Nested Monte Carlo for Log-Variance-Gamma model

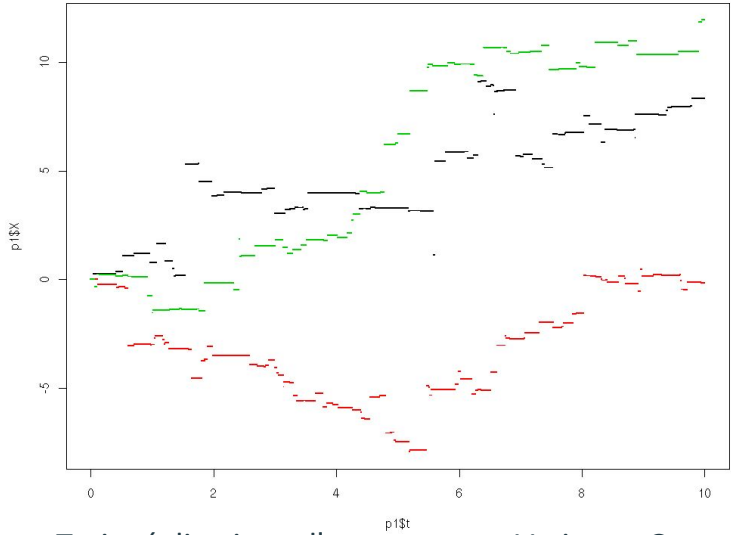
Auguste Crabeil et Gaëtan Narozniak

31 mars 2025

Génération des données - Monte Carlo

$$C(T,K,\kappa,\theta,\sigma) = \mathbb{E}\left[(Y_T - K)^+\right] \approx \frac{1}{N_{\text{traj}}} \overbrace{\sum_{N_{\text{traj}}} (Y_T^i - K)^+}^{\text{one thread}}$$

- Boucle CPU sur la durée des trajectoires T
- Parallélisation sur les 4 autres hyperparamètres sur GPU : chaque thread génère N_traj trajectoires pour un jeu de paramètres donné (pour limiter l'utilisation de cudaDeviceSynchronize)



Trois réalisations d'un processus Variance-Gamma

Notre génération de données :

strike	T	kappa	sigma	theta	benchmark	our price
1.000000	0.250000	0.035000	0.045000	-0.355000	**0.015687**	**0.015804**
1.000000	0.250000	0.035000	0.045000	-0.305000	**0.014248**	**0.014169**
1.000000	0.250000	0.035000	0.045000	-0.255000	**0.012848**	**0.012766**
1.000000	0.250000	0.035000	0.045000	-0.205000	**0.011572**	**0.011738**
1.000000	0.250000	0.035000	0.045000	-0.155000	**0.010485**	**0.010414**
1.000000	0.250000	0.035000	0.045000	-0.105000	**0.009620**	**0.009593**
1.000000	0.250000	0.035000	0.045000	-0.055000	**0.009043**	**0.009014**

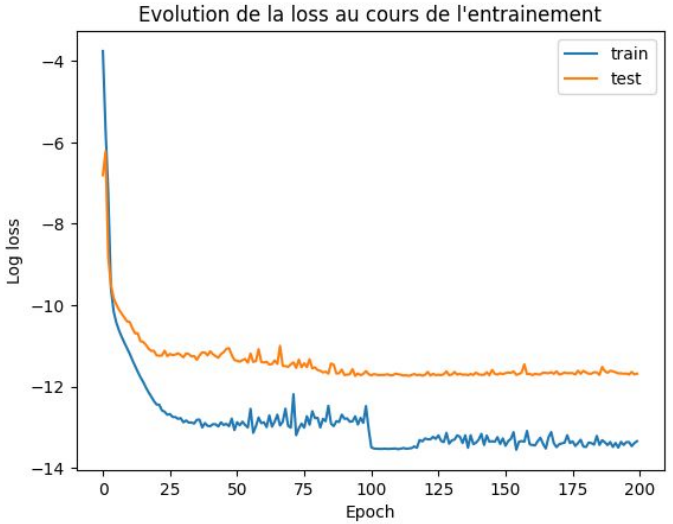
- Génération de 320.000 données pour l'entraînement sur 10.000 trajectoires
- Génération de 1024 données pour le testing sur 500.000 trajectoires

Entraînement d'un modèle de prix

- On entraîne un réseau de neurones sur les prix obtenus par méthode de Monte Carlo :

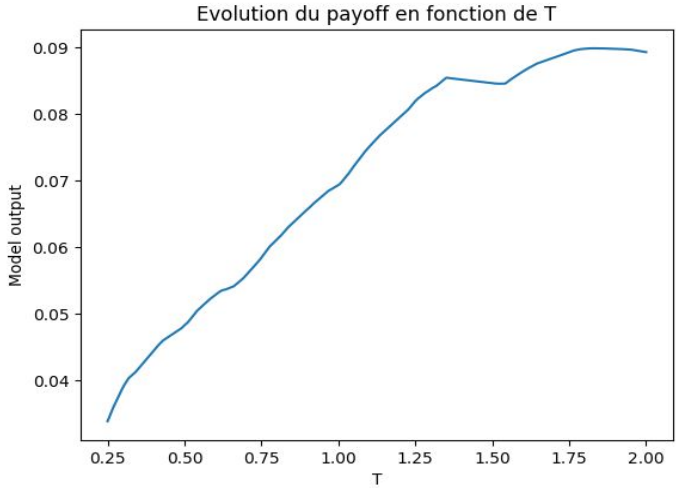
$$\phi = \arg\min_{\phi} \mathbb{E}\left[(f_\phi - \hat{C}(T, K, \kappa, \theta, \sigma))^2\right]$$

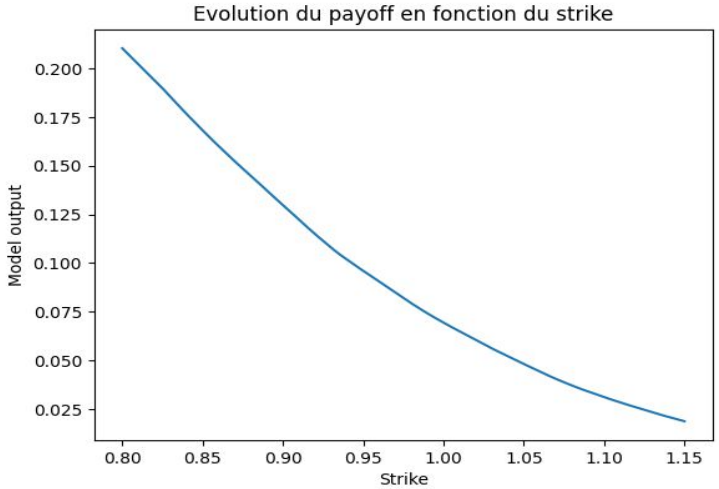
- On utilise un MLP à 4 couches cachées de taille 256



Modèle arbitrage-free ?

- Un modèle arbitrage-free doit être croissant en T, décroissant en K, et convexe en K





Régularisation du modèle

- Tirage aléatoire de 1.000.000 de points à l'intérieur de notre hypercube (T, K, kappa, sigma, theta) et calcul des gradients avec torch.autograd.grad() :

Points où la dérivée par rapport à T est négative (%)	Points où la dérivée par rapport à K est positive (%)	Points où la dérivée seconde par rapport à K est négative (%)
19.53	0	0

- On pénalise l'irrégularité du modèle lors de l'entraînement avec une nouvelle perte :

$$\mathcal{L}(\phi) = \left(f_\phi - \hat{C}\right)^2 + \lambda_1 \left(\frac{\partial f_\phi}{\partial T}\right)^- + \lambda_2 \left(\frac{\partial f_\phi}{\partial K}\right)^+ + \lambda_3 \left(\frac{\partial^2 f_\phi}{\partial K^2}\right)^-$$