

Trading Algorithmique



Finance GR02
Franck Zhang
Seung-hwan Yang

Sommaire

- 1) Introduction**
- 2) Infrastructures**
- 3) Stratégies**
- 4) Deep Learning**
- 5) Données**
- 6) Avantages**

1. Introduction

Qu'est-ce que le Trading algorithmique ?

Le trading algorithmique, aussi appelé trading automatisé ou trading automatique, boîte noire de négociation (en anglais : black-box trading), effectué par des robots de trading, est une forme de trading avec utilisation de plates-formes électroniques pour la saisie des ordres de bourse en laissant un algorithme décider des différents aspects de l'ordre, tel que l'instant d'ouverture ou de clôture (le timing), le prix ou le volume de l'ordre et ceci, dans de nombreux cas, sans la moindre intervention humaine.

Il s'agit d'une stratégie d'investissement qui se base sur les modèles économiques et des logiciels informatiques en mesure de prendre position sur un marché. Avec la dématérialisation des ordres de trading et la puissance de calcul de plus en plus importante des ordinateurs, ce type de trading connaît un véritable essor.

Qu'est-ce que QuantConnect ?

QuantConnect, est une plateforme de trading algorithmique OpenSource, gérée par la communauté. Leur moteur de trading est alimenté par LEAN qui une technologie multi-plateforme et multi-actifs. Il permet principalement de tester des algorithmes de trading et de les tester en effectuant des backtest sur les périodes passées d'actifs.

Comment allons nous implémenter le bot ?

Nous pouvons soit l'implémenter sur la plateforme QuantConnect qui nous propose un environnement pré-configuré pour tester notre algorithme et effectuer des backtests ou alors l'implémenter en local, pour cela il faudra télécharger le projet LEAN Open Source en local et relier notre moteur à une plateformes d'échange (Binance, Bitmex...) via l'API.

2. Infrastructure

L'infrastructure est la partie qui permet au bot d'appliquer la stratégie c.a.d du traitement de données (Supports, Résistances, Moyennes mobiles...) aux passages des ordres (Achats/Ventes). L'infrastructure sera composée de deux parties :

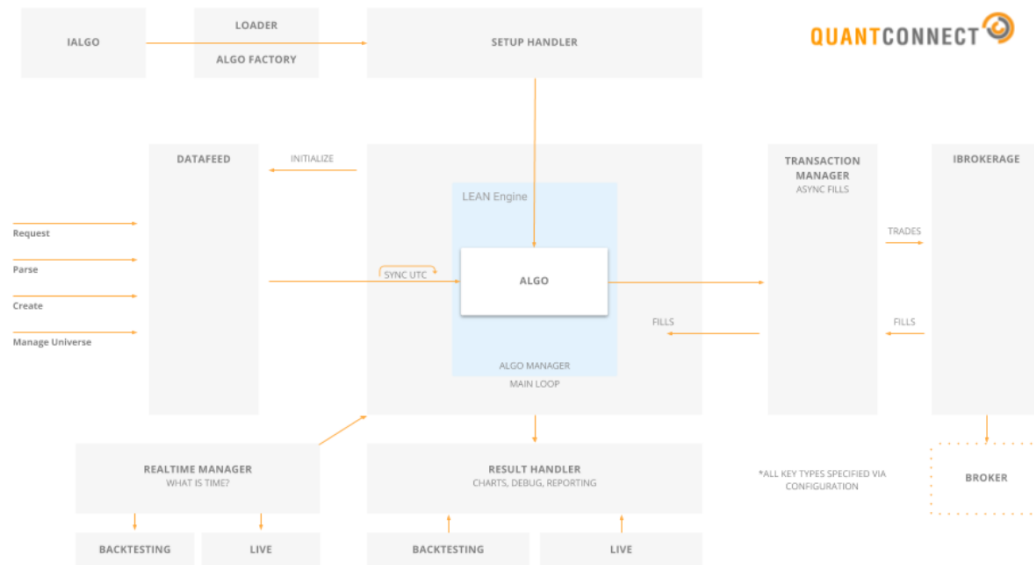
a) La connectivité

Au sein du trading algorithmique, le bot doit permettre à l'utilisateur d'interagir avec un ensemble diversifié d'échanges de devises cryptographiques.

Au sein de ce projet, pour la connectivité du bot, nous allons faire appel à la bibliothèque XChange. XChange est une bibliothèque Java fournissant une API simple et cohérente pour interagir avec plus de 60 Exchange, fournissant une infrastructure cohérente pour le trading et l'accès aux données du marché.

b) Le moteur (librairie lean)

Pour le moteur qui va exécuter l'algorithme, nous allons utiliser le projet LEAN de QuantConnect. Lean est un moteur qui permet de gérer le portefeuille et les flux de données, permettant ainsi aux utilisateurs de se concentrer uniquement sur la partie stratégie et l'exécution de l'algorithme. Les données sont acheminées dans la stratégie via des gestionnaires d'événements, sur lesquels nous pouvons placer des transactions (achats/ventes).



Le moteur est divisé en plusieurs pièces modulaires qui peuvent être étendues sans toucher aux autres fichiers. Les modules sont configurés dans config.json en tant qu'ensembles "environnements". Grâce à ces environnements, vous pouvez contrôler LEAN pour qu'il fonctionne dans le mode requis.

Traitement des résultats (IResultHandler)

Traite tous les messages provenant du moteur de trading algorithmique. Décide de ce qui doit être envoyé et où les messages doivent aller. Le système de traitement des résultats peut envoyer des messages à une interface graphique locale ou à l'interface Web.

Source d'alimentation en données (IDataFeed)

Connectez et téléchargez les données requises pour le moteur de trading algorithmique. Pour le backtesting, cette fonction extrait les fichiers du disque, pour le trading en direct, elle se connecte à un flux et génère les objets de données.

Traitement des transactions (ITransactionHandler)

Traite les demandes de nouveaux ordres ; soit en utilisant les modèles de remplissage fournis par l'algorithme, soit avec un courtage réel. Renvoie les ordres traités au portefeuille de l'algorithme pour qu'ils soient exécutés.

Gestion des événements en temps réel (IRealtimeHandler)

Générez des événements en temps réel - tels que les événements de fin de journée. Déclenchez des rappels vers les gestionnaires d'événements en temps réel. Pour le backtesting, ceci est simulé et fonctionne en temps simulé.

Configuration de l'état de l'algorithme (ISetupHandler)

Configure l'algorithme cash, le portefeuille et les données demandées. Initialise tous les paramètres d'état requis.

3. Stratégies

Les stratégies de trading sont diverses et variées, passant de l'analyse technique au fondamental et puis à l'apprentissage automatisé.

Nous allons dans cette partie tester (backtest) les différents algorithmes disponible sur le git sur la plateforme QuantConnect :

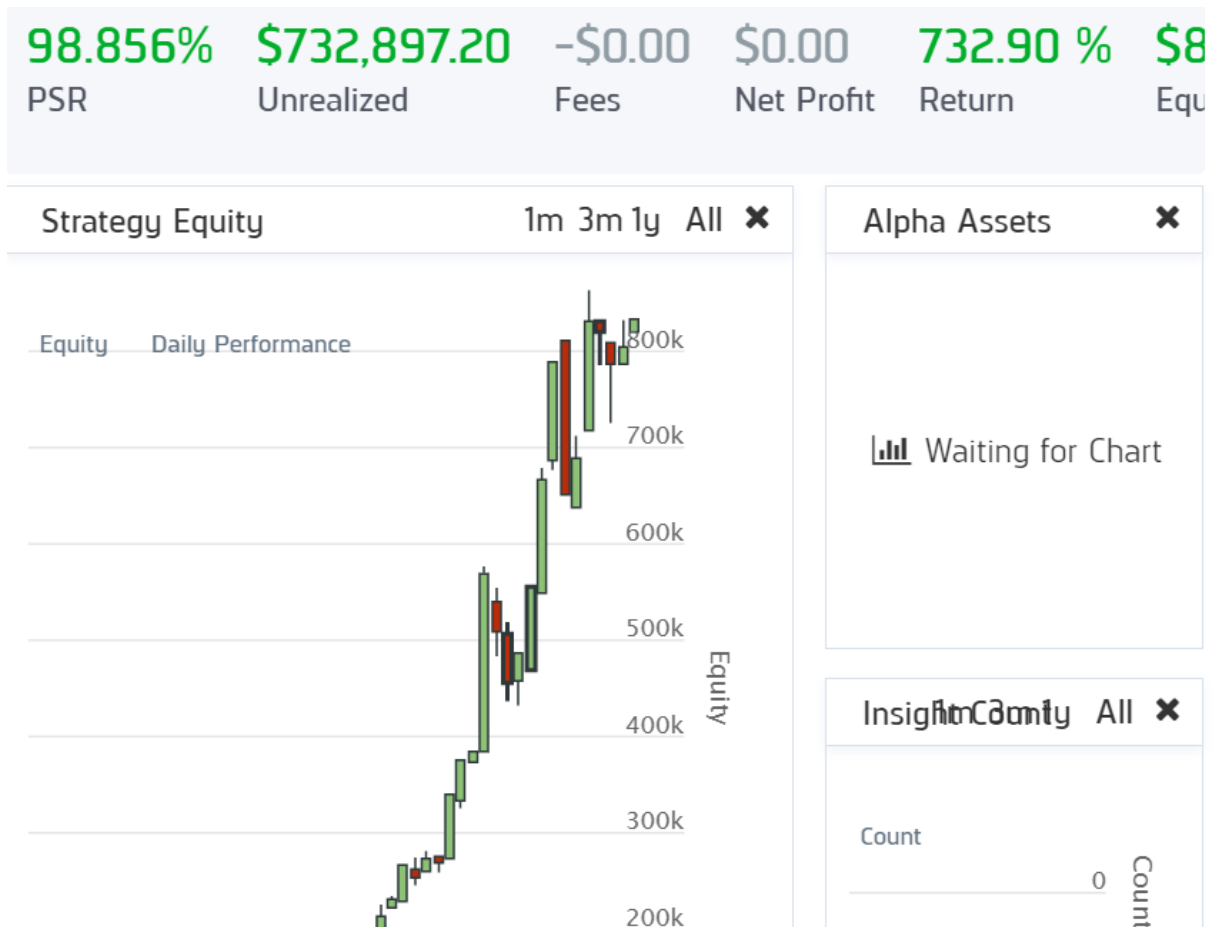
<https://github.com/QuantConnect/Lean/tree/master/Algorithm.CSharp>

Période de Backtest : 06/04/2020 --> 06/04/2021

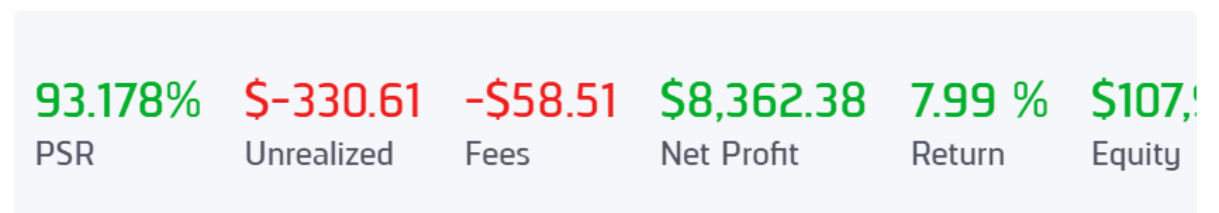
Asset : BITCOIN

Capital de départ : 100 000\$

- 1er Algorithme: CustomDataBitcoinAlgorithm



- 2ème Algorithme: HistoryAlgorithm



4. Deep learning

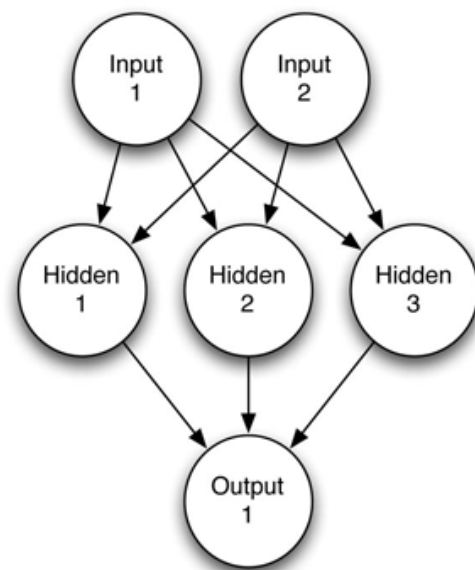
Les analystes techniques tentent de prévoir l'orientation future de la sécurité en analysant les données passées du marché. Le Deep learning pour la prévision du marché consiste à représenter les données du marché d'une

manière qui capture vraiment l'essence des modèles sous-jacents pour que le Deep Learning puisse les reconnaître.

Pourquoi alors le Deep Learning est-il nécessaire pour un bot de trading algorithmique ?

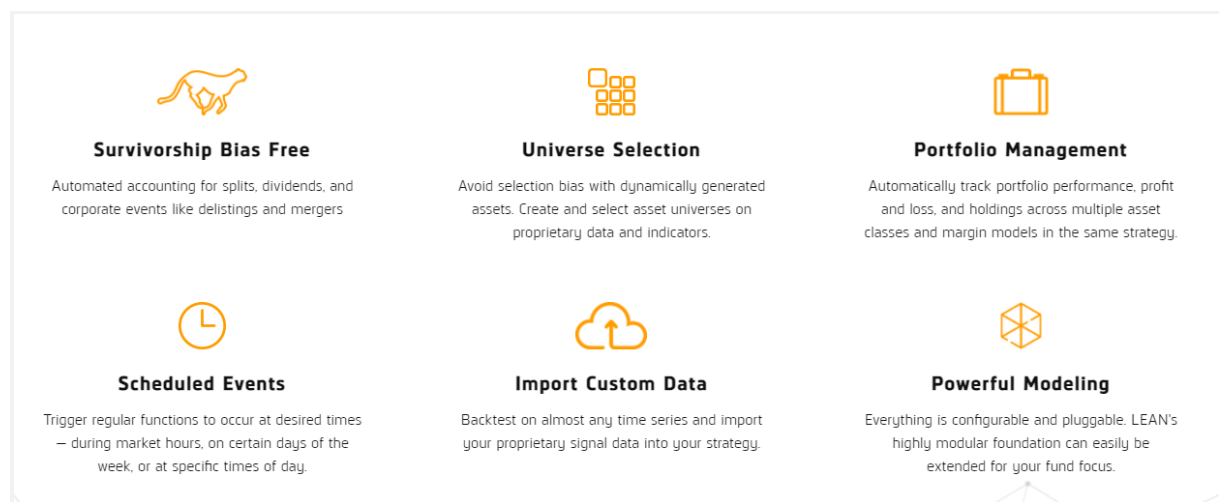
En effet, se basant sur les données passées du marché, le bot va alors essayer de prévoir les hauts et les bas du marché en question.

Prenons un input qui servira de point de départ du réseau de neurones. Le réseau est constitué de connexions entre les neurones. Le processus d'apprentissage, soit le Deep Learning, va permettre d'ajuster les connexions. Après toutes ces démarches, notre bot sera prêt à fonder un réseau de neurones, pour ensuite donner un output unique. Le réseau neuronal sera alors prêt à prédire.



A partir de cela, le bot pourra exécuter des ordres d'achat et de vente.

5. Les avantages du trading algorithmique



6. Conclusion

Pour notre bot de trading algorithmique, nous avons décidé de redéfinir les différents aspects qui peuvent être abordés lors de sa conception. L'infrastructure, les différentes stratégies, ainsi que d'une capacité de Deep Learning munie d'une base de données. Tout en se basant sur ces aspects, nous avons conçu notre bot via la plateforme QuantConnect. Celui-ci peut accepter différentes stratégies sur une base de données assez riche, tout en gardant une infrastructure correcte à l'œil de l'utilisateur.

Pour des améliorations possibles vis-à-vis de notre projet, il serait intéressant de concevoir un bot sans passer par une plateforme pré-configurée.

