Rapport de stage de fin d’études réalisé chez

Delubac Asset Management

Auteur

Victor Bellissant

A l’attention de

Vlad Bally et Damien Lamberton





Master MFD

Université Gustave Eiffel et ENPC Paristech

2021 - 2022

Delubac Asset Management :

Cette société de gestion d'actifs à été créée en 2000, au sein de cette structure il existe une dizaine de fonds investis.

Il y a entre autres un fond long Actions Européennes et un fond obligataire.

J’ai réalisé mon stage au sein d’une équipe de deux personnes, Olivier Berthier et Mathias Choussy. Olivier Berthier est responsable de l’allocation sur l’ensemble des fonds de la structure, tandis que Mathias Choussy est directeur de la dette privée.

# Sujet du stage

Mathias et Olivier ont pour projet de créer une structure d’investissement diversifiée basée sur une stratégie "Global-Macro", l’idée est de caractériser l’état de l’économie actuel et futur en terme de variation de la croissance et de l’inflation; cette caractérisation donne lieu à quatre scénarios différents et sous chacun de ces scénarios est investi une composition de quatre portefeuilles. Chacuns des quatres portefeuille est caractéristique d’un scénario économique, l’objectif est de trouver les portefeuilles qui performent au mieux dans chacuns des régimes tout en gardant une certaine diversification au niveau de leurs compositions, un autre objectif est de trouver la bonne pondération pour chacuns d’entre-eux car ils restent tous les quatres constamment investis. Mon stage s’est donc articulé autour des travaux suivants :

* Mise en place d’un système de réseau de neurones (LSTM) afin de forecaster assez précisément la croissance (GDP) et l’inflation (CPI) des Etats-Unis et de la zone EURO.
* Construction d’un module de Backtest automatique permettant de construire des portefeuilles diversifié et caractéristique de chacuns des quadrants puis d’investir de façon optimale chacun des quatres portefeuilles par rapport au quadrant actuel. Il s’agit d’optimisation de portefeuille basée sur la théorie moderne du portefeuille (Markowitz), d’un calcul des bornes d’allocations dans le but d’obtenir une homogénéitée du risque potentiel de chacuns des actifs, de trouver la meilleur "temporalité d’investissement" puis de valoriser l’ensemble de la stratégie (4 portefeuilles) sur des données historiques (15 ans).
* Optimisation des paramètres de Backtest, notamment au niveau du choix de la période sur laquelle le module calcul les poids de chacuns des actifs, obtention aussi de la meilleure période de rebalancement. J’ai aussi écrit des fonctions capables de transformer les séries de prix par des filtres (MESA/KAMA), ces séries transformées servent ensuite en Input au niveau du module d’optimisation. Au final, l’idée est de trouver le signal qui permet de rendre le système d’allocation le plus performant possible.

# Caractérisation Macro-Économique et réseau de neurones

Je présente ici le procédé utilisé pour caractériser l’économie et les outils de Machine Learning que j’ai mis en place afin de pouvoir anticiper les données officielles.

Les données que je dois forecaster pour cette étude sont le GDP YoY pct change Quaterly (Gross Domestic Product) et le CPI YoY pct change Quaterly (Consumer Price Index) depuis 1960. YoY pct change Quaterly signifie que la donnée obtenue exprime le changement en pourcentage par rapport à l’année précédente de la grandeur mesurée et cela chaque trimestre.

J’ai obtenu ces données via des requêtes sur l’application Bloomberg, puis en quelques lignes de Python j’ai construit un Dataset contenant ces deux données (X,Y).

J’y ai ensuite appliqué la transformation suivante :

Pour chaques valeurs du Dataset, la valeur est remplacée par la différence entre la valeur précédente et elle-même, cela revient à regarder la variation de la croissance et de l’inflation d’un trimestre à l’autre. Mathématiquement parlant ça donne :

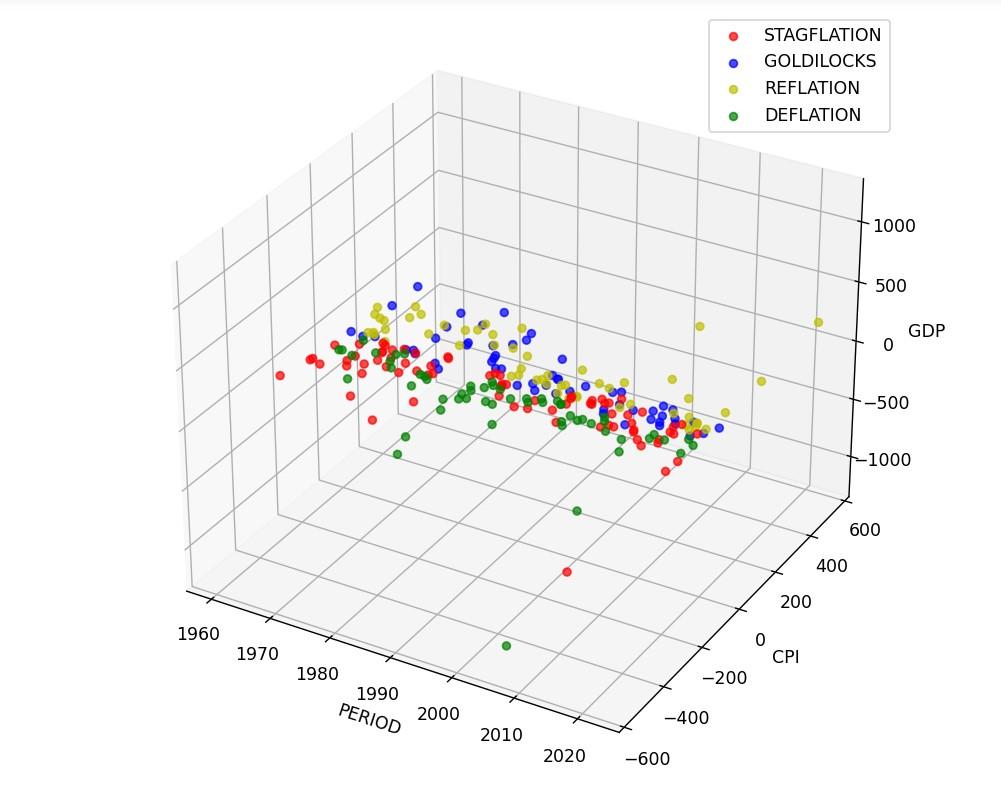
∀*i* ∈ [1*,N*]*,*(*X,Y* )*i* = [(*X,Y* )*i* − (*X,Y* )*i*−1] ∗ 100 (1)

Une fois cette transformation réalisée, j'ai tracé sur un repère orthonormé les couples de valeurs obtenues, la dynamique de l’inflation en abscisse et en ordonnée la dynamique de la croissance. Le repère est donc coupé en quatre parties, chacune de ces portions est caractéristique d’un scénario économique.

Voici les quatres scénarios économique :

* REFLATION : Accélération de la croissance et de l’inflation.
* GOLDILOCKS : Accélération de la croissance et Ralentissement de l’inflation.
* DEFLATION : Ralentissement de la croissance et de l’inflation.
* STAGFLATION : Ralentissement de la croissance et Accélération de l’inflation.

Voici le graphe caractérisant les quatres Quadrants depuis 1960 obtenue en Python :



Je détaille ici en quelques lignes les modules Python utilisés afin d’obtenir le graphique.

J’ai utilisé les modules Python suivant :

* Pandas pour la lecture de données.
* Matplotlib pour l’affichage des données.

Le graphique précédent à été obtenu avec les données officielles du site de la FRED (Federal Reserve Bank of Saint Louis), une fois que j’ai compris qu’un tel graphique était réalisable j’ai mis en place un système de réseau de neurones afin de prévoir le prochain quadrant (trimestre).

Afin de pouvoir prévoir ces deux données, j’ai mis en place un réseau de neurone LSTM capable de faire la chose suivante : à partir de quelques données Macro économique fortement liée (connexe) à la donnée d'intérêt (principale), le réseaux apprends à donner son avis sur le prochain résultat de cette donnée (principale) en se basant sur les valeurs des données connexes et principale.

Avec les conseils de Mathias Choussy, j’ai conduit quelques recherches afin de déterminer une liste de données qui serait susceptible d’avoir un pouvoir prédictif sur le GDP et le CPI.

Voici la liste des données que j’ai récupéré :

Variable connexe pour le GDP (GDP CYOY Index) :

1. : RSTAXAGM Index : Retail Sales Less Food Services Auto Dealers Build Mat Gas Station SA MoM, il s’agit de la variation de la valeur des bien neufs ou d’occasion vendus aux Etats Unis, c’est une mesure du prix des échanges autres que la nourriture, les voitures, les produits de constructions et le gazole.
2. : CONCUN35 Index : Conference Board Consumer Confidence SA 1985=100 Age of Household Head Under 35, il s’agit d’une mesure de la confiance des consommateurs de moins de 35 ans aux Etats-Unis, tranche de citoyen la plus motrice de l’économie générale.
3. : EMPRNEMP Index : US Empire State Manufacturing Survey Number of Employees SA, est une estimation du nombre d'employés dans le secteur de la manufacture, secteur d'activité très exposé aux variations de la croissance.
4. : USURTOT Index : About U-3 US Unemployment Rate Total in Labor Force Seasonally Adjusted, il s’agit d’une mesure du taux de non emploi parmi la force totale de travailleurs aux USA (employé + non employé).
5. : IP CHNG Index : Industrial Production Quote, mesure la variation de la production dans le secteur des transports, l’industrie minière, de la manufacture et des services publiques aux Etats-Unis.
6. : NAPMPMI Index : ISM Manufacturing PMI SA, mesure l’état de l'activité de la manufacture et de l’économie en général.
7. : MPMIUSCA Index : US Composite PMI SA, mesure composite des vues des dirigeants dans le secteur de la manufacture, des services et de la construction à propos de leurs projets de développement.

Variable connexe pour le CPI (CPI YOY Index) :

1. : CPI XCHG Index : US CPI Urban Consumers MoM SA, mesure le prix de la consommation urbaine aux Etats-Unis sur un panier de produits.
2. : CPI XYOY Index : US CPI Urban Consumers Less Food Energy YoY NSA, mesure le prix de la consommation urbaine en excluant la nourriture et l'énergie.
3. : CPI INDX Index, la mesure du prix de la consommation générale, données totalement brute, il ne s’agit pas d’une variation en pourcentage.
4. : PCE CORE Index : Personal Consumption Expenditure Index, mesure la valeur des dépenses personnelles aux Etats-Unis.
5. : CPIQNCCS Index : US CPI New Cars SA, mesure du prix des voitures aux Etats-Unis.
6. : PPI YOY Index : US Producer Price Index YoY, mesure la façon par laquelle les producteurs de biens quel qu’ils soient, sont rémunérés pour leurs productions, il s’agit d’une mesure globale de l’inflation.

1.1 Le réseau de neurones :

Le réseaux de neurone que j’ai utilisé est un réseau LSTM (Long-Short Term Memory), cette architecture à pour avantage d'être dotée d’une "mémoire", en effet les configurations passée sont stockés afin d’être ré-utilisable au moment voulu. Cette architecture étant complexe, je vais tenter d’être le plus clair possible.

* + 1. Motivation du problème :

L’objectif est de construire un outils de Machine Learning étant capable de prédire convenablement les chiffres du GDP et du CPI, ces données étant des séries temporelles, il est judicieux d’utiliser un modèle qui est capable de retenir l’information contenue dans ces séries et d'être en mesure d’utiliser cette information "Au bon moment".

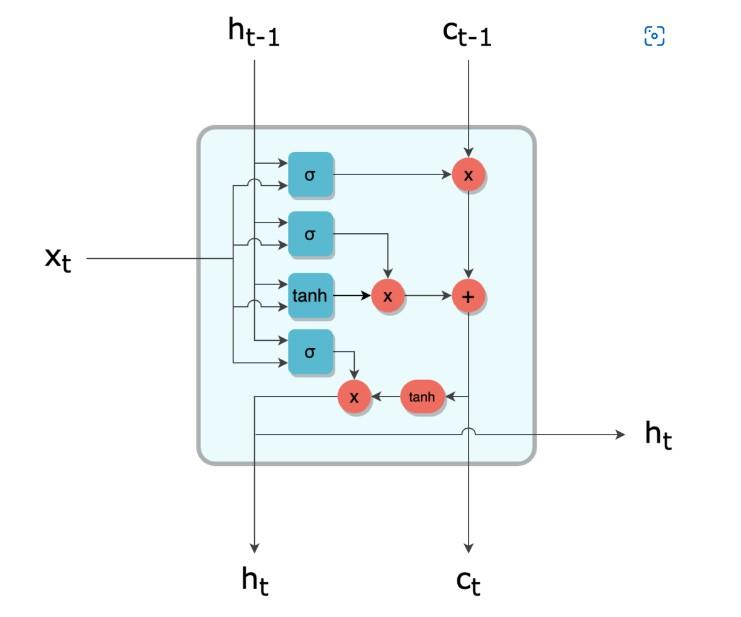
"Au bon moment" signifie ici être capable de se servir de l’apprentissage mené il y a un certain temps afin de prédire la prochaine valeure et c’est justement ce que permettent les architectures de type LSTM.

* + 1. Le fonctionnement d’un réseaux LSTM :

L’idée derrière ce choix d’architecture de réseaux de neurones est de diviser le signal entre ce qui est important à court terme à travers le Hidden State, et ce qui l’est à long terme, à travers le Cell State, ces deux States seront précisés plus bas. Ainsi, le fonctionnement global d’un LSTM peut se résumer en 3 étapes :

* Détecter les informations pertinentes venant du passé, piochées dans le Cell State à travers la Forget gate
* Choisir, à partir de l’entrée courante, les informations qui seront pertinentes à long terme, via l’Input gate. Celles-ci seront ajoutées au Cell State qui fait office de mémoire longue
* Piocher dans le nouveau Cell State les informations importantes à court terme pour générer le Hidden State suivant à travers l’Output Gate.

On peut représenter une cellule LSTM de la façon suivante :



L’information transite d’une cellule à la suivante par deux canaux, h et c. À l’instant t, ces deux canaux se mettent à jour par l’interaction entre leurs valeurs précédentes *ht*−1 et *ct*−1 ainsi que l’élément courant de la séquence *xt*.

Supposons que l’algorithme est en cours d'exécution et que nous sommes à l’instant t, c'est-à dire que l’algorithme regarde maintenant la t ième valeur x*t.*

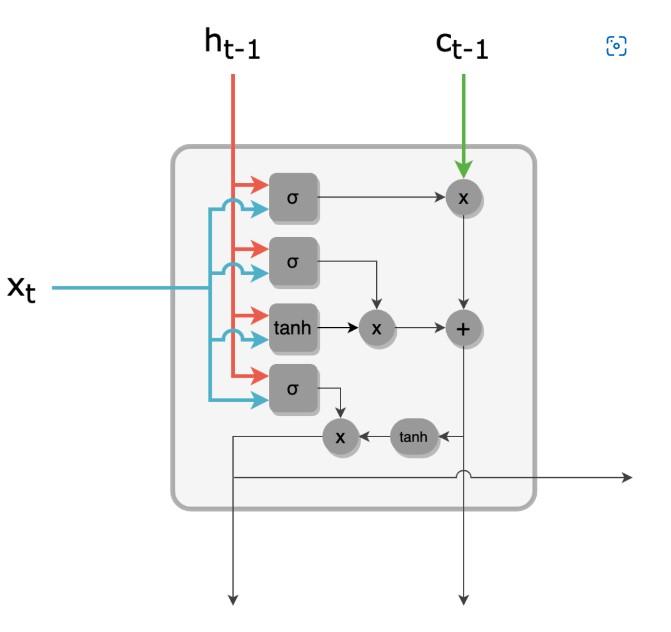
Alors la tième cellule LSTM prend 3 vecteurs en entrée :

L’élément courant de la séquence *xt*.

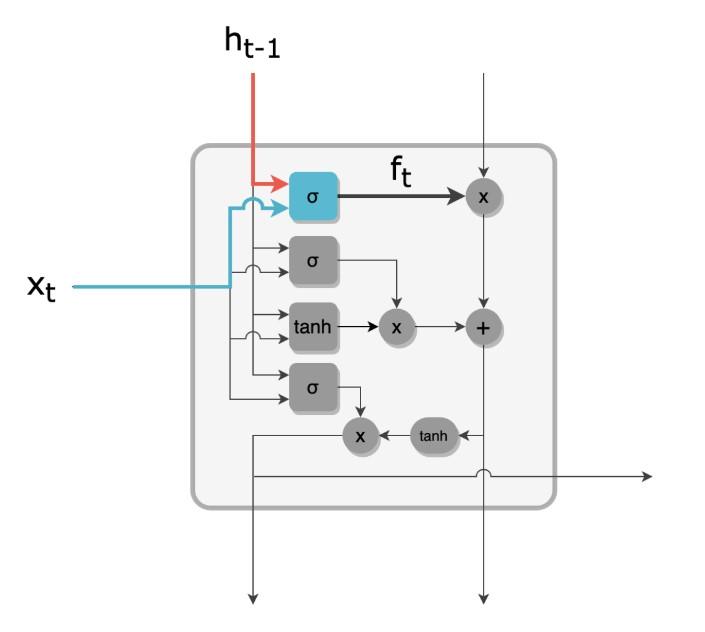
Le Hidden State de la cellule précédente *ht*−1.

Le Cell State de la cellule précédente *ct*−1.

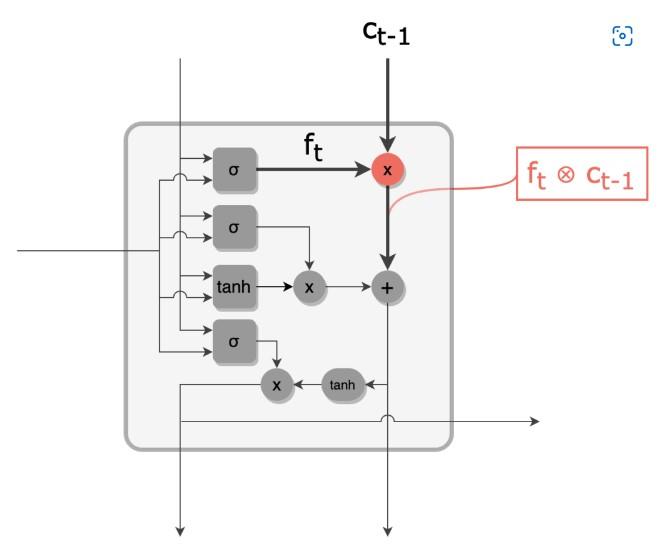
C’est ce dernier vecteur que nous allons suivre particulièrement. Il s’agit d’une route privilégiée de transmission d’information sur la séquence.



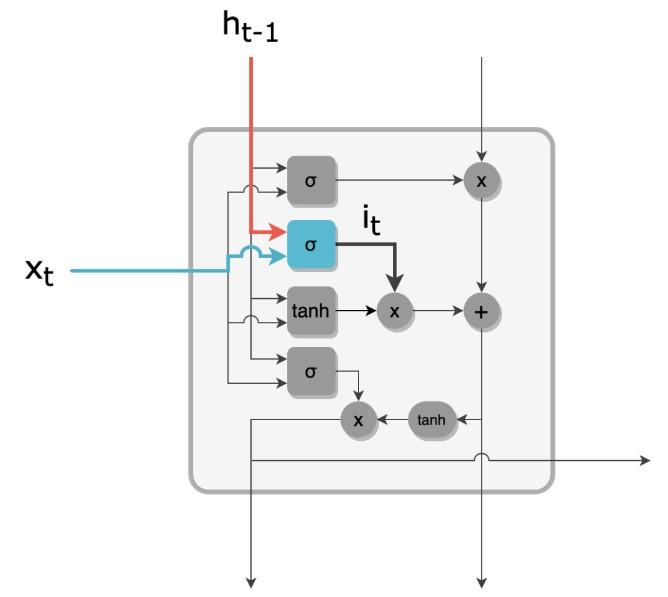
La Forget gate est une couche dense avec une activation sigmoïde. À partir de *ht*−1 et *xt*, cette Forget gate produit un vecteur *ft*, dont les valeurs sont comprises entre 0 et 1 :



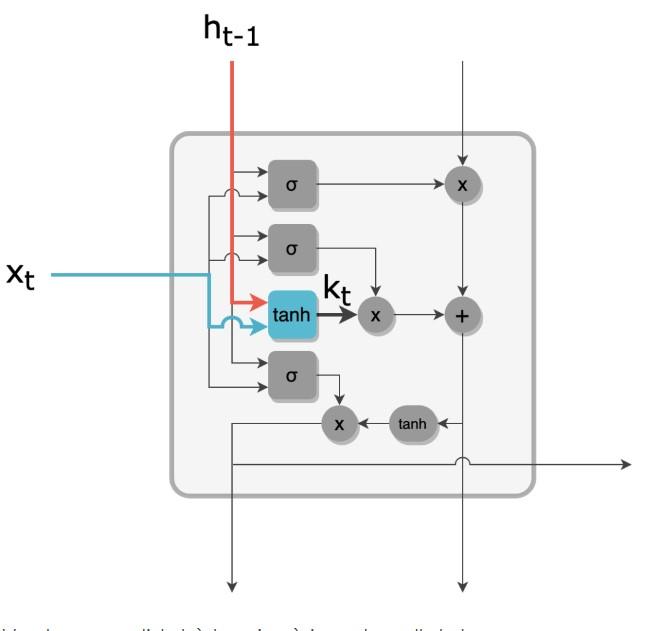
La Forget gate agit comme un filtre pour « oublier » certaines informations du Cell State. En effet, on effectue une multiplication terme à terme entre ft et *ct*−1, ce qui a tendance à annuler les composantes de *ct*−1 dont les homologues côté *ft* sont proches de 0. On obtient alors un Cell State filtré :



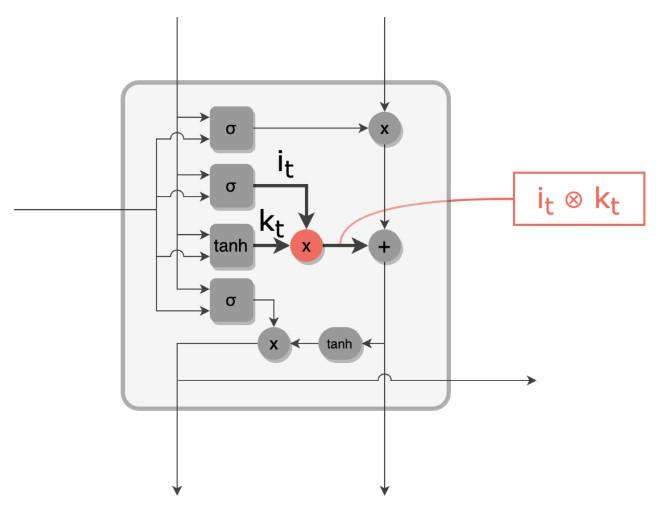
L’input gate produit un filtre *it* de valeurs comprises entre 0 et 1 , à partir de *ht*−1 et *xt*, de façon similaire à la forget gate :



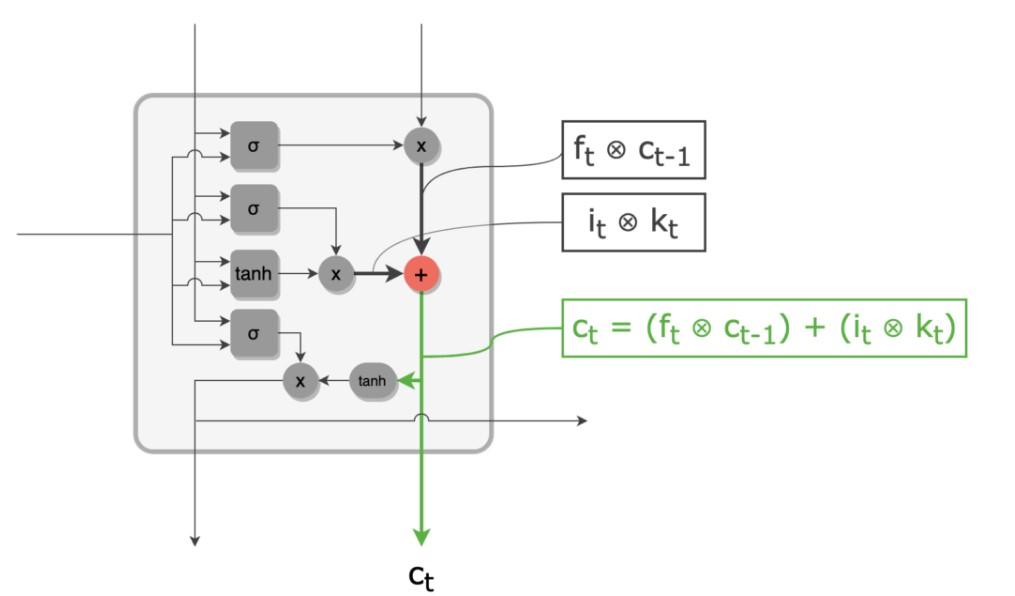
En parallèle, un vecteur *kt* est créé par une couche tanh. *kt* est le vecteur candidat pour mettre à jour le Cell state :



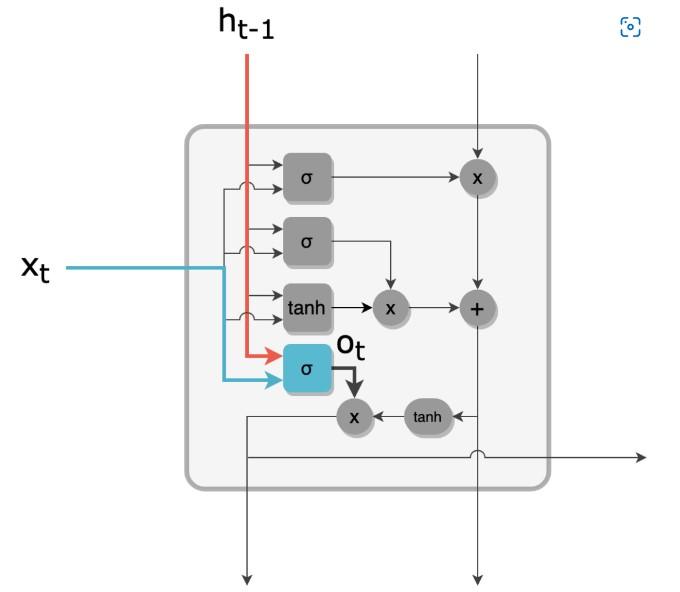
Le candidat *kt* est filtré par l’Input gate *it* via une multiplication terme à terme. On obtient le vecteur de mise à jour du Cell State :



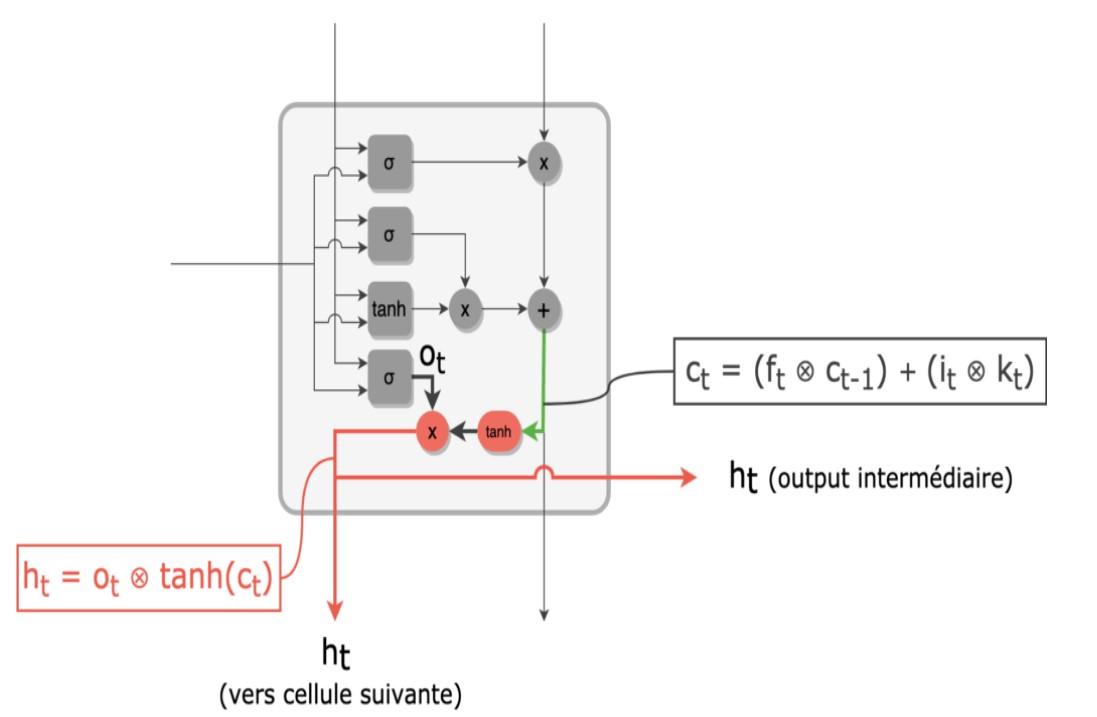
Le Cell State filtré (obtenu à l’étape 3) est mis à jour grâce au vecteur candidat filtré (obtenu à l’étape précédente). La mise à jour est une simple addition terme à terme de ces deux vecteurs. On obtient alors le nouveau Cell State c*t* :



De façon analogue à *ft* et *it*, l’output gate produit un filtre *ot* de valeurs entre 0 et 1 :



Les valeurs du nouveau Cell State *ct* sont ramenées à l’intervalle ]-1, 1[ par une activation tanh. Un filtrage par l’Output gate ot est ensuite effectué pour enfin obtenir la sortie *ht* :



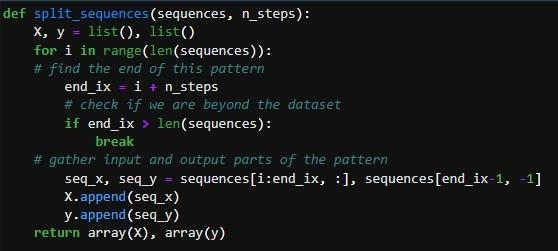
1.1.3 Détails du fonctionnement de l’algorithme

J’ai écrit ce procédé en Python à l’aide du module ScikitLearn, un module très utilisé pour effectuer des tâches nécessitant du Machine Learning.

L’algorithme fonctionne de la manière suivante :

1. : L’ensemble des séries temporelles sont empaquetés sous forme d’un Dataframe, sur chaque colonne se trouve l’une des séries temporelles et chaque ligne donne une nouvelle donnée pour toutes les séries. Un traitement des valeurs manquantes est aussi réalisé, les valeurs manquantes sont comptées puis remplacées par la valeur précédente à chaque fois.
2. : Le Dataframe est découpé de manière à tourner le problème sous forme d’un problème d’apprentissage supervisé, une fonction vient découper le Dataframe en sample de taille à choisir. C'est-à dire que le Dataframe est découpé en série de sous Dataframe de longueur n.

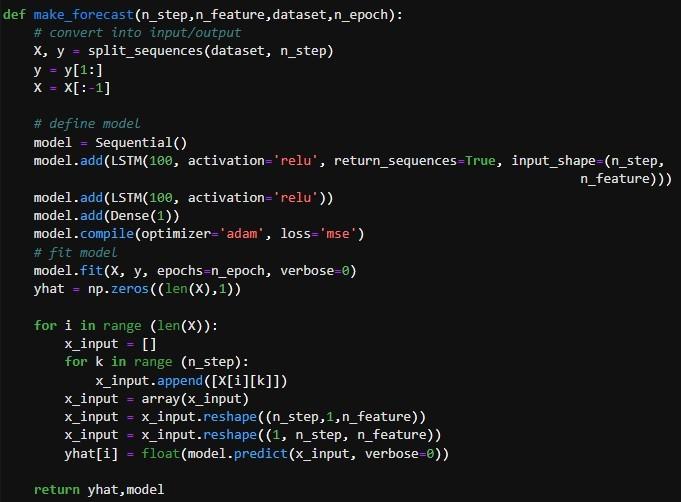
Voici la fonction de découpage :



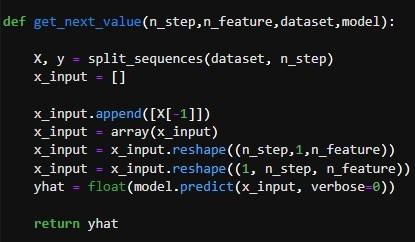
La variable nsteps permet de choisir la taille de chacun des échantillons avec lesquels le réseau de neurone va apprendre à donner son avis sur la prochaine valeur. La variable à prédire n’est autre que l’output Y de cette fonction.

1. : L'algorithme peut ensuite apprendre à donner son avis sous forme d’un apprentissage supervisé puis une autre fonction fait appel au modèle déjà entraîné pour obtenir la prochaine valeur, celle qui n’existe pas encore.

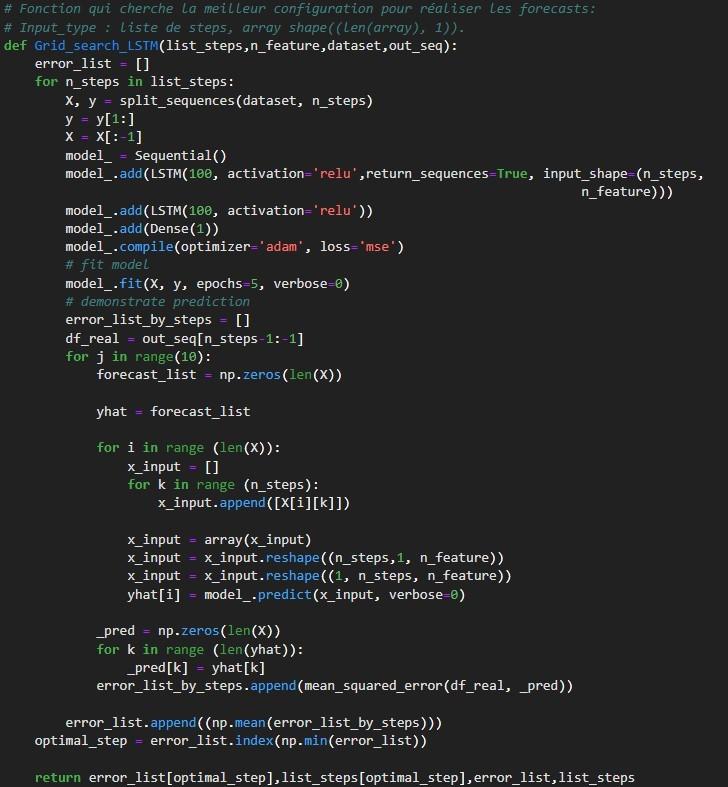
Voici la fonction d'apprentissage :



Puis la fonction de forecast :



Les fonctions que j’ai présenté précédemment sont en réalité utilisées suite à la recherche de la meilleure configuration d’apprentissage. Cette recherche de la configuration optimale se fait via une fonction que j’ai appelée Grid Search LSTM, elle renvoie la meilleure configuration en termes de l’erreur commise avec les données officielles. La fonction d’erreur utilisée est la moyenne des carrés des erreurs d’estimations.



# Construction et Optimisation des portefeuilles

2.1 Construction des portefeuilles et calcul des bornes :

Une fois le système de réseaux de neurones mis en place, on dispose d’une caractérisation de l’état de l’économie selon les quatres quadrants sur un période de 15 ans. A partir de cette caractérisation, la stratégie consiste à construire puis optimiser les quatres portefeuilles de la manière suivante :

1. : L’Univers d’investissement :

Les Bonds :

BERPG2 Index, BEUSG2 Index, BCIT1T Index, BEUSG1 Index, BERPG5 Index, BEUSG3 Index, BEUSG5 Index, BERPG1 Index,

BEUSG4 Index, BERPG4 Index

Les Equity :

SX5E Index, NDX Index, SPX Index, IXJ US Equity, IXP US Equity

Les Commodity :

XAUUSD Curncy

Les Foreign Currency :

JPYEUR Curncy, CHFEUR Curncy, USDEUR Curncy

1. : disposant d’un univers d'investissement diversifié (obligation,actions,matière première,monnaie), repérer les actifs ayant le plus grand potentiel de rendement sous chaques quadrants. Construire les quatres portefeuilles via une composition de ces actifs performants en imposant une contrainte de diversification, chaque portefeuille est au moins constitué de 3 classes d’actifs différentes.
2. : une fois les quatres portefeuilles obtenus, chercher les bornes d’allocation pour chacuns des actifs.

J’ai donc écrit un algorithme qui fait la chose suivante :

Trouver les bornes qui permettent d’avoir exactement le même risque sur chacune des lignes des portefeuilles. Soit X*i i* ∈ [1*,n*]*,* les actifs présents dans un portefeuille,

Pour chaque *Xi*, calcul du Max Drawdown, cela représente une estimation de son risque maximal potentiel.

Puis pour chaques *Xi,* trouver *Yi* tel que : *MaxDD*(*Xi*) ∗ *Yi* = 1*/n*

On obtient finalement une liste de *Yi* qui constituera l’une des contraintes au niveau du module d’optimisation, j’appelle cette liste contrainte1.

La seconde contrainte, contrainte2 consiste à faire en sorte que*.*

La troisième contrainte, contrainte3 est une contrainte de volatilité maximale : *V OL <*= *x*

* 1. Optimisation sous contrainte des portefeuilles :
     1. LA DÉCOUPE DES PORTEFEUILLES :

Chacun des portefeuilles est découpé en deux listes de sous portefeuilles, la première liste est constituée de portefeuilles de taille taille1 et sert au calcul des poids optimaux qui sont stockés dans une matrice d’allocation. La seconde liste est constituée de portefeuilles de taille taille2 et sert à calculer la performance réalisée grâce aux poids calculés au niveau de la première liste. Noté que l’emplois de deux listes est nécessaire car lors du calcul des poids pour le premier portefeuille de la première liste, afin de tester la performance de l’allocation réalisée, il faut calculer la performance réalisée avec ces poids et sur une fenêtre d'investissement postérieur à la période pour laquelle ces poids ont été calculés, le même procédé est réalisé de bout en bout sur toute la durée du Backtest qui est de 15 ans. Au niveau du module que j’ai écrit en Python, le choix de taille1 et de taille2 est libre, ce qui permet de tester différentes configuration pour le calcul de l’inertie et de la période d'investissement.

* + 1. LE CALCUL DES POIDS :

Pour calculer les poids optimaux pour chacun des portefeuilles, j’utilise un minimiseur provenant de la librairie Scikit-Learn.

Il s’agit d’un minimiseur pour lequel il est possible de spécifier des contraintes sous forme de fonctions.

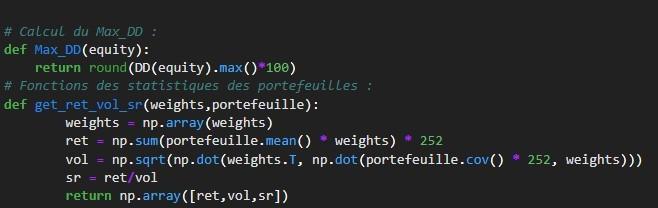
Pour transformer le problème d’optimisation sous forme d’une minimisation, la stratégie est la suivante :

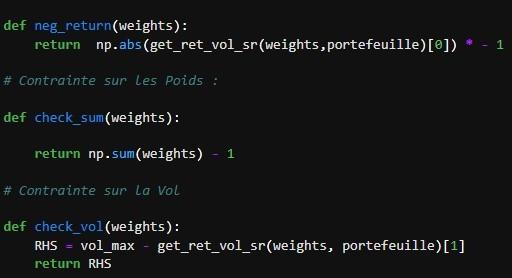
Pour chacuns des portefeuilles de taille taille1, calcul en daily du return de chacunes des lignes puis de la matrice de covariance car sur la diagonale se trouve la variance de chacune des lignes d'où l'on peut obtenir la volatilité, ces deux statistiques sont annualisés. Pour placer le problème sous forme d’une minimisation, le return calculé est multiplié par -1, le problème devient donc un problème de minimisation sous contrainte.

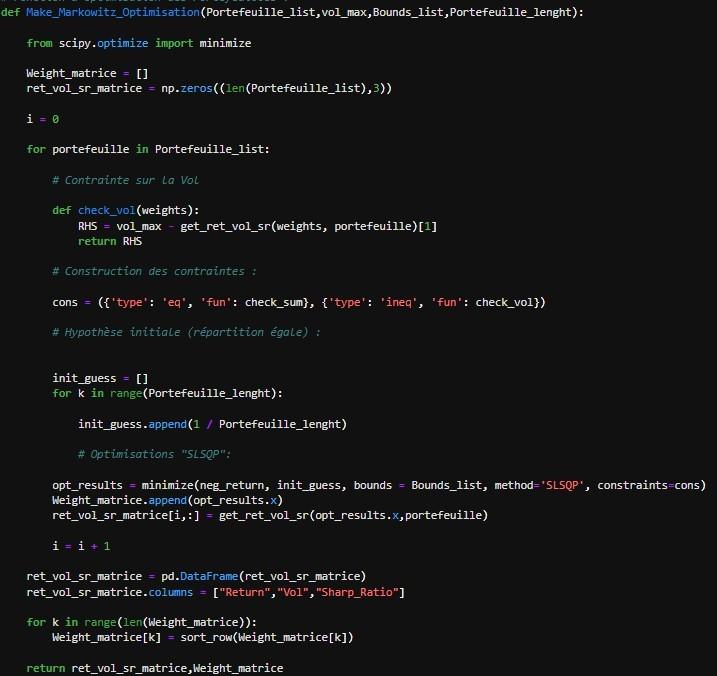
Ensuite le module effectue deux optimisations, la première consiste à minimiser le return négatif uniquement, la seconde minimise le ratio de sharp négatif, ce ratio est le ratio entre le return négatif et la volatilité du portefeuille obtenu.

Pour résumer chacun des quatres portefeuilles est découpé puis optimisé selon deux critères (Return,Ratio de Sharp) en prenant en compte les trois contraintes. Suite à cette optimisation on obtient une matrice d’allocation pour chacun des portefeuilles, sur chaques lignes de chaques matrice d’allocation se trouve une liste de poids qui correspondent à la configuration de portefeuille optimale obtenue suite à la résolution du problème de minimisation, ces matrices seront utilisées lors de la valorisation des portefeuilles de taille taille2.

Voici quelques fonctions utilisée pour mener à bien l’optimisation :







2.2.3 LA VALORISATION DES PORTEFEUILLES :

La valorisation des portefeuilles se fait comme suit : pour chaques date de rebalancement, disposant d’un capital fourni par la valorisation des portefeuilles la plus récente, de la nouvelle liste d’allocation et de la valeur actuel des actifs; je calcul pour chacuns des produits qui vont être investi, le nombre de parts que la pondération du capital total par le poids attribué permet d’acheter. Chacune de ces parts est ensuite valorisée tous les jours selon le cours de l’actif sous-jacent pendant la période d'investissement.

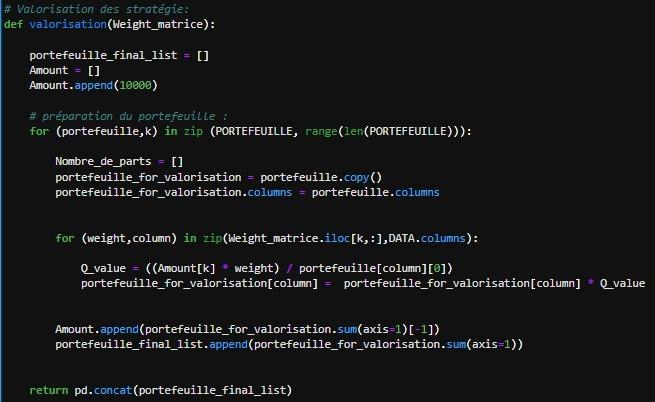
La valeur finale du portefeuille n’est autre que la somme des valorisations au cours du dernier jours de la fenêtre de rebalancement, cette valeur est stockée afin de pouvoir attribuer les nouvelles quantités initiales lors du premier jour du prochain rebalancement.

Ce procédé est réalisé pour chaques rebalancement, donc pour chaques période de taille taille2, la valorisation journalière constitue l’Equity Curve des portefeuilles :

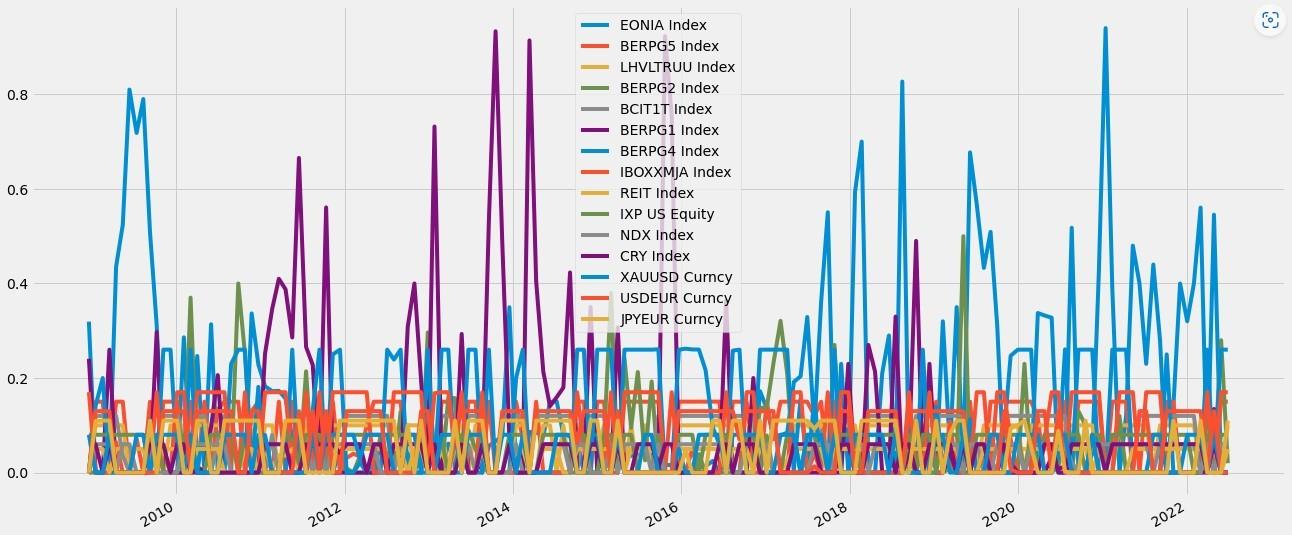


On peut remarquer que les portefeuilles ont des comportement différents, par exemple au niveau de la crise du Covid 19 le portefeuille Déflation étant majoritairement composé d’obligations subit beaucoup moins de perte que le portefeuille Stagflation qui lui est composé majoritairement d’actions Américaine et Européenne.

Voici la fonction de valorisation des portefeuilles :



Puis le profil d’allocation de l’un d’entre eux :



Une fois les quatre Equity Curves obtenu, la stratégie finale consiste à trouver la meilleur pondération possible entre les quatre portefeuilles en fonction du quadrant dans lequel le réseau de neurones se trouve car avec une pondération adéquate on va être en mesure de profiter des bons mouvement de marché en adéquation avec la configuration de chacun des portefeuilles tout en évitant la plupart des mauvaises configurations. Naturellement, lorsque la période d’investissement est caractérisée par l’un des quadrants, c’est le portefeuilles correspondants à ce quadrant qui se trouve être le plus investi, cette logique va guider la recherche de la meilleure pondération entre les quatre portefeuilles.

2.2.4 LA RECHERCHE DE LA MEILLEURE CONFIGURATION D’INVESTISSEMENT :

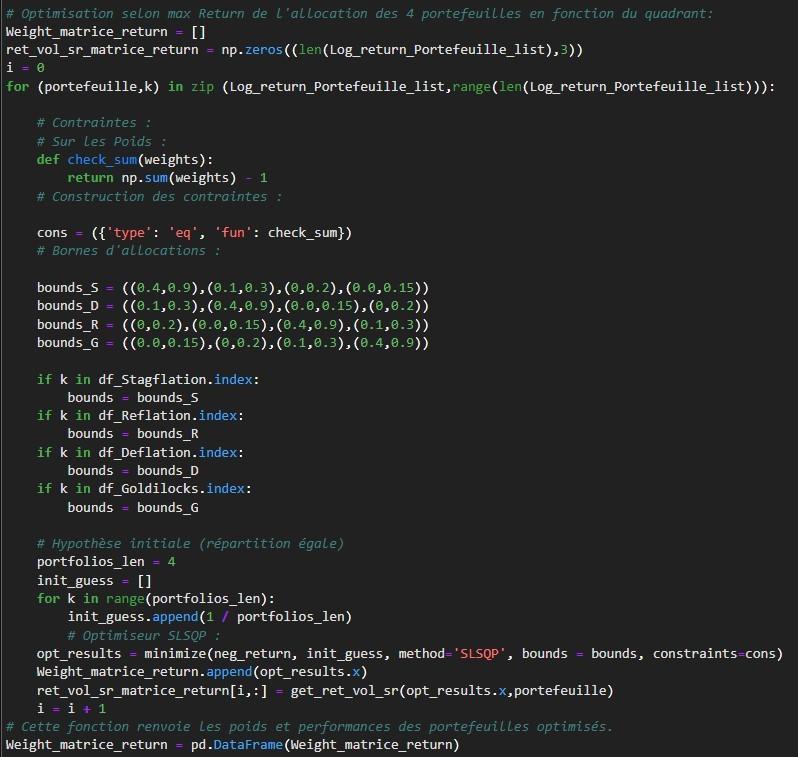
A partir des 4 Equity Curve l’objectif est de trouver la meilleure stratégie de pondération des 4 portefeuilles, l’idée est de trouver une pondération "passe partout" afin d’éviter d’avoir trop de frais de transaction et de slippage. J’ai donc re-construit un portefeuille composé des 4 Equity Curve, celles-ci jouent le même rôle qu’un actif financier. L’idée est de maximiser le Return du portefeuille avec cette fois-ci des contraintes propres à chacuns des quadrants. Le calcul des poids est tout de même un peut diffèrent de l’étape 2, en effet ici les portefeuilles composés de 4 actifs sont coupés seulement selon des sous portefeuilles de taille taille2 ce qui correspond à la fenêtre durant laquelle ils seront investis.

Les contraintes sont les suivantes :

Comme chacun des portefeuilles de taille taille2 se trouve sous l’un des quadrants, je fixe une liste de 4 fenêtres de pondération puis sous chaque quadrants les fenêtres de cette liste sont attribuées à chaque portefeuille.

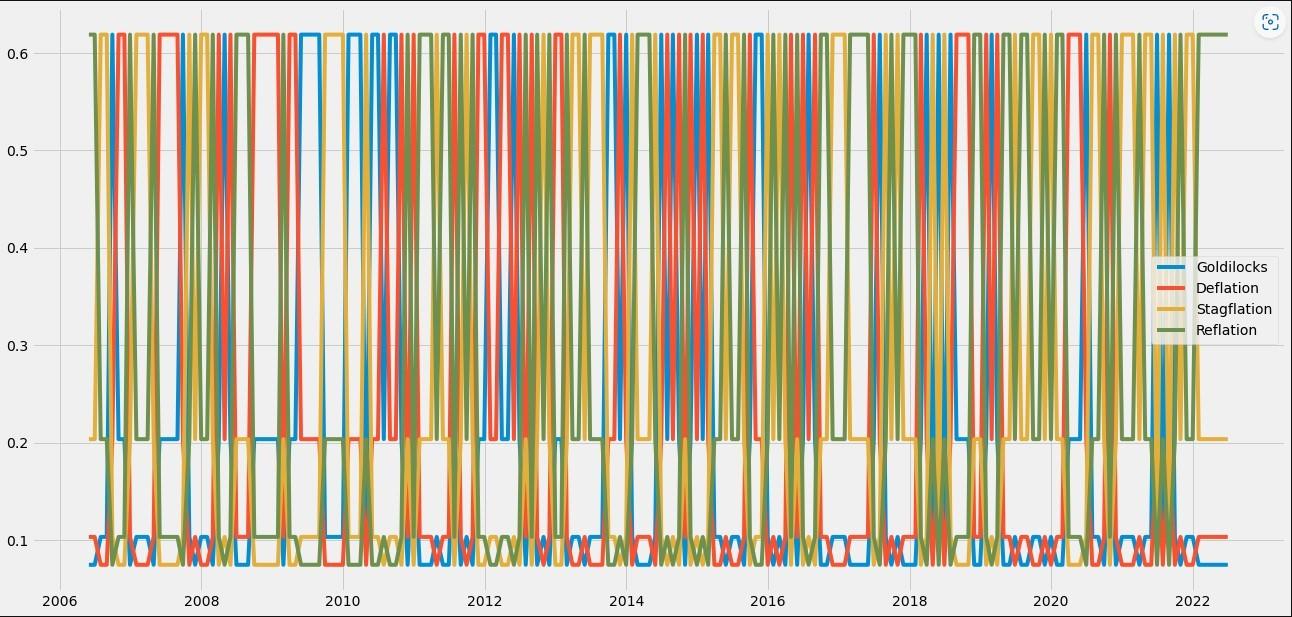
Le portefeuille en phase avec le quadrant se voit attribuer la fenêtre (0.4,0.9), le portefeuille en opposition de phase avec le quadrant actuel se voit attribuer la fenêtre (0.0,0.15), le portefeuille le plus proche en terme de composition se voit attribuer la fenêtre (0.1,0.3) et le dernier, celui qui est proche du portefeuille en opposition de phase se voit attribuer la fenêtre (0.0,0.2).

Cela donne l’algorithme d’optimisation suivant :



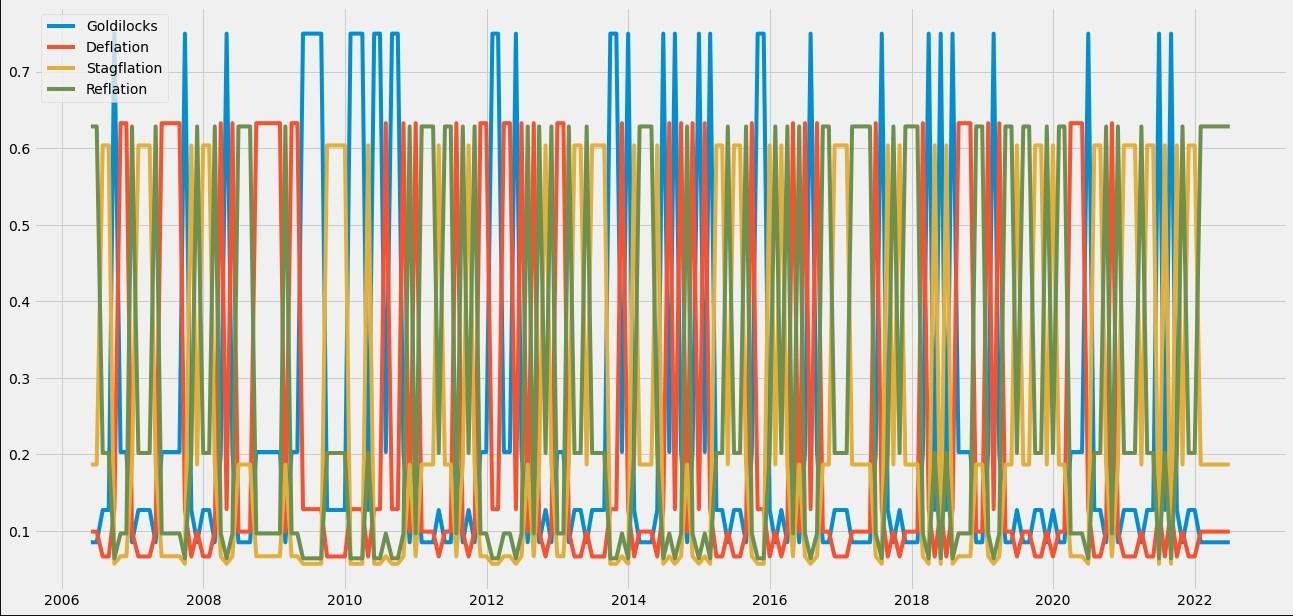
L’algorithme précédent permet de construire deux stratégies différentes, la première : "Statique" consiste à calculer la moyenne de la pondération obtenue pour chacun des portefeuilles. On obtient au final une pondération unique quel que soit le quadrant macroéconomique (La pondération du portefeuille dans son quadrant est identique quel que soit le quadrant et ainsi de suite pour chacun des portefeuilles).

Voici le profil de pondération obtenu suite à cette stratégie :



La seconde stratégie : "Dynamique" consiste à calculer la moyenne de la pondération des 4 portefeuilles propres à chacuns des quadrants. On obtient dans ce cas quatres configurations de pondérations caractéristiques de chacuns des quadrants (4 profils de pondérations, un par quadrant).

Voici le profil de pondération obtenu suite à cette stratégie :

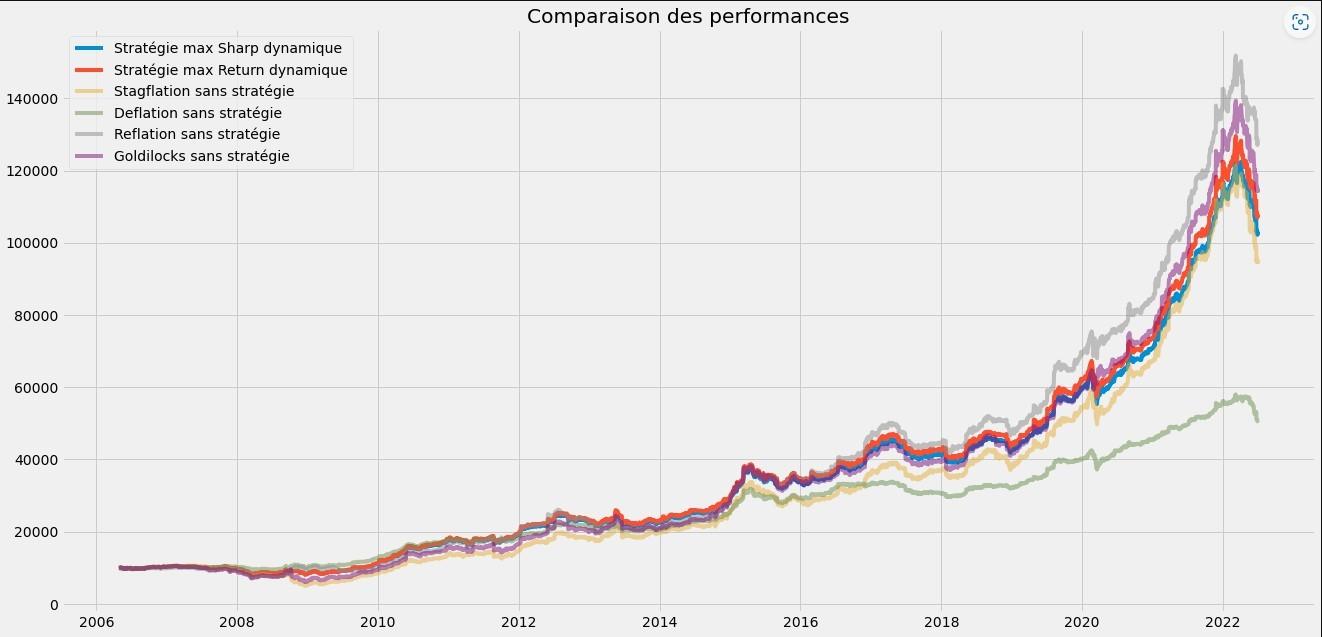


Une fois ces profils obtenus, je passe à la valorisation finale des stratégies :

Stratégies statiques par MaxSharp, MaxReturn et Portefeuilles :



Stratégies dynamiques par MaxSharp, MaxReturn et Portefeuilles :



# Bilan et avis à propos de mon stage

Dans l’ensemble, mon stage s’est bien déroulé. J’ai appris énormément de choses quant à la gestion de portefeuille dans son ensemble, Olivier Berthier à répondu à toute mes questions et m’a permis d’écrire en Python de quoi construire une stratégie de gestion en intégrant des signaux Macroéconomique et du traitement via Machine-Learning.

Mathias Choussy à quant à lui, été très utile pour le développement de ma culture financière en général. J’ai en effet acquis un regard panoramique portant sur l’ensemble des déterminants de la dynamique des marchés financiers, je pense notamment aux cycles du crédit, du business, de l’amélioration et détérioration cyclique de la performance du secteur privé, de la façons par laquelle les politiques des banques centrales visent à réguler l’inflation et encadrer les rebonds de l’économie.

J’ai aussi pris connaissance de la composition du comité financier de la FED des Etats-Unis, de l’importance de ses décisions sur les marchés financiers en général. J’ai même mis en place un réseau de neurones équivalent à celui précisé plus haut afin de prévoir la prochaine annonce des taux U-S.

Les données sur lesquelles il se base sont le suivantes :

1. : ED1 COMB Comdty : Eurodollar 3 mois

Le taux de l’emprunt du dollar à trois mois, le reflet de la valeur dans 3 mois du dollar perçu aujourd’hui.

1. : FEDL01 Index : Le taux FED

Série temporelle du taux officiel de la FED des Etats-Unis.

1. : CPI YOY Index : US Consumer Price Index YOY pctchange

Le chiffre officiel de l’inflation aux USA exprimé en pourcentage de changement d’une année sur l’autre.

1. : SPY US Equity

ETF qui réplique la performance du SPX, indice regroupant les grandes entreprises Américaines.

1. : GDP CYOY Index : US Growth Domestic Power Index YOY pctchange

Le chiffre officiel de la croissance aux US exprimé en pourcentage de changement d’une année sur l’autre.

1. : NAPMPMI Index : ISM Manufacturing PMI SA, mesure l’état de l'activité de la manufacture et de l’économie en général.
2. : OEUSMBAF Index : OECD USA BROAD MONEY M3 YOY SA

Variation d’une année sur l’autre de la quantitée de monnaie en circulation aux Etats-Unis (monnaie physique + comptes courants + grands comptes + monnaie interbancaire).

[1] [2]

# Références

1. SHI YAN. Understanding lstm.
2. VICTOR BELLISSANT. Recherche et développement delubac asset management.