Stratégie de trading Forex

Baptiste COUDERC

Michel-Victor LEVEQUE

Pierrick ROUSSEAU

Nous avons décidé d’entraîner un réseau de neurone qui sera ensuite utilisé pour générer des ordres d’achats ou de ventes.

# Les caractéristiques utilisées

Nous calculons d’abord les différentes features pour notre réseau de neurones.

Afin d’accélérer l’utilisation du réseau de neurones, nous préparons en amont un fichier par paire de devise, contenant les caractéristiques que nous souhaitons utilisées dans le réseau. Les huit fichiers ainsi créés sont générés dans une procédure à part.

Cette procédure ressort donc pour chaque paire de devise ces paramètres, pour chaque timestamp :

* La somme des volumes échangés
* Le prix mid
* La volatilité des prix Buy et la volatilité des prix Sell
* Les ratios volume total par prix Mid, volume Buy par prix Buy et volume Sell par prix Sell
* La moyenne mobile sur des fenêtres de 10 du ratio volume total par prix Mid
* Les bandes supérieures et inférieures de Bollinger du prix Mid pour 2 écart-types
* Les retracements de Fibonacci sur le prix Mid

Les données initialement fournies dans la base de données sont donc remplacées par les champs ci-dessus.

Sauvegarder ces caractéristiques dans des fichiers à part permet d’enregistrer chaque calcul et ainsi, de ne pas les faire deux fois dans la même journée.

# Le réseau de neurones

Ensuite dans le fichier neural\_network\_training nous produisons un fichier sav qui contient le réseau pour chaque devise. Nous avons optimisé le réseau grâce à une optimisation des hyper-paramètres et à un choix de features judicieux. Nous avons optimisé les paramètres suivants :

- Le nombre de couches de neurones et de neurones par couches.

- La fonction d’activation, utilisée pour déterminer le résultat d’un neurone.

- L’alpha, qui est le changement maximal de nos poids lors de l’apprentissage.

- Le solveur pour déterminer les poids.

Le réseau est ensuite chargé dans le fichier main pour produire des ordres d’achats ou de ventes. Nous gardons les positions une période et nous calculons ensuite le calmar ratio.

# Les résultats

Calmar ratio

# Les défis rencontrés

1. La gestion de la mémoire

Les bases de données reçues, sous format CSV, étant très denses et nos ordinateurs ayant une capacité limitée, il nous a fallu trouver une solution afin de lire et d’enrichir les données. Nous utilisons ainsi la fonction *reduce\_mem\_usage* qui, en jouant avec le format de chaque colonne de notre base, a permis d’alléger l’utilisation de la mémoire d’un tiers.

De plus, nous faisons le choix de n’utiliser le réseau de neurones que pour une paire de devise à la fois, afin de ne pas multiplier les données en cours d’utilisation en même temps.

# Résultats

Nous effectuons un bénéfice de 0.08%. Cependant comme on génère un ordre d’achat ou de vente à chaque période, nous faisons une large perte après avoir pris en compte les coûts de transactions.