Dauphine | PSL
UNIVERSITÉ PARIS

Université Paris Dauphine

Projet d'apprentissage statistique

MIDO

Prix de l'option européenne

Auteur :
François JORDAN
Paul CAILLERE

Enseignant référent :
Yating LIU

Année 2022-2023

Table des matières

1 Présentation du projet

Une option européenne call de maturité T et de strike K est un contrat financier qui donne à son détenteur le droit, mais non l'obligation, d'acheter un actif sous-jacent X à un prix prédéterminé (le strike K) à une date d'échéance spécifique (la maturité T). L'option européenne ne peut être exercée qu'à la date d'échéance et son payoff est $(X_T - K)_+$. La dynamique du prix de l'actif sous-jacent $(X_t)_{t\in[0,T]}$ suit l'équation différentielle stochastique :

$$dX_t = rX_t dt + \sigma X_t dB_t$$

où $r > 0$ est le taux d'intérêt sans rique, σ est la volatilité et $(B_t)_{t\in[0,T]}$ est un mouvement brownien standard.

Remarque : Dans ce projet, on fixe une fois pour toutes $x_0 = 100$, $K = 105$, $r = 0.01$ *et* $T = 1$, où $X_0 \overset{p.s.}{=} x_0$ est le prix initial de l'actif sous-jacent.

D'après le lemme d'Itô, on en déduit que

$$X_t = x_0 \exp\left((r - \frac{\sigma^2}{2})t + \sigma B_t\right).$$

Le prix d'une option européenne (Call) est alors donné par

$$C(x_0, K, r, T, \sigma) = \mathbb{E}\left[e^{-rT}(X_T - K)_+\right],$$

mais peut aussi être donné par la formule de Black-Scholes

$$C(x_0, K, r, T, \sigma) = x_0\Phi(d_1) - Ke^{-rT}\Phi(d_2)$$

où

1. Φ est la fonction de répartition de la loi normale centrée réduite $\mathcal{N}(0, 1)$,
2. $d_1 = \frac{\ln(x_0/K)+(r+\frac{\sigma^2}{2})T}{\sigma\sqrt{T}}$,
3. $d_2 = d_1 - \sigma\sqrt{T}$.

Dans ce projet, on souhaite donc déterminer la valeur du prix du call européen en fonction de la volatilité σ. Pour ce faire, nous devons apprendre de la relation

$$\sigma \mapsto C(x_0, K, r, T, \sigma)$$

à partir de données théoriques obtenues par la formule de Black-Scholes. Nos données théoriques sont donc sous la forme $\mathcal{D}_N = \{(\sigma_i, C(x_0, K, r, T, \sigma_i),\ i = 1, ..., N\}$ où les σ_i sont nos features et les $C(x_0, K, r, T, \sigma_i)$ nos labels.

2 Méthode d'apprentissage

2.1 Régression linéaire, polynômiale

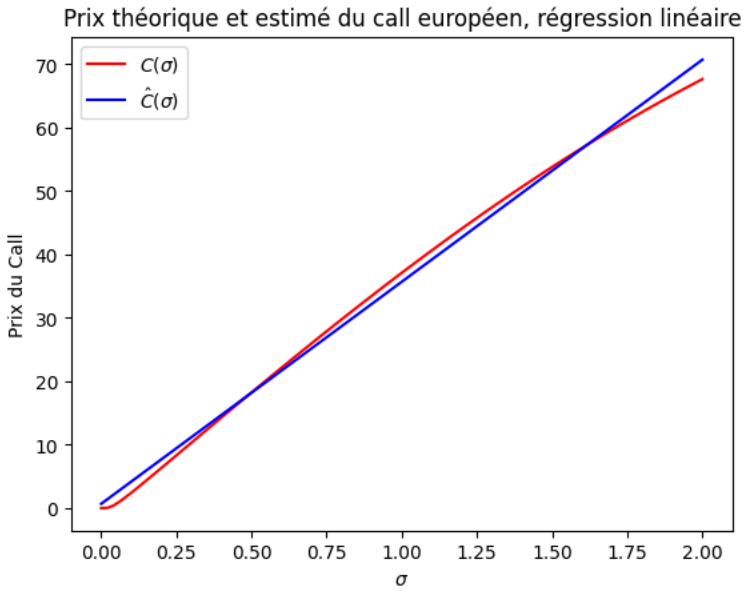
Tout d'abord, nous allons considérer un modèle de régression linéaire, puis polynômial, en choisissant au préalable un nombre arbitraire m de couples de valeurs théoriques $(\sigma, C(x_0, K, r, T, \sigma))$ obtenues par notre fonction **payoff**, après avoir généré un mouvement brownien standard sur $[0, 1]$ (par notre fonction **brownian**) en discrétisant cet intervalle en n intervalles de pas régulier $\frac{1}{n}$. Nous choisissons arbitrairement $m = 100$ et $n = 100$.

2.1.1 Régression linéaire

Dans le cas de la régression linéaire, nous estimons le prix du call européen par une fonction linéaire en σ

$$\hat{C}(x_0, K, r, T, \sigma) = \hat{a}\sigma + \hat{b}.$$

Nous avons séparé nos données en un ensemble d'entraînement de taille m et un ensemble de test de taille $\frac{m}{5}$. Nous obtenons alors pour les paramètres de la régression linéaire, une pente $\hat{a} = 35.038$ et un intercept $\hat{b} = 0.651$. Nous traçons alors la fonction "théorique" C ainsi que $\hat{C}$.



Enfin, nous obtenons un score d'environ 0.99636, ce qui est très proche de 1, et une erreur quadratique moyenne environ égale à 1.54 pour le modèle de régression linéaire. (Ces approximations ont été obtenues en faisant une moyenne des scores et des EQM sur 100 seeds différentes.)

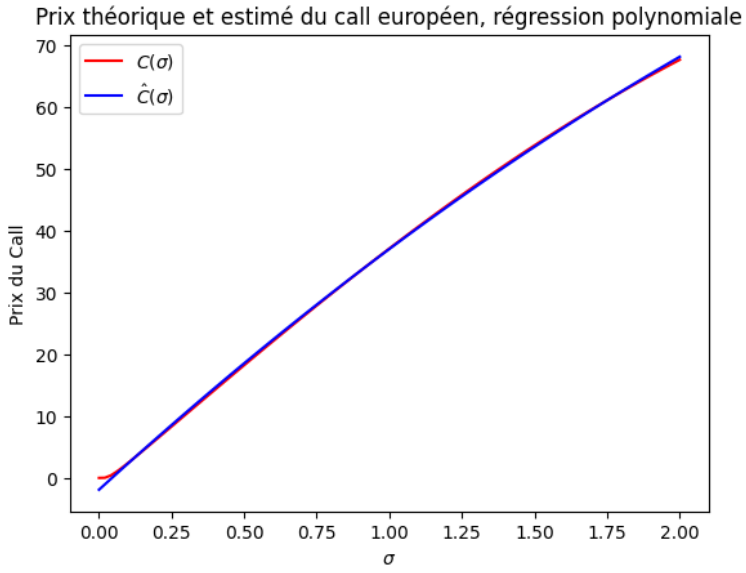
2.1.2 Régression polynômiale

Dans le cas de la régression polynômiale, nous estimons le prix du call européen par une fonction polynômiale en σ ; dans notre cas, nous nous sommes limités au cas où le

degré est 2, ce qui donne

$$\hat{C}(x_0, K, r, T, \sigma) = w_2\sigma^2 + w_1\sigma + w_0.$$

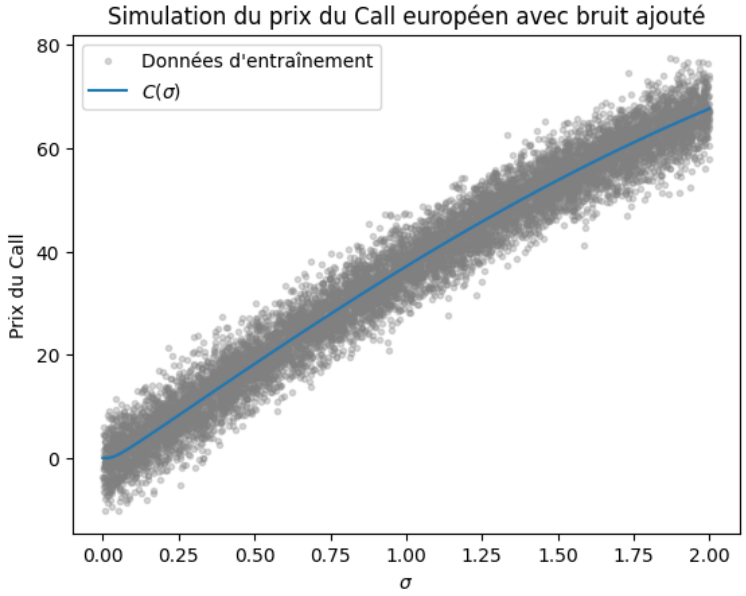
Comme pour la régression linéaire, nous avons séparé nos données en deux ensembles, un d'entraînement et l'autre de test. On obtient alors $w_2 \simeq -3.93$, $w_1 \simeq 42.9$ et $w_0 \simeq -1.95$, et obtenons le graphique suivant :



Enfin, nous obtenons un score d'environ 0.99977, et une erreur quadratique moyenne environ égale à 0.07, soit environ 22 fois moins que dans le cas de la régression linéaire. (Ces approximations ont été obtenues en faisant une moyenne des scores et des EQM sur 100 seeds différentes.)

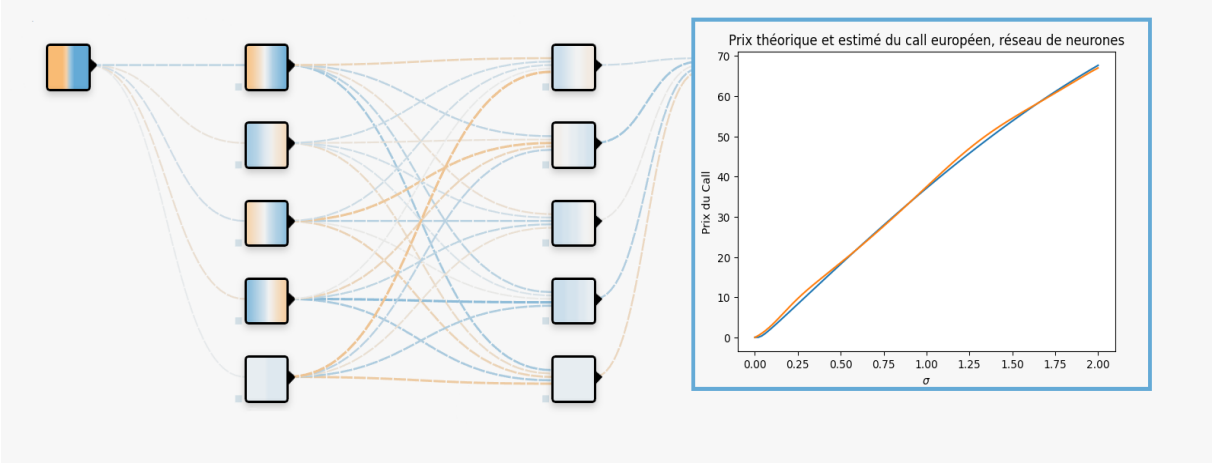
2.2 Réseau de neurones

Nous décidons enfin d'utiliser un réseau de neurones pour notre problème de régression. Nous considérons $N_{train} = 10000$ valeurs de σ choisies aléatoirement selon une $\mathcal{U}(0, 2)$, auxquelles on associe des valeurs de prix bruitées $C(x_0, K, r, T, \sigma) + \varepsilon$ (où $\varepsilon \sim \mathcal{N}(0, 4)$) comme données d'entraînement. On fait de même pour avoir nos $N_{test} = 2000$ données de test. Tous les résultats joints ci-dessous sont obtenus en ayant fixé la seed à 10.



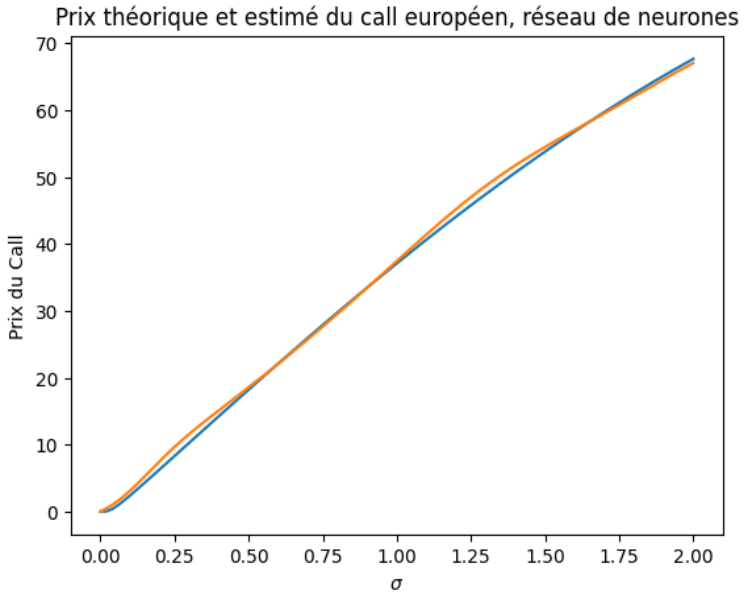
Nous décidons de réaliser un réseau de neurones avec deux couches cachées toutes deux constituées de 5 neurones, et une couche de sortie constituée d'un seul neurone (étant donnée que nous traitons un problème de régression). Notre fonction de perte est alors la fonction $l(y, y') = (y - y')^2$. Enfin, nous choisissons pour fonction d'activation la fonction sigmoïde.

Structure du réseau de neurones

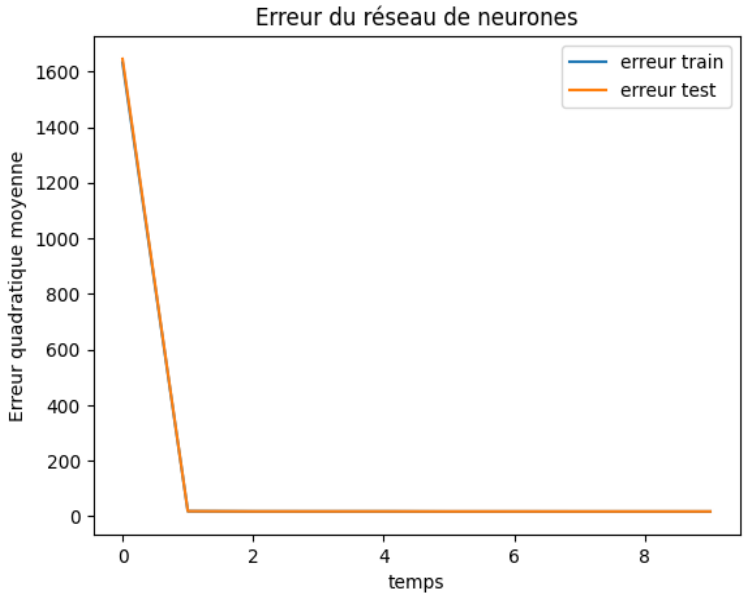


Source : *playground.tensorflow.org*

Après entraînement du réseau par descente de gradient stochastique (**batch_size** = 3000, **num_steps** = 10000, **learning_rate** = 0.01), nous obtenons une nouvelle estimation du prix du call européen.



À laquelle nous joignons les erreurs sur l'ensemble d'entraînement et de test.



Ces erreurs quadratiques moyennes sont calculées sur les données de test bruitées, si nous voulons comparer notre modèle après 10000 itérations avec les deux modèles de régression précédents, il nous faut calculer l'erreur quadratique moyenne sur les données de test sans bruits ce qui nous donne avec la seed fixée à 10 une erreur quadratique moyenne d'environ égale à 0.57 (calculé sur 10 instances d'entraînement de modèles).

Ainsi, notre modèle de réseau de neurones est, du point de vue de l'erreur quadratique moyenne, 8 fois moins précis que le modèle de régression polynômiale, mais environ 2.5 fois plus précis que le modèle de régression linéaire.

3 Conclusion

La valeur théorique du prix du call européen $C(x_0, K, r, T, \sigma)$ en fonction de σ a l'air relativement régulière et s'estime bien par une régression linéaire ou polynômiale. Les erreurs quadratiques moyennes et les scores sont convainquants quant à la bonne estimation de C et cela paraît rassurant que l'on obtienne de bien meilleurs résultats lorsque $\hat{C}$ est un polynôme de degré 2 en la variable σ.
Il serait pertinent de s'intéresser aux cas d'une régression polynômiale de degré supérieur à 2, afin d'obtenir une précision potentiellement plus grande.

Le réseau de neurones approxime lui aussi efficacement C, mais pas aussi bien que la régression polynômiale de degré 2. De plus, les temps de calculs sont significativement plus longs, ce qui nous amène à penser que la régression polynômiale est un bien meilleur compromis. Cependant, l'utilisation du réseau de neurones pourraît potentiellement s'avérer être plus pertinente si l'on trouve de meilleurs hyperparamètres (nombre de neurones par couche, nombre de couches, valeur du taux d'apprentissage), bien que tout nous laisse à penser le contraire vu les temps de calculs en jeu.