

Deep Learning in Finance

TP1 : Hurst

22 octobre 2024

Buts

- Estimer l'exposant et la volatilité de Hurst de mouvement brownien fractionnaire
- Utiliser des CNNs
- Réchelonner des données
- (early stopping)
- (hyperoptimization)
- (CNN2d)

Préparation des données

1. Pour 10 valeurs de $H \in [H_{min}, H_{max}] \subseteq]0, 1[$ également espacées, générer $M = 10000$ séries temporelles de fBm de longueur $T = 100$ avec la librairie `fbm` comme valeurs d'entraînement X_{train} .
2. Générer 1000 séries temporelles de longueur $T = 100$ pour 100 valeurs également espacées de $H \in [0.01, 0.99]$ comme valeurs de test X_{test} .
3. Réchelonnez **les lignes** de X_{train} et X_{test} afin qu'elles aient chacune une moyenne de 0 et une déviation standard de 1. Utilisez `StandardScaler` de `sklearn`.
4. Il est conseillé de sauver X_{train} , Y_{train} et X_{test} , Y_{test} si leur génération prend beaucoup de temps (`joblib.dump()`, `joblib.load()`, ou format `parquet`)

1 Réseaux denses

1. En utilisant une architecture dense, définir et entraîner un réseau profond dense. Combien de paramètres contient-il ? (utiliser la fonction `summary`)
2. Tracer le biais moyen du DNN en fonction de H (groupby de Pandas)
3. Tracer l'erreur absolue moyenne du DNN en fonction de H .

2 Réseaux convolutifs unidimensionnels

1. Implémenter l'architecture proposée par H. Stone, QF (2020) [lien]. Combien de paramètres contient-il ? (utiliser la fonction `summary`)
2. Entraîner ce réseau.
3. Tracer le biais moyen du CNN en fonction de H
4. Tracer l'erreur absolue moyenne du CNN en fonction de H .
5. Comparer le biais et l'erreur avec le résultat des DNN (sur les mêmes graphiques).
6. Combiner les prédictions des deux réseaux afin de diminuer leur biais et l'erreur.
7. Comment rendre l'estimation de H par ANN plus robuste par rapport aux valeurs aberrantes ? (Gedankenexperiment)

3 (Optionel) Hyperoptimisation

Comprendre comment utiliser Talos [lien].

1. Utiliser une fonction pour construire un modèle et calibrer le modèle.
2. Ajouter la partie paramétrique dans cette fonction : choisir le / les paramètres à optimiser
3. Choisir la méthode d'optimisation des hyperparamètres.
4. Faire tourner l'hyperoptimisation.

4 (Optionel) Réseaux convolutifs bidimensionnels

1. Adapter l'architecture proposée par H. Stone, QF (2020) à des réseaux convolutifs dont les entrées sont les images des séries temporelles. Attention à la taille des images. Combien de paramètres contient-il ?
2. Entraîner ce réseau. Utiliser `np.expand_dims(X_train, axis=2)` pour que la matrice `X_train` soit acceptée par les CNNs.
3. Tracer le biais moyen du CNN en fonction de H
4. Tracer l'erreur absolue moyenne du CNN en fonction de H .
5. Comment rendre ce genre d'estimateur plus robuste par rapport aux valeurs aberrantes ?
6. Comparer le biais et l'erreur avec le résultat des réseaux précédents.
7. Comment combiner les prédictions de ces trois réseaux ?