



Projet de recherche: Estimation statistique des volatilités implicites et des prix des dérivés dans les modèles stochastiques

Préparé par :

DORRA BENNOUR
HOUSSEM FENDI
AYMEN FATHALLAH

Encadré par :

M. SERGIO PULIDO NINO

Année universitaire 2019-2020

Table des matières

| | |
|---|-----------|
| Introduction Générale | 5 |
| 1 Etude théorique | 6 |
| Introduction | 6 |
| 1.1 Modèle de Black Scholes | 6 |
| 1.1.1 Hypothèses du modèle | 6 |
| 1.1.2 Formulation du modèle | 6 |
| 1.1.3 La volatilité implicite | 8 |
| 1.2 Modèle de Heston | 8 |
| 1.2.1 Hypothèses du modèle | 8 |
| 1.2.2 Formulation du modèle | 8 |
| 1.3 Méthode de Monte-Carlo | 9 |
| 1.3.1 Présentation | 9 |
| 1.3.2 Application au modèle de Heston | 9 |
| 1.4 Méthode de Fourier | 10 |
| 1.4.1 Présentation de la méthode | 10 |
| 1.4.2 Application au modèle de Heston | 11 |
| 2 Partie pratique et résultats | 13 |
| Introduction | 13 |
| 2.1 Régression linéaire | 13 |
| 2.1.1 Présentation | 13 |
| 2.1.2 Application de régression linéaire | 14 |
| 2.2 Régression polynomiale | 15 |
| 2.2.1 Présentation | 15 |
| 2.2.2 Application de régression polynomiale | 15 |
| 2.3 Réseau de neurones artificiel ANN | 16 |
| 2.3.1 Présentation | 16 |
| 2.3.2 Principe | 16 |
| 2.3.3 Application du réseau de neurones artificiel pour la dé- termination du prix de l'option et de la volatilité implicite | 18 |

| | |
|----------------------|----|
| Conclusion | 26 |
| Références | 27 |

Table des figures

| | | |
|-----|--|----|
| 1.1 | Courbe de la volatilité implicite en fonction de la moneyness par la méthode de Monte-Carlo | 10 |
| 1.2 | Courbe de la Volatilité implicite en fonction de la moneyness par la méthode de Fourier | 12 |
| 2.1 | Illustration du fonctionnement du réseau de neurones artificiel pour un neurone | 17 |
| 2.2 | Ordre de grandeur des erreurs MSE-MAE-RMSE | 20 |
| 2.3 | Distribution de l'erreur résiduelle | 21 |
| 2.4 | Ordre de grandeur des erreurs MSE-MAE-RMSE | 22 |
| 2.5 | Distribution de l'erreur résiduelle | 22 |
| 2.6 | Ordre de grandeur des erreurs MSE-MAE-RMSE | 23 |
| 2.7 | Distribution de l'erreur résiduelle | 24 |
| 2.8 | Surface de volatilité | 25 |

Liste des tableaux

| | | |
|-----|---|----|
| 2.1 | Les paramètres du modèle ANN | 19 |
| 2.2 | Performances BS-ANN sur l'ensemble de données de test . . . | 20 |
| 2.3 | Performances Black Scholes IV-ANN sur l'ensemble de données de test | 21 |
| 2.4 | Performances Heston-ANN sur l'ensemble de données de test . | 23 |

Introduction Générale

Dans un contexte économique de plus en plus exigeant, le secteur financier ne cesse d'évoluer, depuis les dernières décennies. Laissant ainsi un terrain favorable pour mener davantage de recherches afin de mieux saisir le marché et prévoir sa variabilité, qui se traduit notamment par le cours des options. Et l'un des déterminants essentiels lors du calcul du prix d'une option est l'évaluation de la volatilité implicite qui est la mesure de l'écart à la moyenne des mouvements de cours du risque d'un actif sur une période donnée. Comme chaque actif a un niveau de risque qui est à mettre en corrélation avec sa rentabilité, la volatilité implicite permet alors, d'évaluer l'évolution de ce risque dans le temps et donc de déterminer la rentabilité future du titre.

En outre, tout au long de ce projet, nous essayerons de reproduire les résultats fournis par les papier [1] et [2], c'est à dire estimer les options financières et approcher les volatilités implicites.

En effet, nous présenterons en un premier temps le côté théorique de ce projet dans le premier chapitre. Nous entamerons par la suite, dans le second chapitre, l'application des méthodes présentées auparavant, à l'aide de plusieurs techniques de machine-learning notamment le réseaux de neurones artificiel "ANN" qui sert accélérer les méthodes numériques correspondantes.

Chapitre 1

Etude théorique

Introduction

Durant ce chapitre, nous allons présenter les modèles théoriques utilisés, parmi lesquels on cite la méthode de "Black Scholes" et la méthode de "Heston", qui sont considérés parmi les principaux modèles d'approximation des prix des options ou de la volatilité implicite.

1.1 Modèle de Black Scholes

1.1.1 Hypothèses du modèle

Le modèle Black Scholes repose sur un nombre de conditions qui sont nécessaires pour la démonstration de la formule :

1. Les taux d'intérêt sont constants ou déterministes.
2. La possibilité de vendre à découvert l'actif sous jacent.
3. Il n'y a pas d'opportunités d'arbitrage.
4. Il n'y a pas de frais de transaction.

1.1.2 Formulation du modèle

Soit $(S_t)_{0 \leq t \leq T}$ le processus des prix des actifs. A chaque instant, S_t est une variable aléatoire. L'ensemble de ces variables aléatoires constitue ce qu'on appelle un processus stochastique. En finance, il y a plusieurs modèles qui peuvent être utilisés pour modéliser le prix des actifs. Dans cette partie, on

va s'intéresser au modèle de Black-Scholes.

Il est intéressant de savoir que ce modèle est définie par une dynamique :

$$dS_t = rS_t dt + \sigma S_t dW_t \quad (1.1)$$

où r est le taux d'intérêt

σ est la volatilité

W est un Mouvement brownien

Cette équation différentielle stochastique admet une solution définie par :

$$S_t = S_0 \exp\left(\sigma W_t + \left(r - \frac{\sigma^2}{2}\right)t\right) \quad (1.2)$$

$W_t \sim N(0, t)$

S_t suit une loi log-normal ce qui implique que $\ln S_t \sim N(\mu, \sigma^2 t)$ avec $\mu = \ln S_0 + \left(r - \frac{\sigma^2}{2}\right)t$

On désigne par $\Psi(t)$ le payoff à la maturité T . La valeur de l'option à un instant t est donnée par :

$$v(t, x) = \exp(-r(T-t)) \mathbb{E}[\Psi(T) | S_t = x] \quad (1.3)$$

On définit la moneyness par : $\exp(k) = \frac{K}{x \exp(r(T-t))}$

où K est le prix de l'exercice ou le Strike.

La formule explicite d'un call européen ayant une maturité T et un prix d'exercice K est :

$$v(t, x) = x\phi(d_1) - K \exp(-r(T-t))\phi(d_2) \quad (1.4)$$

où ϕ est la fonction de répartition de la loi normale centrée réduite :

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z \exp\left(-\frac{u^2}{2}\right) du \quad (1.5)$$

et

$$d_1 = \frac{1}{\sigma\sqrt{T-t}} \left[\ln\left(\frac{x}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t) \right] \quad (1.6)$$

$$d_2 = \frac{1}{\sigma\sqrt{T-t}} \left[\ln\left(\frac{x}{K}\right) + \left(r - \frac{\sigma^2}{2}\right)(T-t) \right] \quad (1.7)$$

Remarquons que $d_2 = d_1 - \sigma\sqrt{T-t}$

1.1.3 La volatilité implicite

Posons $S_t = x$ et $V^{BS}(t, x, k, t, \sigma)$ le prix d'un call dont la maturité est égale à T , ayant une log-moneyness k dans le modèle de BS et une volatilité σ . Soit $V^{obs}(t, x, k, t)$ le prix observé d'une option call de maturité T et de log-moneyness k .

La volatilité implicite $\sigma_{BS}(t, x, k, t)$ est donc tel que :

$$V^{BS}(t, x, k, t, \sigma) = V^{obs}(t, x, k, t) \quad (1.8)$$

Donc pratiquement, la volatilité implicite va être la solution de :

$$\sigma^*(K, T) = BS^{-1}(V^{obs}, S, K, \tau, r) \quad (1.9)$$

où BS^{-1} est l'inverse de la fonction de Black Scholes.

La valeur de σ^* peut être alors déterminée à l'aide des moyens itératifs numériques, l'équation (1.9) peut alors être reformulée comme un problème de recherche des zéros d'une fonction g définie par :

$$g(\sigma^*) = BS(S, K, \tau, r, \sigma^*) - V^{obs}(S, \tau, K) = 0 \quad (1.10)$$

Tout au long de ce projet, on a utilisé principalement 3 méthodes numériques pour déterminer les zéros de la fonction g qui sont : La méthode de Newton, la méthode de la sécante et finalement la méthode de Brent.

1.2 Modèle de Heston

1.2.1 Hypothèses du modèle

Ce modèle est en fait une généralisation du modèle de Black-Scholes du fait qu'il utilise une volatilité qui varie dans le temps lors du calcul du prix de l'option.

1.2.2 Formulation du modèle

Le processus de diffusion qui décrit le prix de l'action est identique à celui de Black et Scholes, à l'exception de la volatilité qui varie dans le temps. Il est définie par sa dynamique :

$$dS_t = rS_t dt + \sqrt{v(t)}S_t dW_t^1 \quad (1.11)$$

où S est le prix du sous-jacent

r est le taux de rendement espéré instantané de l'action (drift parameter)

$\sqrt{v(t)}$ est la volatilité du rendement de l'action

W_t^1 est une variable aléatoire brownienne

Le processus de diffusion de la variance $v(t)$ s'écrit comme le montre cette équation :

$$dv_t = \kappa(\theta - v_t)dt + \sigma\sqrt{v(t)}dW_t^2 \quad (1.12)$$

où θ est la moyenne de la variance à long terme

κ est un paramètre de retour à la moyenne

σ est la volatilité de la volatilité.

Condition importante :

— Pour que le problème (1.12) soit bien posé il faut que : $\kappa\theta \geq 0$.

Dans la suite, on va résoudre le problème de Heston via deux méthodes : la méthode de Monte-Carlo et celle de Fourier.

1.3 Méthode de Monte-Carlo

1.3.1 Présentation

Cette méthode est basée sur la réalisation de plusieurs scénarios possibles puis le calcul d'une moyenne. En effet il s'agit d'une application de la loi des grands nombres.

$$I = \mathbb{E}(f(X)) = \int_{-\infty}^{+\infty} f(x)d\mathbb{P} \approx \frac{1}{N} \sum_{i=1}^N f(X_i)$$

Notre objectif est de trouver la solution du modèle de Heston. A l'aide de la méthode de Monte Carlo, nous pourrions avoir une solution qui converge vers la vraie valeur. L'avantage de cette méthode se manifeste au niveau de la complexité mais nous perdons en terme de temps et de mémoire.

1.3.2 Application au modèle de Heston

On a simulé le Call Price avec la méthode de monte carlo. On sait que :

$$C(S_t, t) = \mathbb{E}(S_t - K)^+ \approx \frac{1}{N} \sum_{i=1}^N (S_t - K)$$

(1.13)

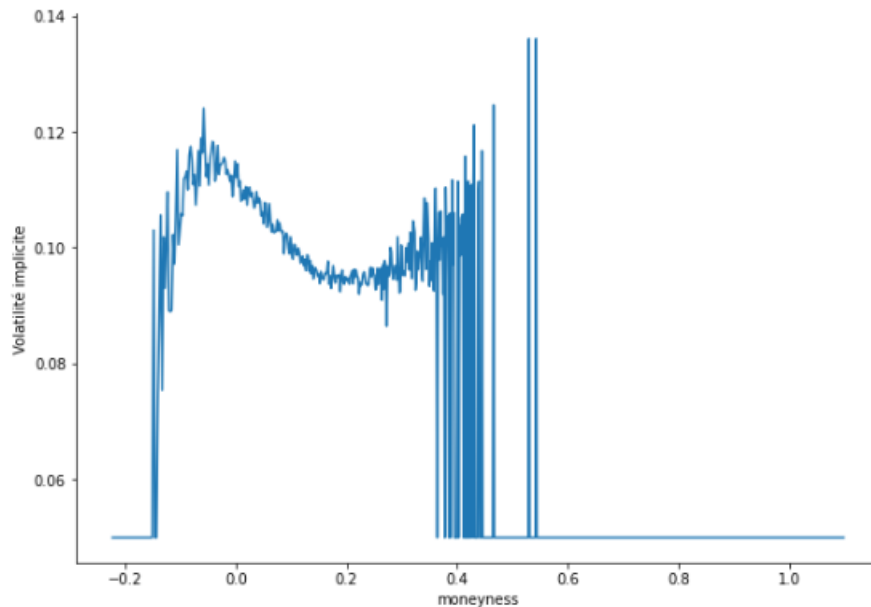


FIGURE 1.1 – Courbe de la volatilité implicite en fonction de la moneyness par la méthode de Monte-Carlo

Commentaire : D’après la figure ci-dessus, on remarque bien l’allure de la volatilité implicite, cependant on distingue beaucoup de variabilité sur la courbe ce qui met en valeur l’insuffisance de la méthode de Monte-carlo.

1.4 Méthode de Fourier

1.4.1 Présentation de la méthode

Il existe deux approches différentes permettant le calcul des prix sachant la fonction caractéristique du processus stochastique des variables d’état en se servant de la transformation de Fourier. La première approche consiste à l’utilisation de la formule d’inversion de Fourier des fonctions de distribution cumulatives. Pour cette approche on pose :

- $k = \ln(K)$
- $C(K)$ est le prix d’un call au temps $t=0$ pour une maturité T et un strike K .

- $x = \ln(S_t)$
- $\phi(X_t)$ la fonction caractéristique de X_t sous la probabilité risque neutre.
- $C_\alpha(k) = \exp(\alpha k) C(k)$

Pour une probabilité risque neutre \mathbb{P} l'expression du call s'écrit :

$$C(K) = \exp(-rt) \mathbb{E}((S_t - K)^+) = \exp(-rt) \int_k^{+\infty} (e^x - e^k) \mathbb{P}(x) dx$$

alors :

$$\lim_{x \rightarrow +\infty} C(K) = \lim_{x \rightarrow +\infty} \exp(-rt) \int_k^{+\infty} (e^x - e^k) \mathbb{P}(x) dx = S_t$$

La fonction $C(K)$ ne tend pas vers 0 donc la condition d'intégrabilité n'est pas vérifiée. On peut corriger ce problème en introduisant la fonction :

$$c(k) = e^{\alpha k} C(k)$$

$$\lim_{x \rightarrow +\infty} c(k) = \lim_{x \rightarrow +\infty} \exp(-rt) \int_k^{+\infty} (e^{\alpha k + x} - e^{(\alpha+1)k}) \mathbb{P}(x) dx = 0$$

On peut maintenant appliquer la formule d'inversion de Fourier. La transformée de Fourier de $c(k)$ est alors :

$$\hat{C}_\alpha(v) = \int_{-\infty}^{+\infty} e^{ivk} c(k) dk = e^{-rt} \frac{\phi(v - i(\alpha + 1))}{(\alpha + iv)(\alpha + 1 + iv)}$$

1.4.2 Application au modèle de Heston

Le prix d'un call est obtenu en appliquant la formule suivante :

$$c(k) = \frac{e^{-rT - \alpha k}}{\pi} \int_{-\infty}^{+\infty} \operatorname{Re} \left(\frac{\phi(v - i(\alpha + 1))}{(\alpha + iv)(\alpha + 1 + iv) e^{-ivk}} \right) dv$$

avec :

$$\phi(u) = \mathbb{E}(\exp(iu \ln(S_t))) = \frac{\exp(iu \ln(S_0) + irut + \frac{\kappa \theta t (\kappa - i\rho \sigma u)}{\sigma^2})}{(\cosh \frac{\gamma t}{2} + \frac{\kappa - i\sigma u}{\gamma} \sinh \frac{\gamma t}{2})^{\frac{2\kappa \theta}{\sigma^2}}} \exp\left(\frac{-(u^2 + iu)V_0}{\gamma \coth \frac{\gamma t}{2} + \kappa - i\rho \sigma u}\right)$$

et

$$\gamma = \sqrt{\sigma^2(u^2 + iu)(\kappa - i\rho \sigma u)^2}$$

Pour pouvoir calculer l'intégrale, on doit fixer L très grand pour obtenir à la fin le prix du call. Ceci dépend du choix de la valeur de α . On a trouvé que

l'erreur est de plus en plus petite quand α est petite. la courbe trouvée de la volatilité implicite en fonction de la moneyness est alors :

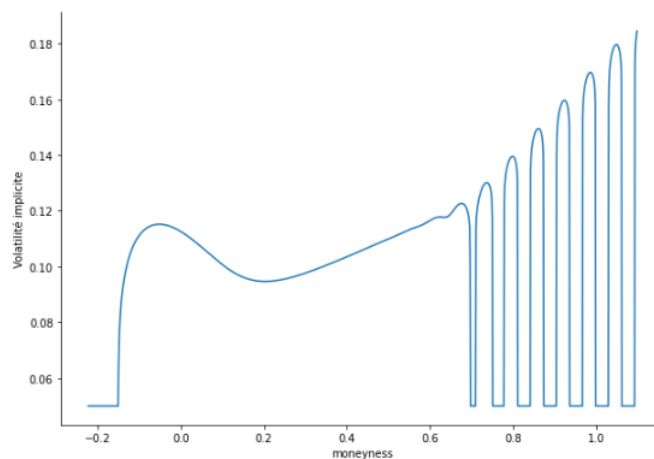


FIGURE 1.2 – Courbe de la Volatilité implicite en fonction de la moneyness par la méthode de Fourier

Commentaire : Ce graphe montre bien un défaut de la méthode de Fourier qui se manifeste au niveau des extrémités de la moneyness.

Chapitre 2

Partie pratique et résultats

Introduction

Dans le chapitre précédent, on a bien constaté l'insuffisance des méthodes classiques. Dans ce chapitre on va donc présenter les différentes méthodes statistiques qu'on va utiliser tel que : la régression linéaire multiple, la régression polynomiale et principalement le réseau de neurones artificiel.

2.1 Régression linéaire

2.1.1 Présentation

Le modèle de régression linéaire multiple est une généralisation du modèle de régression simple lorsque le nombre des variables explicatives est quelconque. Nous supposons donc que les données collectées suivent le modèle suivant :

$$Y = X\beta + \epsilon$$

où Y est un vecteur aléatoire de dimension n ,

X est une matrice de taille $n * p$ connue, appelée matrice du plan d'expérience,

β est le vecteur de dimension p des paramètres inconnus du modèle,

ϵ est le vecteur de dimension n des erreurs. On a alors l'estimateur des paramètres $(\beta_i)_{1 \leq i \leq p}$ vérifie l'équation suivante :

$$\hat{Y} = X\hat{\beta}$$

où $\hat{\beta} = (X'X)^{-1}X'Y$

2.1.2 Application de régression linéaire

L'avantage de l'algorithme de régression linéaire consiste en sa simplicité d'interprétation et sa facilité de calcul. Donc il est trivial d'utiliser cet algorithme comme premier pas pour estimer la volatilité implicite, par les variables d'entrées : Stock price, Time to maturity, Risk free rate et Call price. Tout en vérifiant qu'il existe une relation linéaire entre les paramètres d'entrée et celle de sortie.

— coefficient d'ajustement

Le coefficient d'ajustement ou coefficient de variation de la variance, noté R^2 représente la proportion de la variance totale de Y qui peut être prise en compte par les variables X, on le définit comme :

$$R^2 = 1 - \frac{\sum(Y - \hat{Y})^2}{\sum(Y - \bar{Y})^2} = 1 - \frac{SCE}{SCT}$$

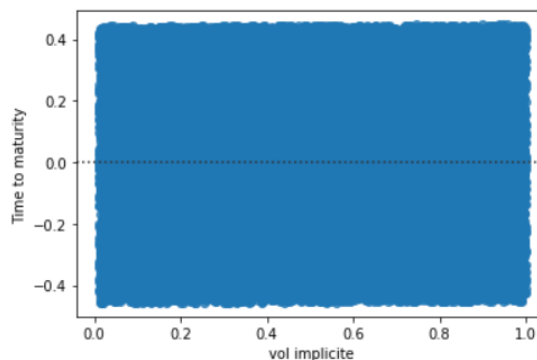
où SCE est la somme des carrés résiduels.

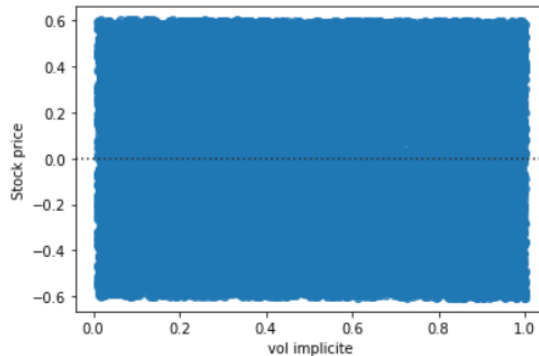
et SCT désigne la somme des carrés totale

R^2 est un indicateur qui permet de juger la qualité d'une régression linéaire. Il mesure l'adéquation entre le modèle et les données observées ou encore à quel point l'équation de régression est adaptée pour décrire la distribution des points.

— Parcelles résiduelles

En ce qui concerne les parcelles résiduelles par rapport au temps de maturité et le cours de l'option on trouve que l'erreur est répartie uniformément.





Conclusion

Le modèle de Régression Linéaire n'est donc pas valable pour l'estimation de la volatilité implicite. En effet, on a trouvé $R^2 = 0.44$ est considérée comme relativement faible.

2.2 Régression polynomiale

2.2.1 Présentation

Ce modèle est une amélioration du modèle de regression linéaire. En effet chaque Y_i s'écrit en fonction des puissances des entrées. On se contente ici des polynômes d'ordre 2.

$$Y = w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2 + w_4x_1^2 + w_5x_2^2$$

Le modèle consiste à estimer les coefficients w_i . En l'appliquant sur nos données on a trouvé R^2

2.2.2 Application de régression polynomiale

Dans notre cas, en ce qui concerne le coefficient d'ajustement, il est égal pour les données d'entraînement à 0.0004, ce qui est considéré comme très faible. Et l'erreur quadratique RMSE=0.21. Pour les données tests l'erreur quadratique est égale à 0.048 et R^2 est égal à -0.0007. En effet un R^2 négatif n'est pas une impossibilité mathématique où le signe d'un bug informatique. Cela signifie simplement que le modèle choisi, avec ses contraintes, n'est pas adéquat. En effet, R^2 n'est négatif que lorsque le modèle choisi ne suit pas la tendance des données.

2.3 Réseau de neurones artificiel ANN

2.3.1 Présentation

Un réseau de neurones artificiel ANN, est un système dont la conception est à l'origine schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques. Les réseaux de neurones étant une méthode de deep learning donc elle semble plus performante que les autres techniques statistiques. En effet ,en découvrant les relations entre les variables, ce système n'oblige pas à s'interroger sur la forme de la fonction à estimer.

2.3.2 Principe

Les réseaux de neurones sont constitués des éléments suivants :

- Une couche d'entrée, \mathbf{x}
- Un nombre arbitraire de couches cachées
- Une couche de sortie, $\hat{\mathbf{y}}$
- Un ensemble de poids W et de biais b entre chaque couche.
- Un choix de fonction d'activation pour chaque couche cachée, σ . Dans ce didacticiel, nous utiliserons une fonction d'activation Relu qui est définie par : $\frac{1}{1+\exp(-x)}$

Comme le montre la figure 2.1 ci-dessous un réseau de neurones artificiel consiste en 3 principales étapes :

1. Calcul d'une somme pondérée des entrées
2. Ajout d'un terme de biais à la somme
3. Calcul de la sortie à travers la fonction d'activation

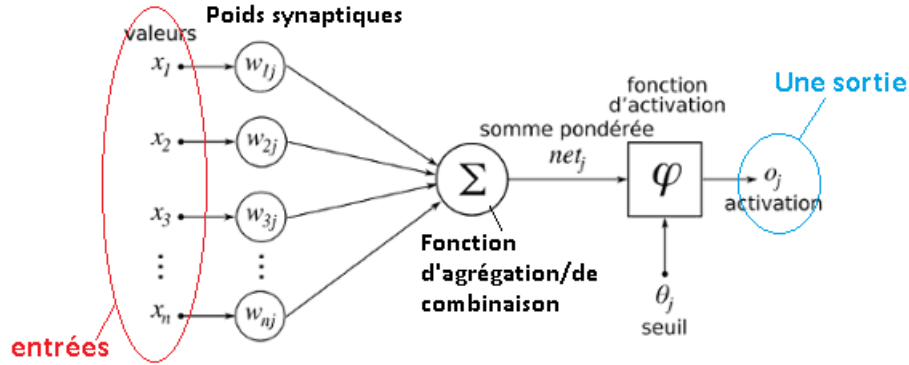


FIGURE 2.1 – Illustration du fonctionnement du réseau de neurones artificiel pour un neurone

Mathématiquement, l'algorithme Perceptron à plusieurs couches est défini par les paramètres suivants :

$$\theta = (W_1, b_1, W_2, b_2, \dots, W_L, b_L) \quad (2.1)$$

où $(W_j)_{1 \leq j \leq L}$ est la matrice des poids
 $(b_j)_{1 \leq j \leq L}$ est le vecteur biais du L-ième neurone. La sortie peut donc s'écrire :

$$y(x) = F(x|\theta) \quad (2.2)$$

Soit $z_j^{(l)}$ la valeur du j-ième neurone dans la l-ième couche, alors la fonction de transfert correspondante est écrite comme suit :

$$z_j^{(l)} = \varphi^{(l)} \left(\sum_i w_{ij}^{(l)} z_i^{(l-1)} + b_j^{(l)} \right) \quad (2.3)$$

où $z_i^{(l-1)}$ est la valeur de sortie du i-ème neurone dans la (l-1)-ième couche
 $\varphi(\cdot)$ est la fonction d'activation qui ajoute un caractère non linéaire au système

Avec $w_{ij}^{(l)} \in \mathbf{W}_l$ et $b_j^{(l)} \in \mathbf{b}_l$

Remarque

Pour $l=0$, $z^{(0)} = x$ est la couche d'entrée

Pour $l=L$, $z^{(L)} = y$ est la couche de sortie

Selon le théorème de l'approximation universelle, un ANN à couche cachée

unique avec un nombre suffisant de neurones peut approximer n'importe quelle fonction continue. Ce processus d'apprentissage permet d'apprendre les poids et les biais optimaux, afin de minimiser l'erreur ou Loss définie par :

$$L(\theta) := D(f(x), F(x|\theta)) \quad (2.4)$$

où $D(f(x), F(x|\theta))$ est la distance entre la fonction objectif f , et la fonction approchée par le réseau de neurones F . Le processus peut-être donc ramené à un problème d'optimisation :

$$\arg \min_{\theta} L(\theta|(x, y)) \quad (2.5)$$

Avec le couple d'entrée-sortie (x, y) et $L(\theta)$ sont données.

Afin de résoudre ce problème, plusieurs méthodes de gradient de descente ont été appliquées, entre autres le gradient de descente stochastique, et ses variantes Adam et RMSprop. Ces algorithmes d'optimisation commencent à partir des valeurs initiales, et progresse de façon à minimiser l'erreur quadratique. La formulation de l'algorithme est la suivante :

$$\left\{ \begin{array}{l} W \leftarrow W - \eta(i) \frac{\partial L}{\partial W} \\ b \leftarrow b - \eta(i) \frac{\partial L}{\partial b} \\ i = 0, 1, 2, \dots \end{array} \right. \quad (2.6)$$

où η est le taux d'apprentissage.

Nous allons tout au long de la prochaine partie détailler l'application des réseaux de neurones sur les modèles de Black Scholes et Heston.

2.3.3 Application du réseau de neurones artificiel pour la détermination du prix de l'option et de la volatilité implicite

En tant qu'approche basée sur les données, la qualité d'un ensemble de données a un impact sur les performances du modèle résultant. Nous avons procédé lors de la création de la base de donnée à la génération d'un nombre arbitraire d'échantillons. Lorsque l'échantillon de données pour les paramètres d'entrée est disponible, nous sélectionnons les méthodes numériques appropriées pour générer les résultats des données entraînées. Pour le modèle Black-Scholes, les prix des options sont obtenus à l'aide des formules fermées. Pour le modèle Heston, les prix sont calculés par la méthode Fourier expliquée dans la partie 1.4. Une fois les prix Heston déterminés, la méthode de Brent

sera utilisée pour trouver la volatilité implicite correspondante. L'ensemble des données est divisé en deux groupes, 90% seront des données d'entraînement et 10% des données de test. Dans notre projet on a fixé la taille de la base de données à 50.000 échantillons.

Dans cette section, nous présentons alors un réseau de neurones pour approximer une fonction pour les modèles financiers. La méthodologie suivie est la suivante :

Algorithme

1. Générer les paramètres d'entrées
2. calculer la sortie correspondante (prix ou volatilité implicite) par les différentes méthodes mentionnées auparavant dans ce rapport.
3. Diviser la base de données en deux groupes : données d'entraînement, et données test.
4. Appliquer le modèle du réseau de neurones

En ce qui concerne les paramètres du modèle du réseau de neurones artificiel, la table 2.1 résume les différents éléments d'entrée que nous avons utilisé.

| Paramètres | Valeurs |
|--|----------------|
| Couches cachées | 4 |
| Nombre de neurones dans chaque couche | 400 |
| Fonction d'activation | ReLU |
| Fonction d'activation de la couche de sortie | Linéaire |
| Initialisation | Glorot Uniform |
| Optimisateur | Adam |
| taille du lot | 1024 |

TABLE 2.1 – Les paramètres du modèle ANN

Résultats de l'application du ANN au modèle de Black Scholes

1. Call price comme sortie

On applique tout d'abord le réseau de neurones artificiel, à une base de données constituée des entrées : cours de l'option "Stock price", date d'échéance "Time to maturity", le taux d'intérêt "Risk free rate" et la volatilité "volatility", et le prix du call "Call price" comme sortie.

On a alors obtenu des résultats assez intéressants. En effet, les erreurs sont de l'ordre de 10^{-4} comme le montre la table 2.2 ci-dessous et la figure 2.2.

| BS-ANN | MSE | RMSE | MAE |
|--------------|------------------------|--------|--------|
| Données test | $1.5050 \cdot 10^{-5}$ | 0.0037 | 0.0031 |

TABLE 2.2 – Performances BS-ANN sur l’ensemble de données de test

Avec :

— $MSE = \frac{1}{n} \sum (y_i - \hat{y}_i)^2$

— $RMSE = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2}$

— $MAE = \frac{1}{n} \sum |y_i - \hat{y}_i|$

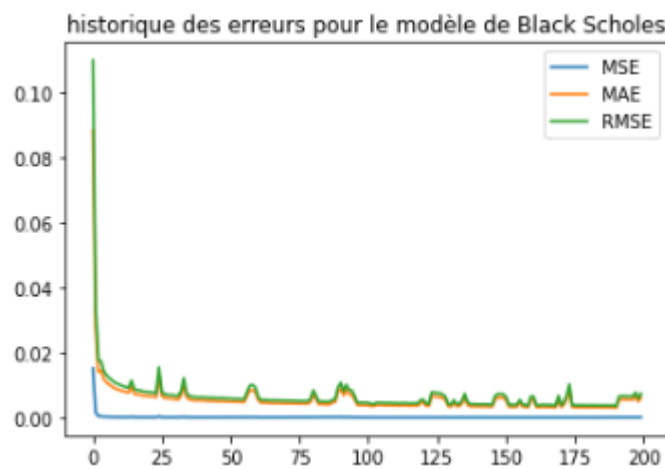


FIGURE 2.2 – Ordre de grandeur des erreurs MSE-MAE-RMSE

Quant à l’erreur résiduelle définie par la différence entre la sortie souhaitée et la sortie obtenue elle a bien l’allure d’une gaussienne, comme le montre la figure 2.3.

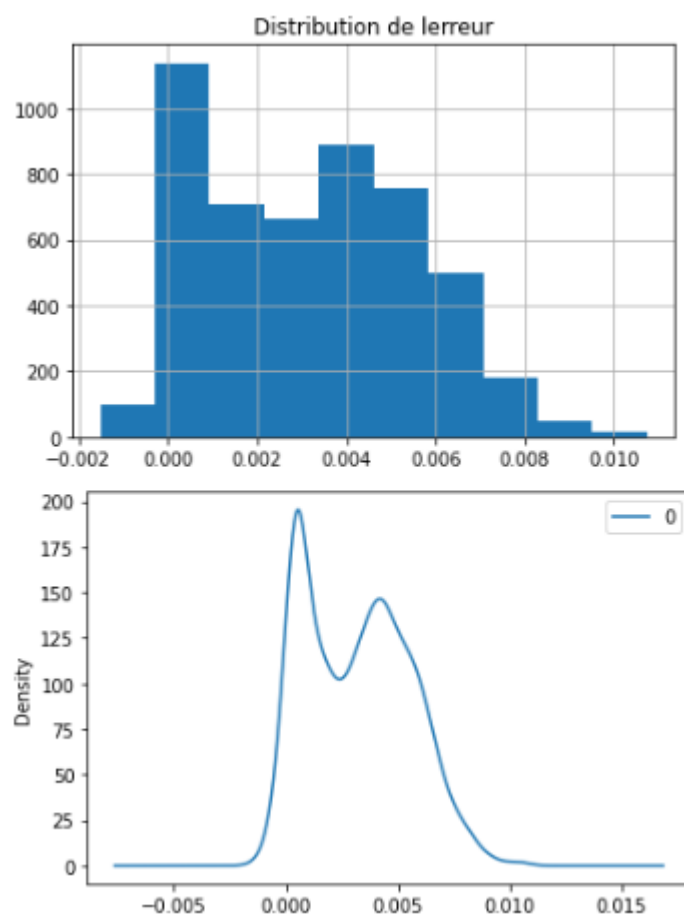


FIGURE 2.3 – Distribution de l'erreur résiduelle

2. Volatilié implicite comme sortie

Cette fois-ci, rien ne change, sauf la sortie, en effet le prix du call devient une entrée et la volatilité implicite devient une variable cible. On trouve alors, des erreurs d'ordre de grandeur indiquées dans le tableau ci-dessous :

| BS-ANN | MSE | RMSE | MAE |
|--------------|----------------------|--------|--------|
| Données test | $7.17 \cdot 10^{-5}$ | 0.0073 | 0.0061 |

TABLE 2.3 – Performances Black Scholes IV-ANN sur l'ensemble de données de test

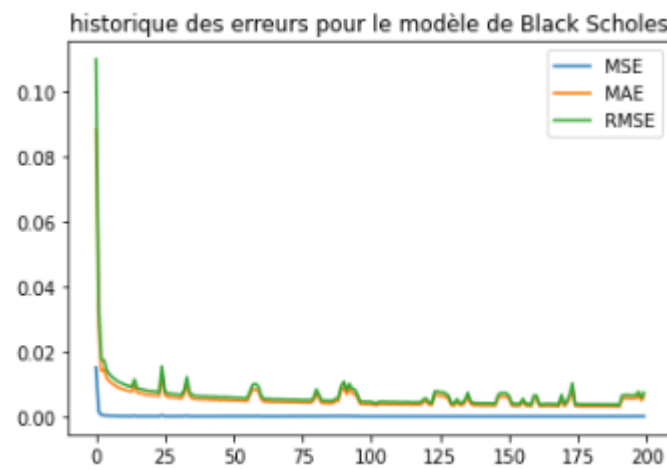


FIGURE 2.4 – Ordre de grandeur des erreurs MSE-MAE-RMSE

L'erreur résiduelle a alors une allure gaussienne (figure 2.5) :

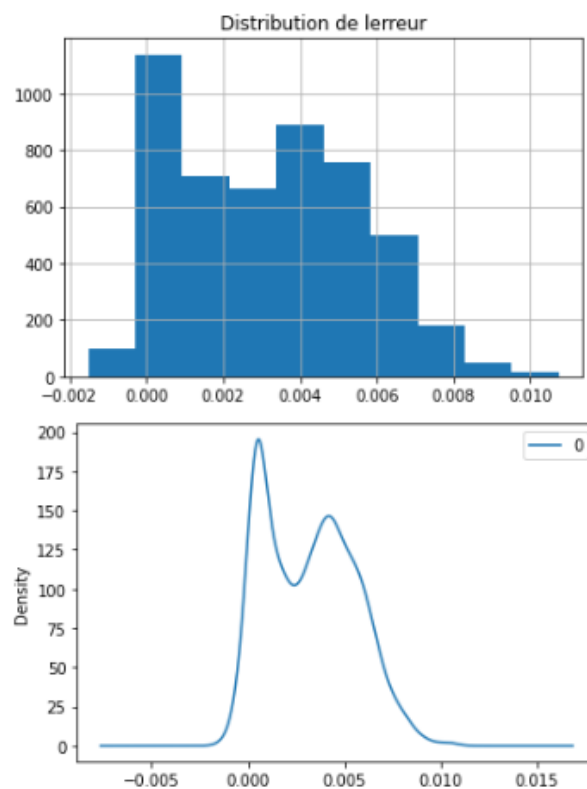


FIGURE 2.5 – Distribution de l'erreur résiduelle

Résultats de l'application du ANN au modèle de Heston

1. Call price comme sortie

On applique maintenant le réseau de neurones artificiel, à une base de données constituée des entrées : la variance initiale "initial variance", date d'échéance "Time to maturity", le taux d'intérêt "Risk free rate" et la volatilité "volatility", "moneyness", "correlation", "reversion speed", "long a" et le prix du call "Call price" comme sortie. On alors obtenu des résultats assez intéressants. En effet, les erreurs sont aussi de l'ordre de 10^{-4} comme le montre la table 2.3 ci-dessous :

| BS-ANN | MSE | RMSE | MAE |
|--------------|----------------------|--------|--------|
| Données test | $5.17 \cdot 10^{-5}$ | 0.0069 | 0.0051 |

TABLE 2.4 – Performances Heston-ANN sur l'ensemble de données de test

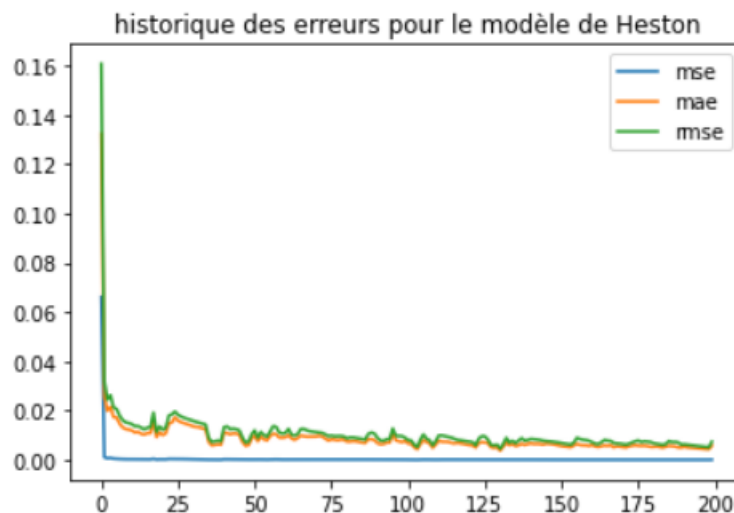


FIGURE 2.6 – Ordre de grandeur des erreurs MSE-MAE-RMSE

De même l'erreur résiduelle a bien l'allure d'une gaussienne beaucoup plus fine que celle pour le modèle de Black et Scholes comme le montre la figure 2.5 :

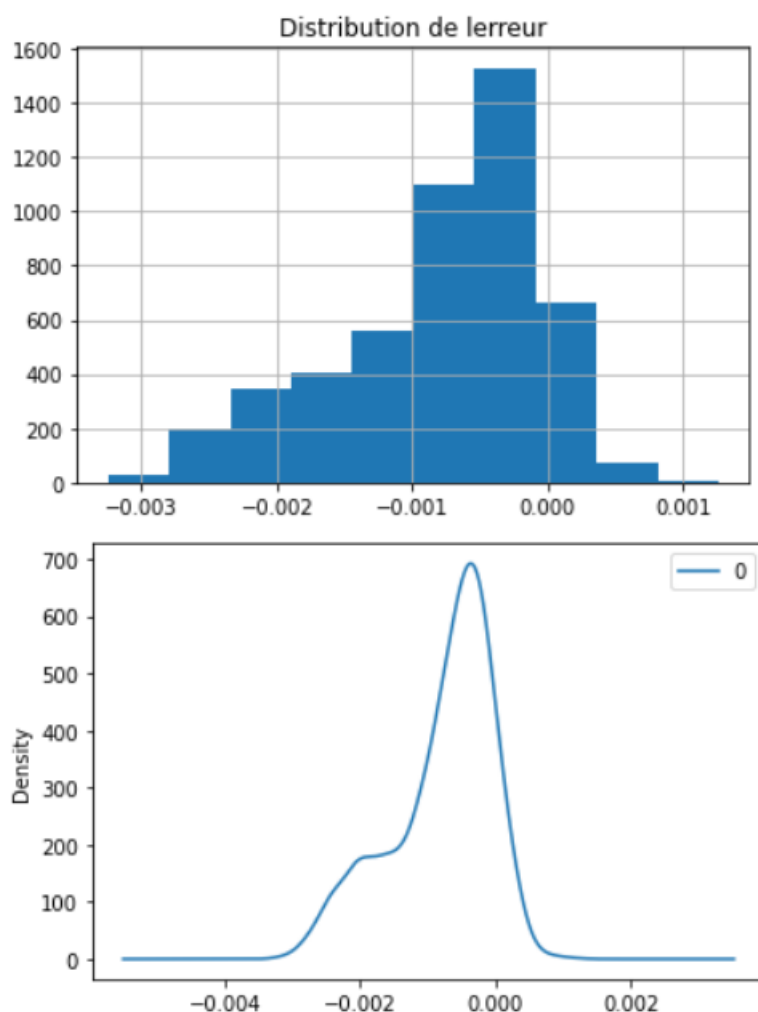


FIGURE 2.7 – Distribution de l'erreur résiduelle

2. Volatilité implicite comme sortie

La volatilité implicite par le modèle de Heston est déterminée en appliquant la méthode de Brent aux prix des options obtenus auparavant par l'application des réseaux de neurones. On obtient alors la surface de volatilité suivante :

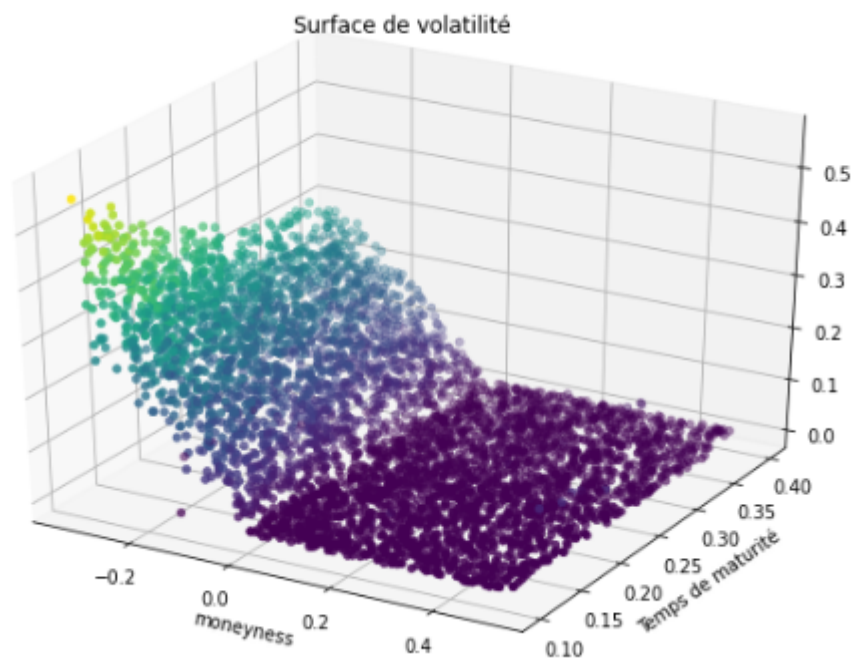


FIGURE 2.8 – Surface de volatilité

Conclusion On peut donc conclure, que le réseau de neurones artificiel peut approcher les prix aussi bien que les volatilités implicite efficacement.

Conclusion

Tout au long de ce projet, nous avons appliqué pas mal de modèles afin de raffiner les outils de pricing actuels non seulement en terme de temps de compilation mais aussi en terme de précision et rigueur. En effet, les modèles classique tels que la régression linéaire et la régression polynomiale semblent donner un résultat mais qui n'est pas considéré assez rigoureux. A l'encontre de ces méthodes, nous avons traité l'efficacité des réseaux de neurones artificiels lors de son application aux modèles financiers de Black Scholes et Heston, et les résultats trouvés sont considérés assez exacts et précis vu que les erreurs de prédiction sont très faibles. De plus, le temps de compilation de ces algorithmes est très faible comparé à celui des autres méthodes, ce qui rend le réseau de neurones le meilleur candidat pour le pricing des actifs financiers.

Bibliographie

- [1] Shuaiqiang Liu, Cornelis W. Oosterlee, et Sander M.Boht. *Pricing options and computing implied volatilities using neural networks*.
- [2] Blanka Horvath, Aitor Muguruza, Mehdi Tomas. *Deep Learning Volatility A deep neural network perspective on pricing and calibration in (rough) volatility models*