Réseaux de tenseurs pour l'évaluation d'option

Soutenance de projet

Bouadjadja I.
Davion C.
Kahla A.

CentraleSupélec

Juin 2024

Encadrement

Christophe Michel
Head of Global Market Division
Quantitative Research

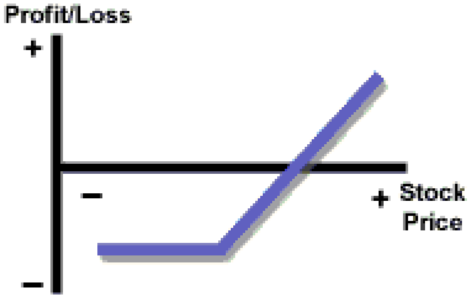
Zeno Toffano
Benoît Valiron

Contexte

- Un actif a un prix X_t
- L'acheteur de l'option peut l'acheter ou vendre à un prix fixé d'avance (prix d'exercice ou strike)
  - option call : option d'achat
  - option put : option de vente
- On distingue deux classes d'options
  - européennes : on l'exerce à maturité ($t = T$)
  - américaines : on peut l'exercer jusqu'à maturité
- Son prix est défini à l'avance

Comment évaluer le prix de l'option ?

Contexte



Sommaire

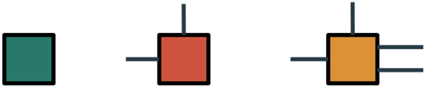
1. Outils techniques
2. Matrix Product Operator
3. Tensor trains et perspectives

1 Outils techniques : Tenseurs

Un tenseur est un tableau fini multi-indices:

$$A = (a_{i_1 i_2 \ldots i_k}), \quad i_k \in [1, N_k]$$

k est appelé le rang du tenseur.



Rang 0: scalaire   Rang 2: matrice   Rang 4

Représentation et Notation de Penrose

[1]

1 Outils techniques : Contraction de tenseurs

Soit $A = (a_{i_1 i_2 \ldots i_p \ldots i_k})$ et $B = (b_{j_1 j_2 \ldots i_p \ldots j_m})$ deux tenseurs.
On définit la contraction de A et B selon l'axe i_p comme le tenseur

$$C \text{ où } c_{i_1 i_2 .. i_k j_1 j_2 \ldots j_m} = \sum_{i_p} a_{i_1 i_2 \ldots i_k} b_{j_1 j_2 \ldots j_m}$$

Figure: Contraction de tenseur

1 Outils techniques : Contraction de tenseurs

Cette opération permet de généraliser les opérations vectorielles.

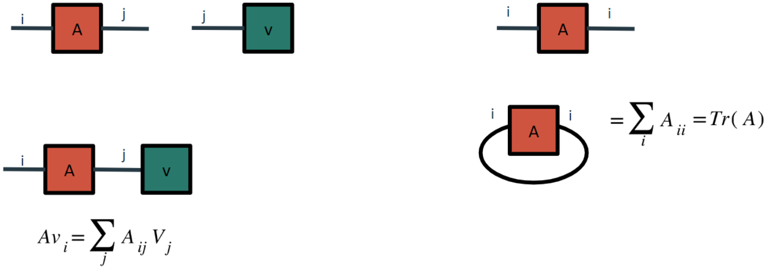


Figure: Exemple d'opérations tensorielles

Outils techniques : Équation différentielle stochastique rétrograde [4]

Soit B_t le mouvement brownien standard de $\mathbb{R}^d$, T un temps de fin et ξ une condition de fin. Une équation différentielle stochastique rétrograde (BSDE) est une équation de la forme :

$$\begin{cases} dY_t = f(t, \omega, Y_t, Z_t) - Z_t dB_t, Y_T = \xi & \textit{forme différentielle} \\ \\ Y_t = \xi + \int_t^T f(t, \omega, Y_s, Z_s) dB_s - \int_t^T Z_s dB_s & \textit{forme intégrale} \end{cases}$$

On cherche les solutions (Y_t, Z_t) parmi les processus stochastiques.

BSDE Couplées

$$
\begin{aligned}
dX_t &= \mu(t, X_t, Y_t, Z_t)\, dt + \sigma(t, X_t, Y_t)\, dB_t, \quad t \in [0, T],\\
X_0 &= \xi,\\
dY_t &= \varphi(t, X_t, Y_t, Z_t)\, dt + Z_t' \sigma(t, X_t, Y_t)\, dB_t, \quad t \in [0, T),\\
Y_T &= g(X_T).
\end{aligned}
$$

EDP associée

$$\frac{\partial u}{\partial t} = f(t, u, x, \nabla u, \nabla^2 u)$$

Lemme d'Ito [2]

$$\mathrm{Y}_t = u(t, X_t), \mathrm{Z}_t = \nabla Y_t$$

Equation de Black-Scholes-Barenblatt

L'équation de Black-Scholes-Barenblatt est, en dimension d,

$$\begin{aligned} dX_t &= \sigma diag(X_T) dB_t, \quad t \in [0, T], \\ X_0 &= \xi, \\ dY_t &= 0.05(Y_t - Z_t' X_t)dt + \sigma Z_t' diag(X_T) dB_t, \quad t \in [0, T), \\ Y_T &= ||X_T||^2. \end{aligned}$$

où $\xi = (1, 0.5, 1, 0.5..) \in \mathbb{R}^d$

Equation de Black-Scholes-Barenblatt

L'EDP associée est,

$$\frac{\partial u}{\partial t} = -\frac{1}{2} Tr(\sigma^2 diag(X_t)^2 \nabla^2 u) - 0.05(u - (\nabla u)' x)$$

Avec condition terminale $u(T, x) = ||x||^2$.
On a une solution exacte [5]:

$$u(t, x) = \exp\left((0.05 + \sigma^2)(T - t)\right) ||x||^2$$

Fonction de perte

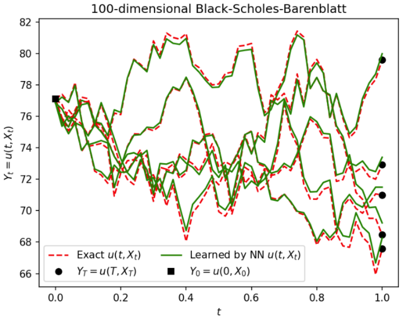
On utilse la fonction de perte suivante pour les réseaux, après avoir discrétisé [5]

$$\sum_{m=0}^{M}\sum_{n=0}^{N-1}|Y_m^{n+1}-Y_m^n-\Phi_m^n\Delta t^n-(Z_m^n)'\Sigma_m^n\Delta B_m^n|^2+\sum_{m=1}^{M}|Y_m^N-g(X_m^N)|^2$$

où N est le nombre de pas et ΔB^n est gaussien centré de variance Δt

Réseaux de Neurones

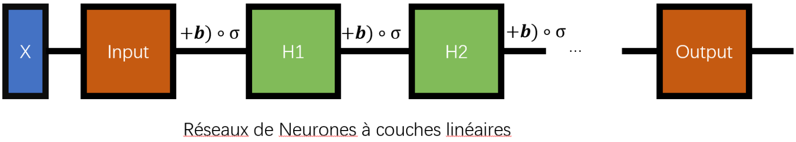
On utilise un réseau classique avec des couches linéaires pour prédire $u(t, X_t)$



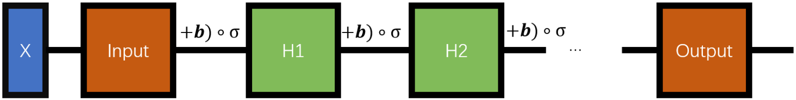
[5]

- Écart relatif de 0.5%
- Architecture: (100,256)-3*(256,256)-(256,1) (>200 000 paramètres)
- Lr=1e-3
- Itération=2*1e4
- Temps: 5800s
- Optimiser: Adam

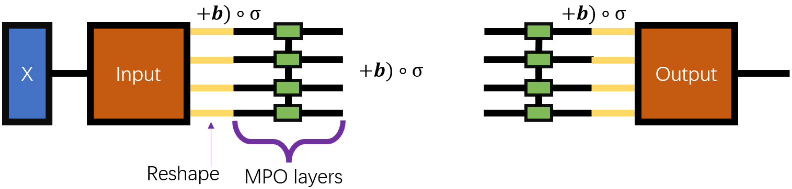
Modification de l'architecture



Modification de l'architecture



Réseaux de Neurones à couches linéaires



Réseaux de Neurones avec MPO

Matrix Product Operator

On peut représenter une matrice par le MPO, essentiellement, cela revient à faire une SVD de la matrice.

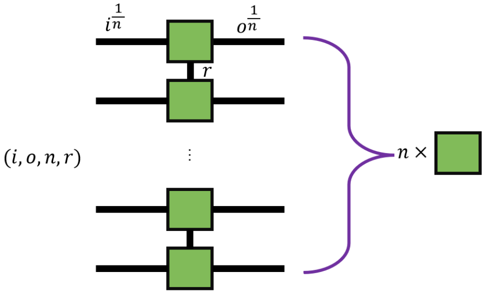
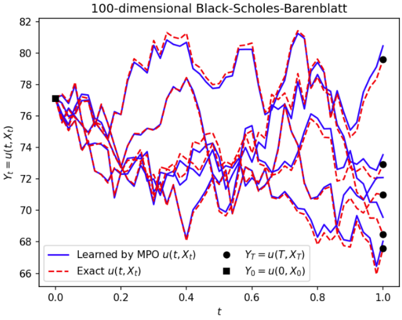


Figure: Notation pour le MPO

Avec le MPO, on a



- Écart relatif de 0.6%
- Architecture: (100,8)-3*(8,8,3,4)-(8,1) ($\sim$ 1000 paramètres)
- Lr=1e-3
- Itération=2*1e4
- Temps: 9000s
- Optimiser: Adam

Réseaux de Neurones MPO

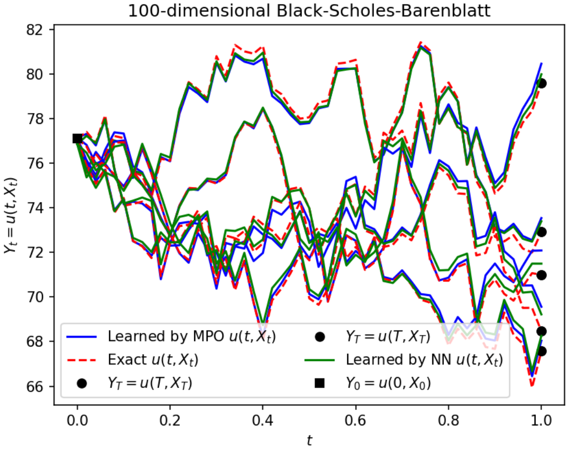


Figure: le NN et le MPO ensemble

Convergences et pertes

Figure: Perte MPO et NN

Convergences et pertes

En architecture (64,64,3,4), donc 2100 paramètres,

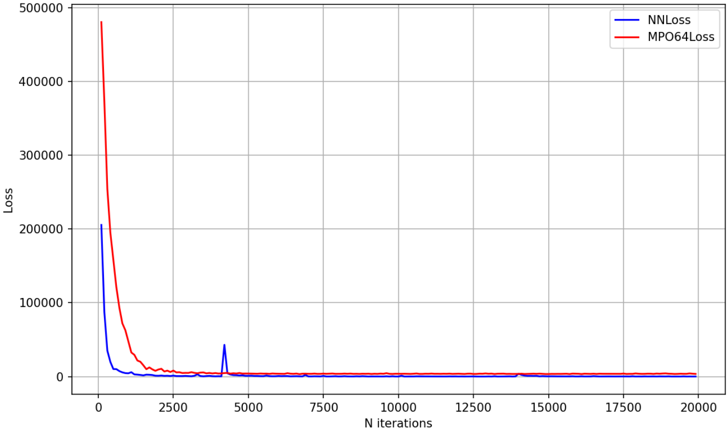


Figure: Avec plus de paramètres

Convergences et pertes

En architecture (64,64,3,4), donc 2100 paramètres,

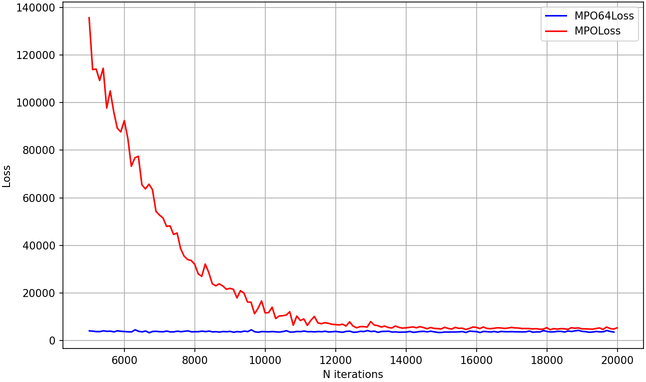


Figure: Comparaison entre les MPO

Optimisation des calculs

Avec une optimisation de la méthode de calcul de la couche MPO, on perd 45% de temps d'entrainement.

(8,8,3,4)	9000s
(8,8,3,4) OPT	5000s

Figure: Temps d'entrainement avec l'optimisation

Possiblement parallèlisable.

Une autre perspective: le train de tenseur

Soit $(t_1, t_2...t_N)$ une discrétisation du temps. Alors, la BSDE devient

$$X_{n+1} = X_n + b(X_n, t_n)\Delta t + \sigma(X_n, t_n)\xi_{n+1}\sqrt{\Delta t}$$

et alors

$$Y_{n+1} = Y_n + h_{n+1}\Delta t + Z_n \cdot \xi_{n+1}\sqrt{\Delta t}$$

où ξ_k suit une distribution gaussienne centrée de variance Δt, h une conséquence du lemme d'Ito, et $Y_N = g(X_N)$

Représentation par train

On va représenter pour tout t_i, $Y_i = V(X_i, t_i)$ par un train de tenseur. [3]

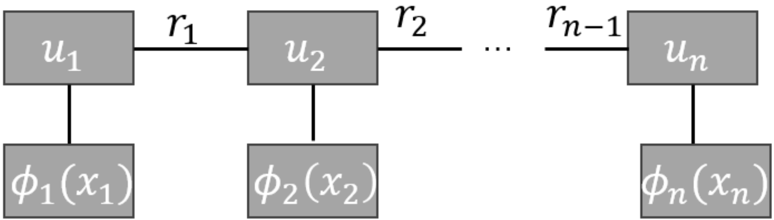


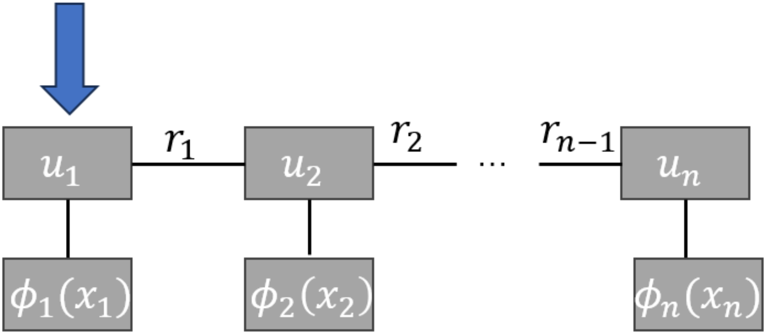
Figure: Train de tenseur

Les fonctions ϕ_k sont préchoisies, ici avec des polynômes.
$(r_1, r_2 ... r_{n-1})$ est le rang du train.

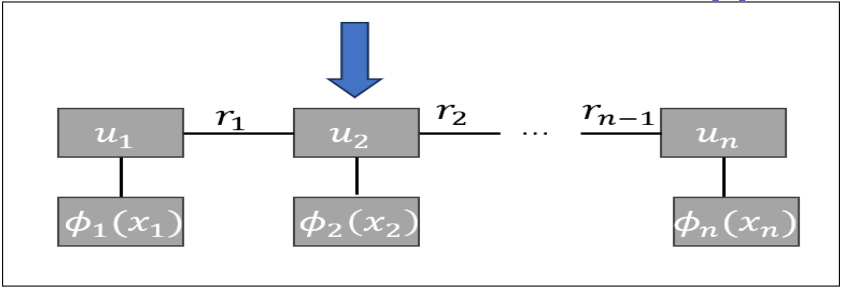
Optimisation du train

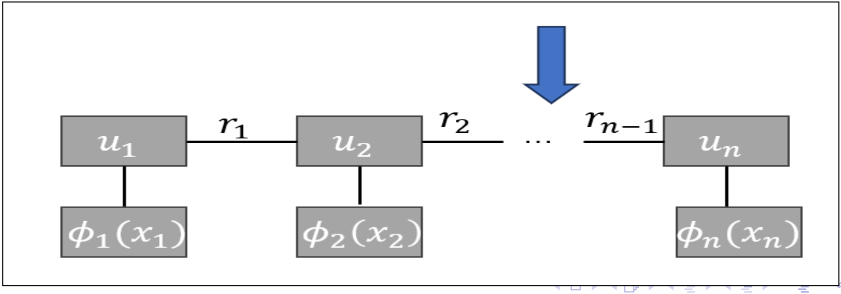
On intialise les u_i aléatoirement. Puis on optimize les u_i un par un en tentant de miniser la fonction de perte suivante : [3]

$$E[(V_n - h(X_n) - V_{n+1})^2]$$



Optimisation du train : Algorithme ALS [3]





Processus de Bachelier

- Processus de Bachelier : $dX_t = dB_t$
  Les actifs sont des mouvements browniens.
- $dV_t = \nabla_X V \cdot dB_t$
- Minimiser $E[(V_n(X_n) - V_{n+1}(X_{n+1}))^2]$
- Somme des carrés des résidus
- Bases de fonctions : Base BSpline et base des polynômes
- Solution exacte connue

Résultat pour processus de Bachelier

Pour le processus de Bachelier, $h = 0$.

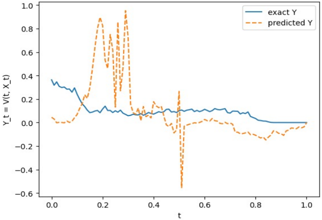


Figure: 10 simulations

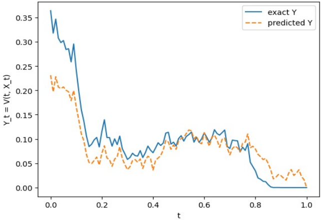


Figure: 100 simulations

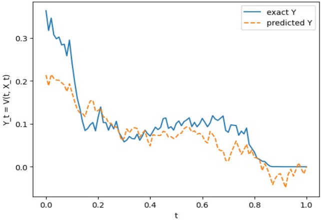


Figure: 50 simulations

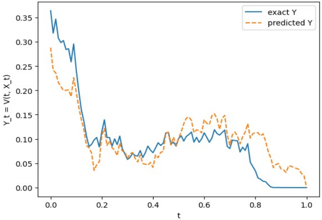


Figure: 1000 simulations

Perspectives futures

1. Étudier différentes architectures avec les MPO
2. Explorer d'autres domaines que les BSDE avec les MPO
3. Comparer les deux approches
4. Le code MPO sera sur `git@gitlab-student.centralesupelec.fr:corentin.davion/code-info-q-reseaux-de-tenseur.git`, (README en cours de rédaction), amusez-vous !

Références

Jacob Biamonte.
Lectures on quantum tensor networks, 2020.

Pardoux E.
Backward stochastic differential equations and viscosity solutions of systems of semilinear parabolic and elliptic PDEs of second order.
Springer, stochastic analysis and related topics vi edition, 1998.

Richter L., Sallant L., and Nusken N.
Solving high-dimensional parabolic pdes using the tensor train format, 2021.
arXiv:2102.11830v2 [stat.ML] 17 Jul 2021.

Nicolas Perkowski.
Backward stochastic differential equations: an introduction.

Maziar Raissi.
Forward-backward stochastic neural networks: Deep learning of high-dimensional partial differential equations, 2018.