$\mathbf{G}_j := \begin{pmatrix} G_j^1 \\ G_j^2 \end{pmatrix} \sim \mathcal{N}_{\mathbb{R}^2}(0, C), \quad \text{where} \quad C = \text{ covariance matrix } = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$$

  The vector increments $(\mathbf{G}_0, \mathbf{G}_1, \dots, \mathbf{G}_{N-1})$ form an independent sequence.

- Additional parameter values : $T = 1$ year, $N = 100$, $\mu_1 = 0.025$, $\mu_2 = -0.01$, $\sigma_1 = 0.22$, $\sigma_2 = 0.3$, $\rho = 0.5$, $r = 0.04$.

Approach

- We aim at learning an appropriate two-dimensional hedging strategy

$$(\delta_t^1, \delta_t^2)$$

over the time grid $\{t_j\}_j$ using a deep hedging approach: we parameterize both deltas with neural networks

$$\delta_{t_j}^1 = h_j^1(S_{t_j}^1, S_{t_j}^2)$$
$$\delta_{t_j}^2 = h_j^2(S_{t_j}^1, S_{t_j}^2),$$

where each network h_j^k, for $k \in \{1, 2\}$, is a function of the values $(S_{t_j}^1, S_{t_j}^2)$ of the two assets at current time t_j. We also parameterize the initial option price (or option premium) with an additional network $\pi_\theta(S_0^1, S_0^2)$.

- As in the 1-dim application we have considered in session 5, the criterion to be minimized is the square of the final heding error

heding error = option premium + contribution from hedging - option payoff

along the trajectories in the training set.

Your task

1. Construct a training set of N_{train} two-dimensional trajectories

$$\left(S_{t_j}^{1,i}, S_{t_j}^{2,i}\right)_{0 \le j \le N}^{1 \le i \le N_{\text{train}}}$$

for the two assets under the log-normal dynamical model above. This will require to simulate a sequence of i.i.d. Gaussian vectors $\begin{pmatrix} G_j^1 \\ G_j^2 \end{pmatrix}$ with correlation parameter ρ.

2. Construct and train a deep learning model tackling the deep hedging problem described above. You are free to choose the architecture of the neural networks entering into the model.

3. Assess and display the performance of the resulting model. Similarly to what we did in session 5, this can be done by providing some statistics of the final hedging error (mean, standard deviation, histogram,...), and displaying the profiles of the delta functions yield by the deep hedging model.