

# Deep Learning pour la calibration du modèle de Heston.

Study carried out by the Quantitative Practice  
Special thanks to Hamza ANDAM.



# Summary

Introduction	1
1. Modèle de Heston	1
1.1. Définition du modèle	1
1.2. Impact des paramètres du modèle sur la volatilité implicite	1
1.2.1 Impact de $v$	1
1.2.2 Impact de $\sigma$	2
1.2.3 Impact de $\rho$	3
1.2.4 Impact de $\kappa$	3
2. Calibration classique du modèle de Heston	3
2.1. Calibration naïve	3
2.2. Calibration en cascade	3
3. Calibration du modèle de Heston par des réseaux de neurones	4
3.1. Les réseaux de neurones artificiels	4
3.2. Deep Calibration du modèle de Heston	5
3.3. Résultats de la calibration avec le réseaux de neurones artificiels	6
Conclusion	8
References	8

## Introduction

Un produit dérivé est un instrument financier qui tire sa valeur d'un autre actif financier. Pour que les traders de ces produits financiers dérivés puissent les acheter et les vendre efficacement et fournir des liquidités au marché, ils doivent être en mesure de les valoriser au juste prix. Le cadre traditionnel dans lequel s'inscrit le pricing est celui de Black-Scholes [1], généralement appelé cadre Black-Scholes (BS), qui est un modèle mathématique permettant de trouver des solutions à forme fermée pour la valeur des prix des options vanilles. Cependant, depuis l'introduction du modèle BS, les marchés financiers y compris les marchés des produits dérivés, ont connu une expansion et un développement considérables. L'évolution du marché des produits dérivés, qui a suivi de près le développement de la technologie informatique moderne, a eu un impact considérable sur le pricing des produits dérivés.

Motivés par les limites du modèle Black-Scholes, d'autres modèles ont vu le jour. Certains modélisent la volatilité, elle-même, comme un processus stochastique, comme le modèle de Heston. Ce modèle, introduit par Heston [2], est une extension du modèle de Black-Scholes qui modélise explicitement la volatilité, c'est-à-dire la variation de toute série de prix mesurée par l'écart type des rendements logarithmiques, par une équation différentielle stochastique.

Cependant, les modèles financiers deviennent complexes car ils dépendent de plusieurs paramètres, ce qui introduit un problème de calibration du modèle. La calibration du modèle est le processus par lequel les paramètres d'un certain modèle sont, généralement, déterminés en minimisant la différence entre ce qui est calculé par le modèle et ce qui est observé dans le marché (prix, volatilités implicites, ...). Ainsi, l'applicabilité de ces modèles est limitée quand le temps nécessaire pour la calibration est long ou quand la précision de la méthode est faible. Par conséquent, il existe un besoin pour des méthodes de calibration plus rapides et, idéalement, plus précises.

Les ordinateurs modernes ont permis l'accès à des méthodes numériques complexes pour pallier la lenteur de résolution de certains problèmes existants. L'une de ces méthodes introduites récemment est appelée le Machine Learning [3]. Le Machine Learning est le processus par lequel les systèmes informatiques utilisent des algorithmes et des modèles statistiques pour effectuer certaines tâches, indépendamment d'instructions explicites sur la manière de l'exécuter. En général, les algorithmes de Machine Learning utilisent des données pour l'apprentissage, sur lesquelles l'algorithme construit une fonction qu'il cherche à optimiser à partir des données. Finalement, le modèle est ensuite généralement utilisé pour prédire des données hors échantillon. La qualité de ces prédictions permet de valider le modèle.

Un exemple de tels modèles est les réseaux de neurones artificiels (ANN), qui sont des systèmes informatiques qui rappellent les réseaux neuronaux biologiques. En tant que tel,

la procédure d'apprentissage des ANN est vaguement similaire à celle du cerveau humain. Ainsi, un tel réseau apprend, sans instructions spécifiques à la tâche, en alimentant une entrée et en activant les neurones du réseau pour arriver à une conclusion sur ce que devrait être le résultat.

## 1 Modèle de Heston

Le modèle de Heston est aussi l'un des modèles les plus populaires pour l'évaluation des options. Il peut être calibré en utilisant les prix des options vanille et ensuite utilisé pour évaluer les produits dérivés exotiques pour lesquels il n'existe pas de formule d'évaluation fermée. Le modèle de Heston permet de faire une calibration sur les données réelles du marché en utilisant la solution de forme semi-fermée pour les prix des options vanilles (calls et puts) européennes.

### 1.1 Définition du modèle

En se référant à [4], les dynamiques du sous-jacent et de la volatilité sous la mesure  $\mathbb{Q}$  sont données par le système d'équations stochastiques suivant :

$$\begin{cases} dS_t = S_t(rdt + \nu\sqrt{V_t}dW_t) \\ dV_t = \kappa(\theta - V_t)dt + \sigma\sqrt{V_t}dZ_t \end{cases} \quad (1)$$

où :

- $W$  et  $Z$  sont deux mouvements Browniens corrélés, de corrélation  $\rho$  :

$$d\langle W_t, Z_t \rangle = \rho dt, \quad (2)$$

- $\nu$  est la valeur initiale de la volatilité puisque nous supposons que  $V_0 = 1$ .
- $\theta$  est la valeur attendue de  $V_t$ , c'est-à-dire la variance moyenne du prix à long terme.
- $\kappa$  est le taux de retour à la moyenne de  $\nu_t$  vers la moyenne de long terme.
- $\sigma$  est la "vol du vol", c'est-à-dire la variance de  $V_t$ .

Nous supposons que  $\theta = 1$  et nous notons qu'aucun des paramètres n'est dépendant du temps.

### 1.2 Impact des paramètres du modèle sur la volatilité implicite

Pour comprendre l'impact des paramètres du modèle sur les volatilités implicites, nous traçons à maturité fixée les volatilités implicites pour les strikes 50, 60, ..., 150 où  $S_0 = 100$ .

#### 1.2.1 Impact de $\nu$

Plus la volatilité initiale est élevée, plus le prix du call est important. Au niveau de la volatilité implicite, la plus grosse différence est observée at the money et se réduit lorsque l'on est dans la monnaie. Voir la figure 2

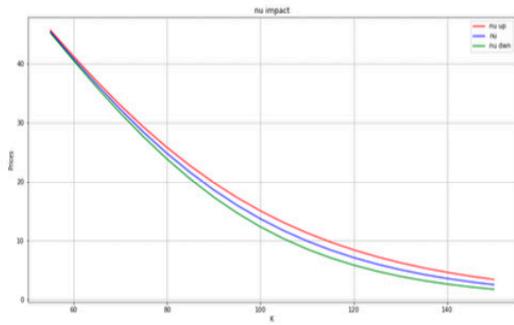


Figure 1: L'impact de  $\nu$  sur le prix d'une option d'achat

Enfin, les courbes dans la figure 1 illustrent la croissance du prix d'un call lorsque la volatilité initiale est croissante. Cela est attendu car plus  $\nu$  est grand, plus la probabilité d'exercer l'option est importante.

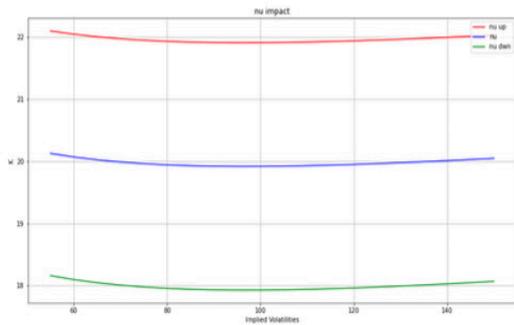


Figure 2: L'impact de  $\nu$  sur la volatilité implicite d'une option d'achat

### 1.2.2 Impact de $\sigma$

Sur la figure 3, on peut remarquer deux choses :

1. Lorsque le prix de l'action est très loin du prix d'exercice (deeply out of the money), l'option ayant un grand  $\sigma$  coûte plus cher.
2. A l'inverse, lorsque le prix de l'action est proche du prix d'exercice et même dans la monnaie, l'option ayant un petit  $\sigma$  coûte plus cher.

En fait, il faut visualiser qu'une volatilité de volatilité importante entraîne une plage de volatilité énorme, ce qui est confirmé par la figure 4.

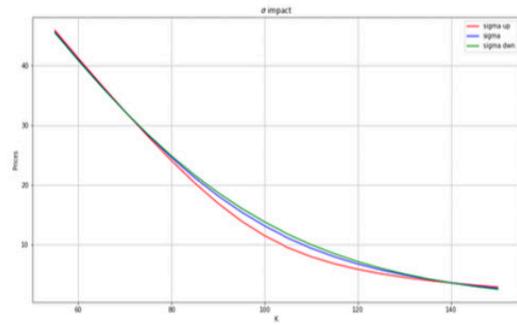


Figure 3: L'impact de  $\sigma$  sur le prix d'une option d'achat

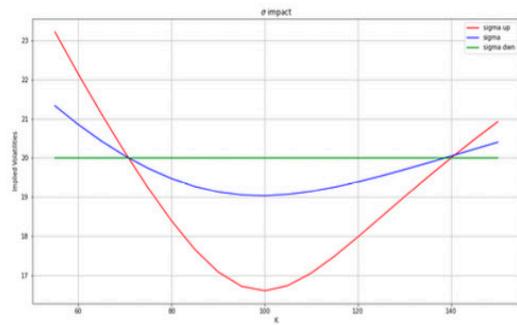


Figure 4: L'impact de  $\sigma$  sur la volatilité implicite d'une option d'achat

### 1.2.3 Impact de $\rho$

Si  $\rho = 0$  alors on a un smile de volatilité classique mais, si  $\rho \geq 0$  alors la courbure de la volatilité est décalée (figure 6).

On remarque que les courbes se croisent en un même point. Si on trace une droite passant par ce point et perpendiculaire à l'axe des abscisses on pourrait dire que les courbes pour  $\rho = 0.5$  et  $\rho = -0.5$  sont symétriques :

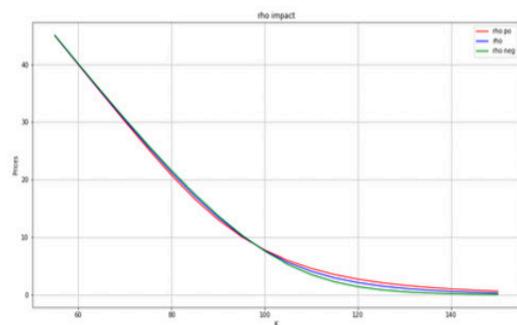


Figure 5: L'impact de  $\rho$  sur le prix d'une option d'achat

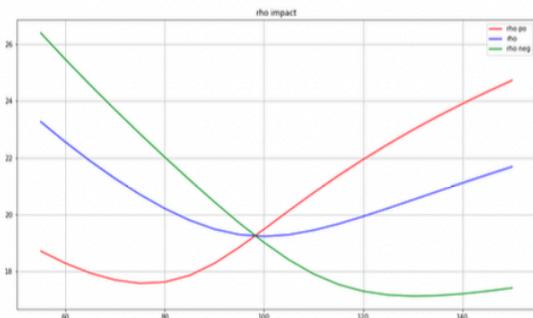


Figure 6: L'impact de  $\rho$  sur la volatilité implicite d'une option d'achat

### 1.2.4 Impact de $\kappa$

Nous observons que ce paramètre a un impact sur la vitesse de retour à la moyenne. Voir la figure 7: La courbe des volatilités implicites rouge ( $T = 1$ ) converge vers la courbe noire ( $T = 10$ ) plus rapidement quand  $\kappa$  est grand.

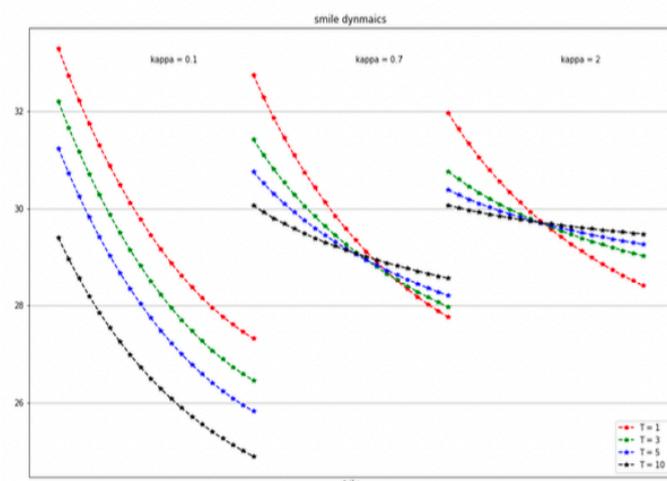


Figure 7: L'impact de  $\kappa$  sur la volatilité implicite d'une option d'achat

## 2 Calibration classique du modèle de Heston

La calibration des paramètres  $\nu$ ,  $\kappa$ ,  $\sigma$  et  $\rho$  est un problème difficile. Il faut non seulement trouver une solution qui reproduit les données, mais elle doit être stable dans le sens où l'ajout de nouvelles données ne doit pas changer les paramètres radicalement.

Les données peuvent être un mélange d'options d'achat et de vente pour différentes échéances.

Nous faisons la calibration en utilisant des volatilités implicites (plutôt que des prix) puisqu'elle est plus stable.

### 2.1 Calibration naïve

Nous optimisons les paramètres du modèle d'Heston en minimisant l'erreur quadratique entre les volatilités implicites du modèle et les volatilités implicites du marché pour des options de strikes  $K$  et de maturités  $T$ . Mathématiquement, nous devons optimiser la fonction objective suivante :

$$F_0(\Theta) = \sum_{(T,K)} (\text{MarketVol}_{(T,K)} - \text{ModelVol}_{(T,K)}(\Theta))^2,$$

où  $\Theta$  est le vecteur des paramètres  $\rho$ ,  $\sigma$ ,  $\nu$ ,  $\kappa$  sous les contraintes suivantes :

- La positivité de la volatilité initiale :  $\nu \geq 0$ ;
- La positivité de la vitesse de retour à la moyenne :  $\kappa \geq 0$ ;
- La positivité de la volatilité de la volatilité :  $\sigma \geq 0$ ;
- La corrélation doit vérifier la condition :  $1 \geq \rho \geq -1$ ;
- la condition de Feller :  $2\kappa\theta \geq \sigma^2$ .

### 2.2 Calibration en cascade

Cette méthode scinde l'optimisation en deux étapes :

1. Dans un premier temps, nous procédons, la calibration de  $\rho$ ,  $\sigma$  et  $\nu$  :

(a) Nous fixons  $\kappa$  à une valeur arbitraire et nous considérons la fonction objective suivante :

$$F_1 : (\rho, \sigma, \nu) \rightarrow \|\text{MarketVol}_1 - \text{ModelVol}(\rho, \sigma, \nu, \kappa = \kappa_0)\|_2$$

(b) Le choix logique pour les volatilités dans cette étape est les volatilités implicites pour des maturités  $T$  entre 0.1 et 1.

(c) Cette étape permet d'avoir  $\rho^*$ ,  $\sigma^*$ ,  $\nu^*$ .

Ce choix est justifié par le fait que l'impact de  $\kappa$  pour les volatilités de maturité petite est négligeable.

2. Et dans un deuxième temps, nous calibrons  $\kappa$  :

(a) Nous considérons la fonction objective suivante :

$$F_2 : \kappa \rightarrow \|\text{MarketVol}_2 - \text{modelVol}(\rho^*, \sigma^*, \nu^*, \kappa)\|_2$$

(b) Le choix logique pour les volatilités dans cette étape est les volatilités implicites pour des maturités  $T$  entre 3 et 5.

(c) Cette étape permet d'avoir  $\kappa^*$

Le résultat final de l'optimisation est constitué des résultats des deux étapes :  $\rho^*$ ,  $\sigma^*$ ,  $\nu^*$  et  $\kappa^*$ .

### 3 Calibration du modèle de Heston par des réseaux de neurones

#### 3.1 Les réseaux de neurones artificiels

Un réseau de neurones artificiels est un système inspiré du réseau de neurones du cerveau et est utilisé pour approximer des fonctions. Il consiste en des groupes interconnectés de nœuds, qui représentent les neurones du cerveau, avec des connexions qui représentent les synapses du cerveau. Dans [5], l'auteur en fait une présentation plus approfondie.

Il existe plusieurs type de réseaux de neurones artificiels. Par exemple, on trouve les réseaux de type Feed Forward et les réseaux de neurones convolutifs.

Mathématiquement, un réseau de neurones artificiels est une application, non linéaire par rapport à ses paramètres  $\theta$  qui associe à une entrée  $x$  une sortie  $y = F(x, \theta)$ .

Comme indiqué précédemment, les neurones constituent le principal composant des ANNs.

Un neurone artificiel est une fonction  $F_j$  de l'entrée  $x = (x_1, \dots, x_d)$  pondérée par un vecteur de poids de connexion  $w_j = (w_{j,1}, \dots, w_{j,d})$ , complétée par un biais de neurone  $b_j$ , et associée à une fonction d'activation  $\phi$ , à savoir

$$y_j = F_j(x) = \phi(\langle w_j, x \rangle + b_j).$$

Plusieurs fonctions d'activation peuvent être considérées :

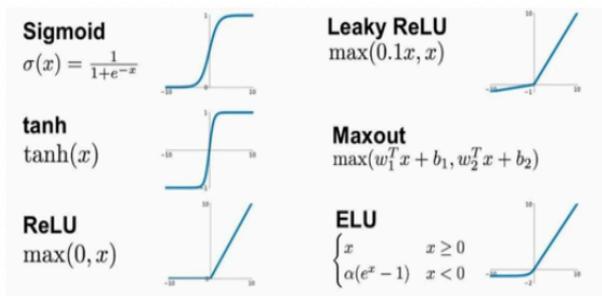


Figure 8: Des exemples de fonctions d'activation

Un réseau de neurones Feed Forward est une structure composée de plusieurs couches cachées de neurones où la sortie d'un neurone d'une couche devient l'entrée d'un neurone de la couche suivante. De plus, la sortie d'un neurone peut également être l'entrée d'un neurone de la même couche ou d'un neurone des couches précédentes. On peut appliquer une fonction d'activation différente de celle des couches cachées en fonction du type de problème que l'on a à résoudre : régression ou classification.

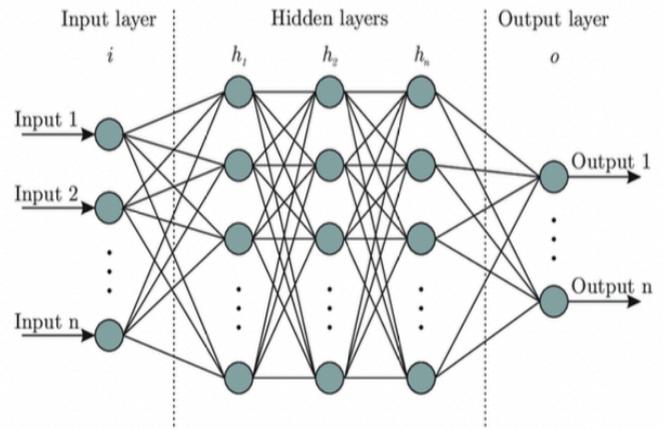


Figure 9: Un réseau de neurones Feed Forward

Une fois que l'architecture du réseau a été choisie, les paramètres (les poids  $\omega_j$  et les biais  $b_j$ ) doivent être estimés à partir d'un échantillon d'apprentissage. Comme d'habitude, l'estimation est obtenue en minimisant une fonction de perte avec un algorithme de descente de gradient.

Comme nous l'avons mentionné précédemment, l'intérêt des ANNs réside dans leur capacité à "apprendre". Le but d'un ANN est d'approximer une fonction  $F : \mathbb{R}^n \mapsto \mathbb{R}^n$ , ce qui, en termes simples, est fait par l'entraînement avec des données étiquetées, c'est-à-dire que l'entrée est fournie au ANN et que des sorties sont produites, celles-ci sont ensuite comparées avec les valeurs réelles.

Pour exprimer cela en termes mathématiques, considérons une fonction de coût  $C(\cdot)$ , des paramètres de réseau  $w$ , des entrées  $x_1, x_2, \dots$  et  $y_1, y_2, \dots$ , où  $y_i = F^*(x_i)$  pour tous  $i \geq 1$ . Le poids qui approxime  $F(x, \hat{w}) \approx F^*(x)$  est trouvé par :

$$\hat{w} = \operatorname{argmin}_{\text{all } w} \sum_{i=1}^{N_{\text{Train}}} C(F(x_i, w) - y_i).$$

On note que l'apprentissage est effectué à l'aide d'un ensemble d'apprentissage de taille  $N_{\text{Train}}$ . La solution  $\hat{w}$  est trouvée en utilisant la descente de gradient et la backpropagation.

Pour certains types de données, notamment pour les images, les réseaux de neurones type feed forward ne sont pas bien adaptés. En effet, ils sont définis pour des vecteurs comme données d'entrée, donc, pour les appliquer aux images, il faut transformer les images en vecteurs, perdant ainsi de l'information. Avant le développement de l'apprentissage profond pour la vision par ordinateur, l'apprentissage était basé sur l'extraction des variables d'intérêt, appelées caractéristiques, mais ces méthodes nécessitent beaucoup d'expérience pour le traitement des images. Les réseaux de neurones convolutifs ont révolutionné le traitement d'images, et supprimé l'extraction manuelle des caractéristiques. Les CNN sont maintenant largement utilisés pour la classification d'images, la segmentation d'images, la reconnaissance d'objets et la reconnaissance des visage [6].

### 3.2 Deep Calibration du modèle de Heston

Dans cette section, nous allons considérer la calibration d'un modèle de Heston avec un réseau de neurones Feed Forward et un réseau de neurones convolutif, nous limitons notre calibration aux trois paramètres  $\kappa$ ,  $\sigma$  et  $\rho$ . Les travaux dans cette partie sont inspirés des travaux dans [7].

Nous construisons notre ensemble de données de manière synthétique : l'ensemble d'apprentissage considéré est de taille 10000 que nous séparons en deux parties : 8000 pour l'entraînement et 2000 pour le test.

Les paramètres	$\kappa$	$\sigma$	$\rho$
borne inférieur	1	0.15	-0.9
borne supérieur	10	1	-0.1

Table 1: Les bornes inférieures et supérieures des paramètres du modèle considéré pour la calibration avec les ANNs.

L'ensemble de validation considéré est de taille 2000.

Ensuite, nous évaluons, à l'aide du modèle de Heston, les prix des calls sur une grille de maturités et de stikes:

$$\begin{cases} T = [0.2, 0.3, 0.6, 1.2, 1.5, 1.8, 2.], \\ K = [60, 70, 80, 90, 100, 110, 120, 130, 140] \end{cases}$$

1. Tout d'abord, nous commençons par générer aléatoirement les paramètres du modèle de Heston : Nous générons aléatoirement chaque paramètre à l'aide d'une distribution uniforme entre les bornes inférieures et les bornes supérieures.
2. Une fois les paramètres générés, nous gardons ceux satisfaisant la condition de Feller.
3. Nous calculons la surface des prix correspondante à chaque combinaison de paramètres gardée.
4. Nous extrayons les volatilités implicites.

Le temps nécessaire pour générer ces données synthétiques est de l'ordre de 4 heures.

L'architecture des réseaux de neurones artificiels que nous proposons (et certains hyperparamètres utilisés, comme le learning rate, les epochs et la batch size) a été optimisée en minimisant l'erreur sur l'ensemble de validation.

Nous considérons l'architecture suivante pour l'apprentissage avec le réseau de neurones Feed Forward et nous utilisons la fonction ELU comme fonction d'activation pour tous les neurones :

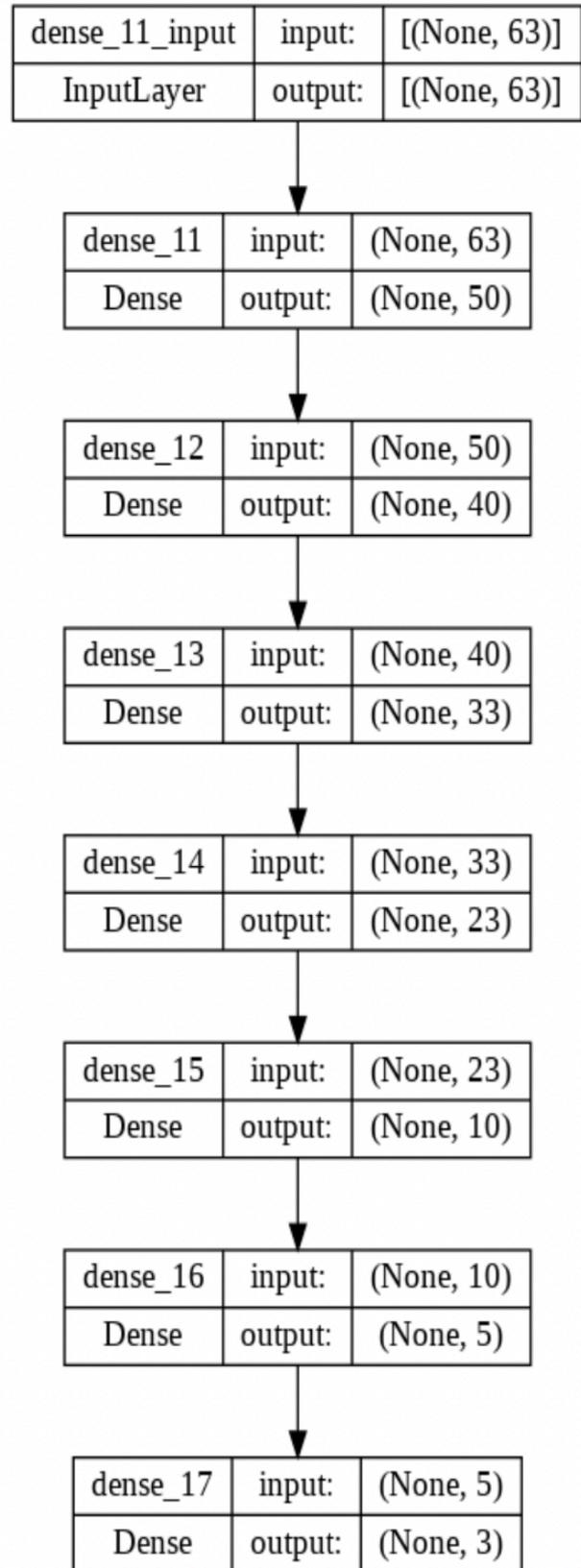


Figure 10: L'architecture du réseau de neurones Feed Forward

Pour l'apprentissage avec le réseau de neurones convolutifs, nous utilisons l'architecture suivante, avec la fonction d'activation tangente hyperbolique :

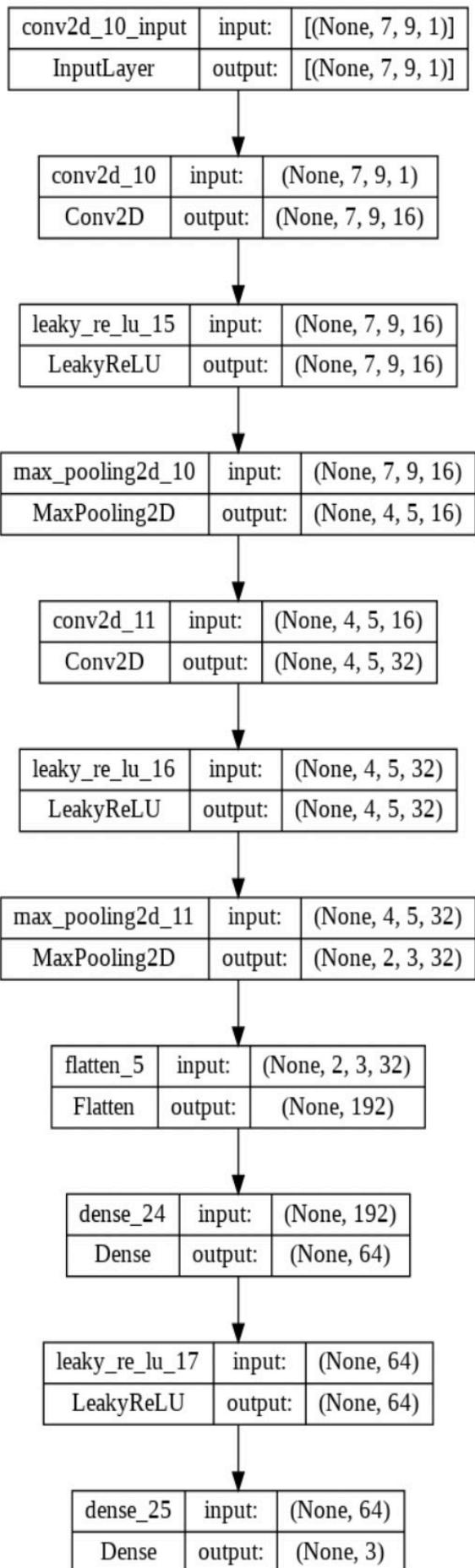


Figure 11: L'architecture du réseau de neurones convolutif

### 3.3 Résultats de la calibration avec le réseaux de neurones artificiels

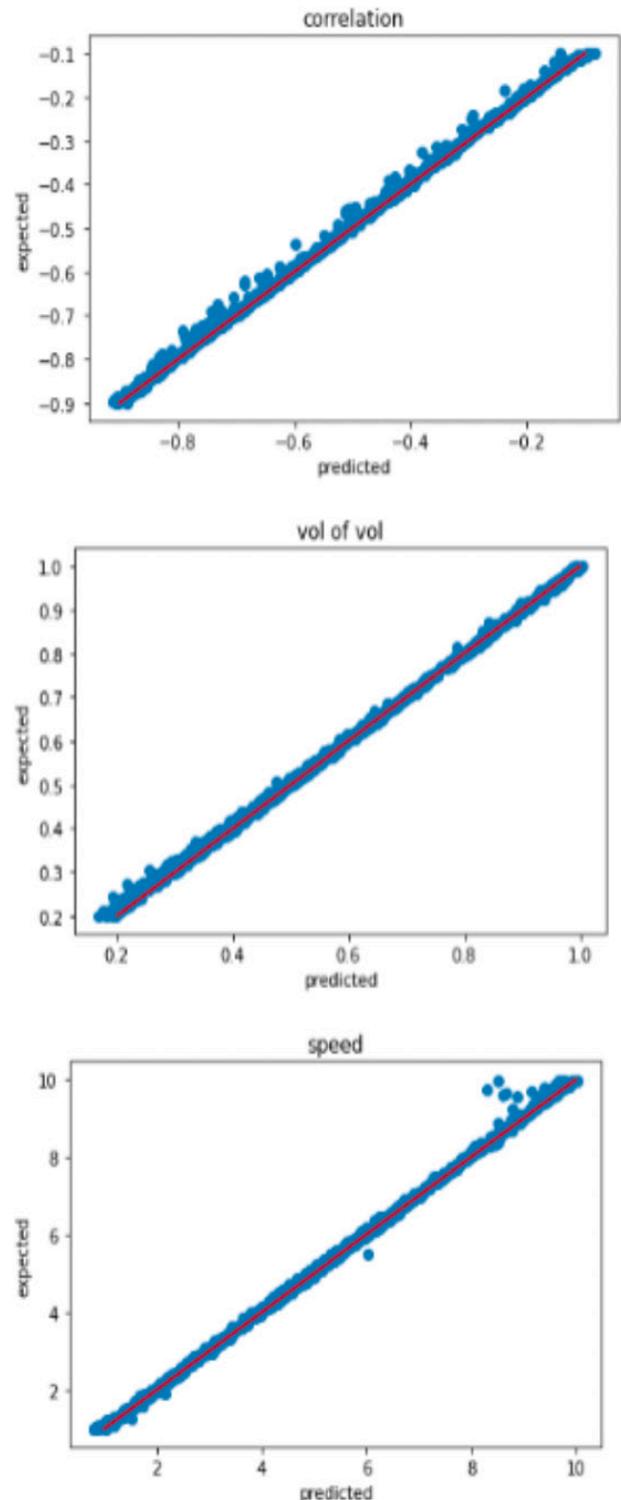


Figure 12: Les valeurs espérée en fonction des valeurs prédites par le réseau de neurones Feed Forward pour un apprentissage sur des volatilités implicites (test data)

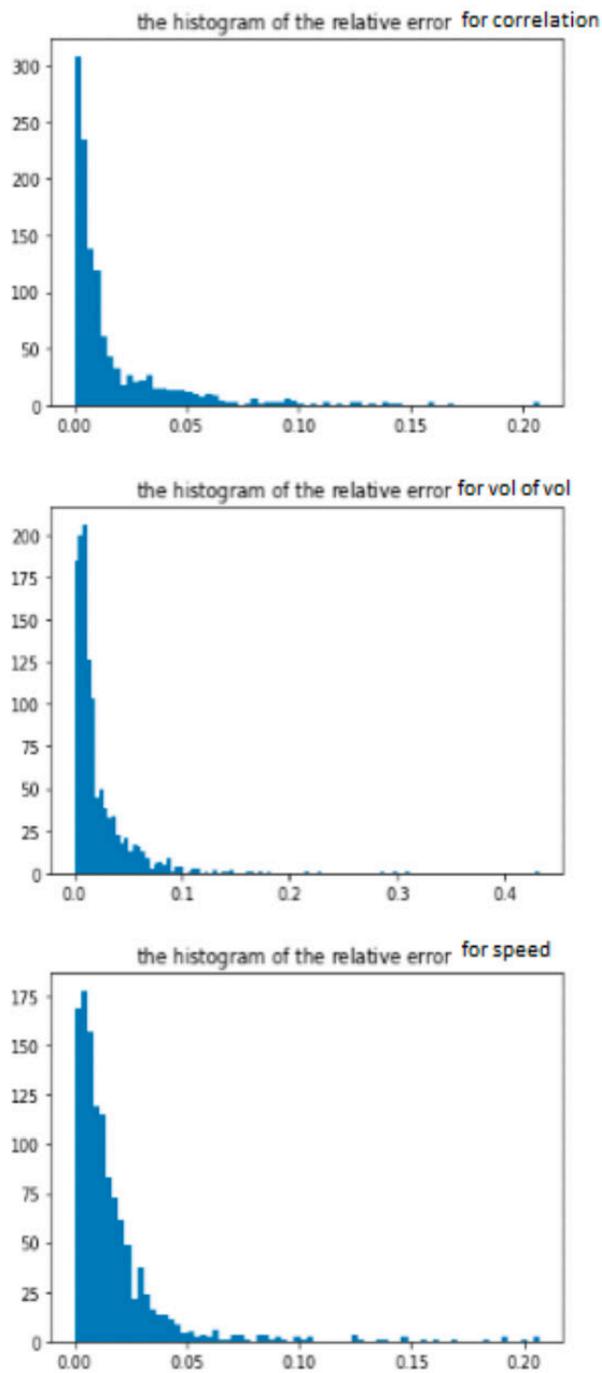


Figure 13: Les histogrammes des erreurs relatives pour la calibration avec un réseau de neurones Feed Forward avec des volatilités implicites (test data)

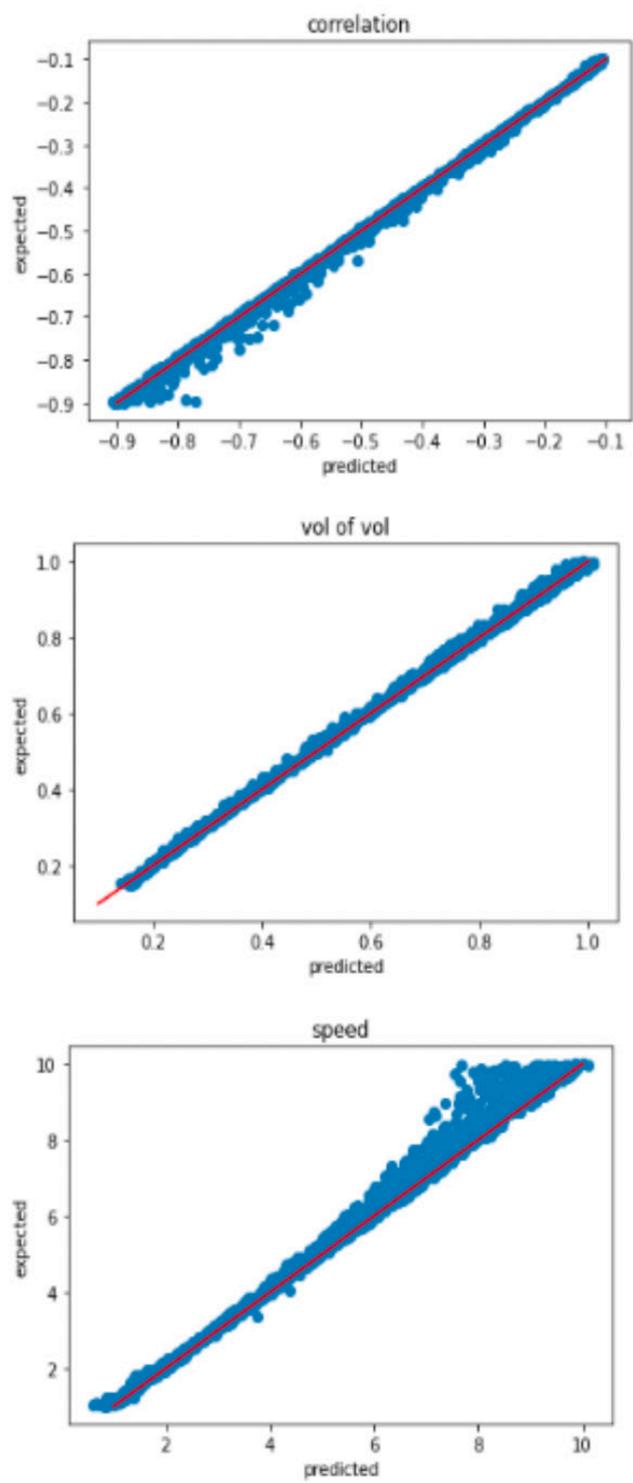


Figure 14: Les valeurs espérées en fonction des valeurs prédites par le réseau de neurones convolutif pour un apprentissage sur des prix (test data)

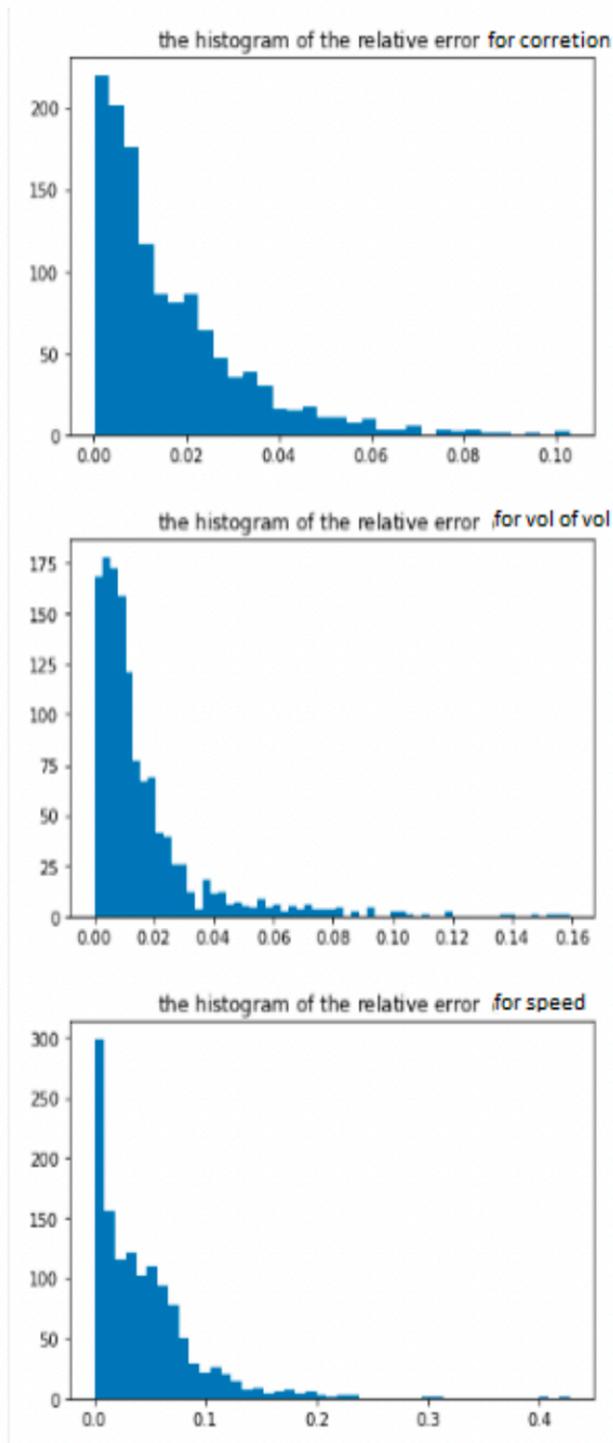


Figure 15: Les histogrammes des erreurs relatives pour la calibration avec un réseau de neurones convolutif avec des volatilités implicites (test data)

Les figures précédentes montrent les erreurs entre les paramètres prédits du modèle de Heston et les paramètres espérés pour le réseau de neurones Feed Forward et pour le réseau de neurones convolutif.

Le fait que la performance des réseaux de neurones est bonne à la fois pour les données d'entraînement et de test

implique que les réseaux de neurones sont capables d'obtenir de bonnes performances lors de la prédiction de données non vues. On note que seules les prédictions pour  $\rho$  sont légèrement plus mauvaises que celles des autres paramètres du modèle.

Comme on peut le voir sur les figures, les erreurs relatives obtenues par le réseau de neurones Feed Forward sont sensiblement plus faibles. En effet, le réseau de neurones convolutif applique des filtres et un maxpooling aux données d'apprentissage, ce qui, dans une certaine mesure, entraîne une perte d'informations. De plus, le but des réseaux de neurones convolutifs est souvent de simplement étiqueter l'image, alors que pour calibrer le modèle de Heston, les valeurs exactes des sorties des paramètres sont requises.

## Conclusion

Les modèles stochastiques complexes rendent nécessaire l'utilisation de méthodes numériques efficaces pour la calibration. Dans cette note, nous avons étudié l'exemple du modèle de Heston. Nous avons présenté et implémenté les méthodes classiques pour sa calibration. Cela nous a permis d'avoir une référence pour le temps d'exécution. Ensuite, nous avons testé des réseaux de neurones artificiels pour la calibration.

Nous avons constaté que la calibration avec les réseaux de neurones est largement plus rapide par rapport aux méthodes classiques (présentées dans la section 2) mais, elle présente deux grandes limites : La première est le coût de la génération de données synthétiques pour des prix d'options dont le sous-jacent est modélisé par le modèle de Heston. La deuxième est que nous n'avons pas trouvé de moyen d'améliorer la précision des nos prédictions notamment pour la corrélation.

## References

- [1] M. Black, F. & Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 1973.
- [2] S. Mikhailov and Ulrich Nogel. Heston's stochastic volatility model implementation, calibration and some extensions. July 2003.
- [3] Reyners S De Spiegel J, Madan DB. Machine learning for quantitative finance. 2018.
- [4] Leif B.G. Andersen and Vladimir V. Piterbarg. Interest rate modeling.
- [5] M. A. Nielsen. Neural networks and deep learning. 2015.
- [6] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. 2014.
- [7] Damiano Brigo and Haitz Sáez de Ocáriz Borde. Interpretability in deep learning for finance: a case study for the heston model. April 2021.

## A propos d'Awalee

Cabinet de conseil indépendant spécialiste du secteur de la Finance.

Nous sommes nés en 2009 en pleine crise financière. Cette période complexe nous a conduits à une conclusion simple : face aux exigences accrues et à la nécessité de faire preuve de souplesse, nous nous devons d'aider nos clients à se concentrer sur l'essentiel, à savoir leur performance.

Pour accomplir cette mission, nous nous appuyons sur trois ingrédients : habileté technique, savoir-faire fonctionnel et innovation.

Ceci au service d'une ambition : dompter la complexité pour simplifier la vie de nos clients.

«Run the bank» avec Awalee !



## Contactez-nous

Ronald LOMAS  
Partner  
rlomas@awaleeconsulting.com  
06 62 49 05 97