ECE PARIS
ÉCOLE D'INGÉNIEURS

L'arrivée de l'attention dans la prédiction de séries temporelles

Matthias NGD

Valentin DAVIS

07/04/2021

Responsable : XXXXXXX
Matière : IA Finance

Table des matières

Introduction

Depuis longtemps les prévisions de séries chronologiques sont au cœur des recherches dans plusieurs domaines (Finance/Economie, Santé, Météorologie, Santé) car beaucoup de ces sujets sont indexé à une date. Les ressources en données ne font que croitent depuis des années et les méthodes pour les exploiter et améliorer les prédictions aussi.

L'un des outils les plus performant aujourd'hui pour traiter des data-set toujours de plus en plus volumineux est le Deep Learning. L'un des avantages qui le distingue de ses prédécesseurs (modèles autorégressifs-AR) est le fait qu'il n'ait pas besoin qu'on lui implémente à « la main » les fonctionnalités, ils les apprennent d'eux-mêmes. Les modèles d'apprentissage permettent de travailler le process de données plus rapidement et donc d'en gérer des plus compliqués aux critères plus précis.

A travers cette recherche, nous allons dans un premier temps définir les séries temporelles et ses différentes composantes. Nous complèterons avec l'explication en détails de plusieurs méthodes très utilisées dans l'étude des prévisions séries chronologiques. Nous ne finirions pas axée l'utilisation de ces méthodes dans le cas des prévisions des cours boursiers, les inconvénients d'une telle méthode et les nouvelles solutions pour pallier les problèmes.

En effet, le facteur aléatoire sur le court terme est un poids non négligeable lorsqu'on cherche à déterminer le potentiel prix d'une action. On parle de ***Bruit*** lorsqu'un semble de données possèdent des fluctuations aléatoires souvent perceptibles à court terme. Il faut donc une architecture capable de pouvoir gérer ces mouvements à court terme mais aussi les adapter pour des études sur le long terme.

Avant de rentrer dans le détail de nos méthodes de Deep Learning, il est aussi important de rappeler que d'autres modèles de Machine Learning connu des analystes existent et sont aussi utilisés pour résoudre ce genre de problématique. Les modèles ARIMA (cf. fichier python) défini comme des combinaisons d'approches autorégressive et de moyenne mobile, sont simples et fournissent des résultats probants même s'ils possèdent quelques limitations :

- Les valeurs manquantes dans un data-set affectent souvent les performances des modèles
- Incapacité à reconnaître les modèles complexes
- Plus adapté aux modèles à court terme, moins à long terme

Le Deep Learning palie aux problèmes rapportés ci-dessus mais surtout possèdent des approches différentes.

I – Prévision des séries temporelles : Théories & méthodes

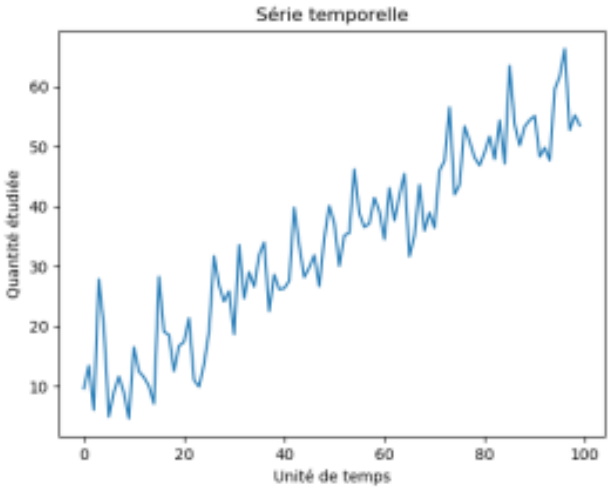
Comme vu précédemment, Une série temporelle, est une séquence / suite de valeurs numériques indexées dans le temps, souvent avec un même pas de temps séparant deux observations successives.
Les séries temporelles n'étant qu'une suite de mesures numériques à intervalles réguliers, elles sont effectivement utilisées dans bon nombre de domaine. Cependant dansa cette partie nous nous pencheront uniquement sur la théorie et les méthodes concernant la prédiction via séries temporelles.

a) Les composantes d'une série temporelle :

Avant de découvrir les méthodes de traitement applicable aux séries temporelles, nous allons nous attarder sur les composantes de celles-ci :

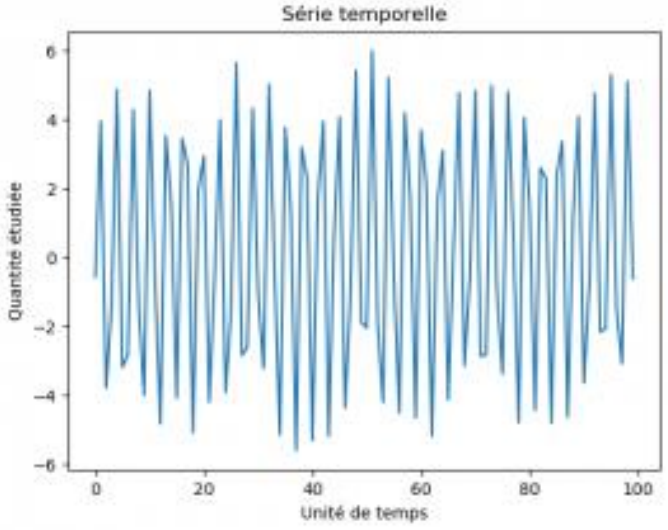
- Une tendance générale ?

La tendance représente la direction générale des données. Pour ce faire on observe la série uniquement sur une longue durée, on ignore les variations à court terme ou les variations liées au bruit.



- Une saisonnalité ?

La saisonnalité fait référence aux fluctuations périodiques qui se répètent tout au long de la période de la série chronologique.



- Autocorrélation

L'autocorrélation est la corrélation entre la série chronologique et une version décalée d'elle-même, et est utilisée pour identifier la saisonnalité et la tendance dans les données de séries chronologiques.

$$\rho_k = \frac{\gamma_k}{\sigma^2}$$

Avec

$$\gamma_k = E\big((X_t - \mu)(X_{t+k} - \mu)\big)$$

On note également ρ_k l'autocorrélation, σ l'écart-type et μ la moyenne de la variable X_t.

- Stationnarité ?

La stationnarité est une caractéristique importante des séries temporelles. Une série temporelle est dite stationnaire si sa moyenne, sa variance et sa covariance ne subissent pas de changements significatifs dans le temps.

Maintenant que l'on sait ce qu'est une série temporelle, on peut se pencher vers les différentes méthodes dans lesquelles elle intervient.

b) La méthode Convolutional Neural Networks

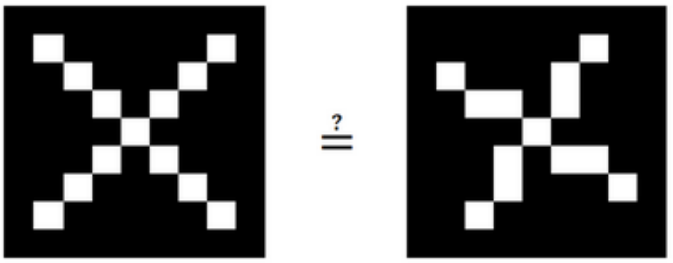
Un réseau de neurones convolutifs (CNN) est un type de réseau de neurones artificiels. Les réseaux neuronaux convolutifs ont de larges applications dans la reconnaissance d'image et vidéo, les systèmes de recommandation et le traitement du langage naturel.

Un CNN applique généralement 2 types d'opérations différentes à une image afin d'en extraire les informations pertinentes.

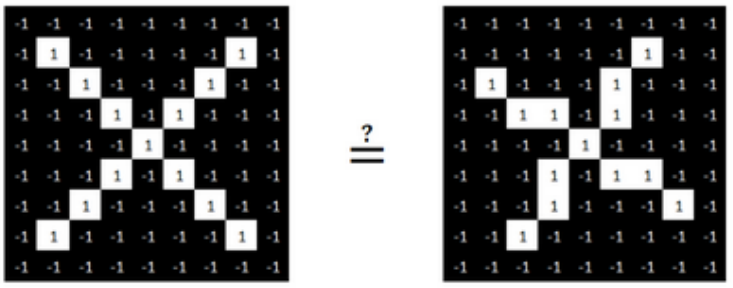
Ces 2 types d'opérations sont les suivantes :

- La *convolution*
- La fonction d'activation

Nous prendrons ici l'exemple le traitement d'une image. Les deux images ci-dessous nous serviront donc d'exemple.



En fonction des couleurs, on attribue le chiffre -1 ou 1 :



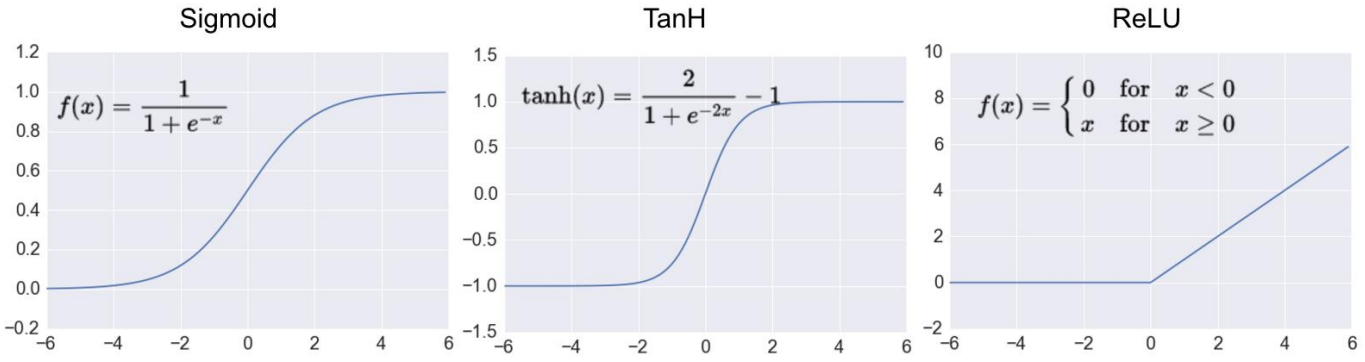
La convolution : On cherche donc à savoir si c'est deux images sont identiques. Pour ce faire, on utilise un filtre, par exemple le filtre suivant :

1	-1
-1	1

Une fois le filtre déterminé, on va l'appliquer à l'image que l'on souhaite tester. On obtient alors une nouvelle matrice de valeur.

0,8	-0,1	0,1	0,3	0,6	-0,1	0,3
-0,1	1,0	-0,1	0,3	-0,1	0,1	-0,1
0,1	-0,1	1,0	-0,3	0,1	-0,1	0,6
0,3	0,3	-0,3	0,6	-0,3	0,3	0,3
0,6	-0,1	0,1	-0,3	1,0	-0,1	0,1
-0,1	0,1	-0,1	0,3	-0,1	1,0	-0,1
0,3	-0,1	0,6	0,3	0,1	-0,1	0,8

C'est à ce moment-là que la fonction d'activation intervient. On en dénombre 3, visible ci-dessous :

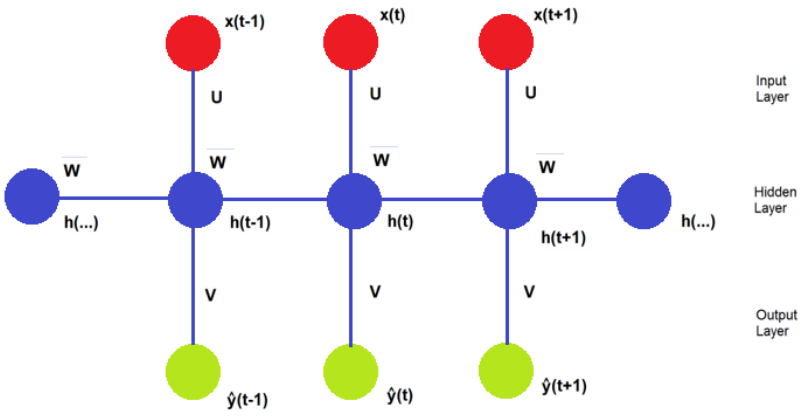


Actuellement, le ReLu est de loin la fonction d'activation non linéaire la plus utilisée. Les principales raisons sont qu'elle réduit la probabilité d'un gradient évanescent et la sparsité. La dernière couche du modèle utilise généralement une fonction d'activation différente, car nous voulons que cette couche ait une certaine sortie. La fonction softmax est aussi très populaire pour la classification.
Finalement, cette fonction d'activation permet de transformer les valeurs négative en 0 tout en conservant les valeurs positives.
Au fur et à mesure des itérations, la fonction de coût s'optimise et on obtient nôtre résultat.

c) La méthode Recurrent Neural Networks

Les réseaux de neurones artificiels sont des systèmes informatiques inspirés des réseaux de neurones biologiques « standard » qui constituent les cerveaux biologiques. Ces systèmes apprennent un phénomène en considérant et analysant un grand nombre de données. Un réseau est basé sur une collection d'unités interconnectées appelées neurones. Les neurones sont divisés en couche d'entrée, couches cachées et couche de sortie.

Dans le cas d'une série temporelle, chaque neurone est affecté à un pas de temps fixe. Les neurones de la couche cachée sont également transmis dans une direction dépendante du temps, ce qui signifie que chacun d'entre eux n'est entièrement connecté qu'avec les neurones de la couche cachée ayant le même pas de temps assigné, et est connecté avec une connexion à sens unique à chaque neurone assigné au pas de temps suivant. Les neurones d'entrée et de sortie sont connectés uniquement aux couches cachées avec le même pas de temps assigné.



W → X_t est connecté aux neurones de la couche cachée du temps t par une matrice U, les neurones de la couche cachée sont connectés aux neurones des temps t-1 et t+1 par une matrice de poids W, et les neurones de la couche cachée sont connectés au vecteur de sortie Y_t par une matrice V ; toutes les matrices sont constantes pour chaque pas de temps.

Input → Le vecteur X_t est l'entrée du réseau au pas de temps t.

Hidden Layer → Le vecteur h(t) est l'état caché au temps t, et est une sorte de mémoire du réseau ; il est calculé sur la base de l'entrée actuelle et de l'état caché du pas de temps précédent :

$$h(t) = tanh\big(w_h(t-1) + \cup x(t)\big)$$

Output →Le vecteur $\hat{y}(t)$ est la sortie du réseau au temps t :

$$\hat{y}(t) = soft\,max\big(V_s(t)\big)$$

Ces neurones permettent à l'algorithme « d'apprendre » et pour cela il fait appel à un « Learning Algorithm ». On cherche à trouver les paramètres optimaux pour U, V et W ce qui nous permettra d'obtenir la meilleure prédiction possible. Pour ce faire, on utilise une fonction appelée fonction de perte et notée J, qui quantifie la distance entre les valeurs réelles et les valeurs prédites sur l'ensemble de l'apprentissage. Elle est donnée par :

$$j(\theta) = \frac{1}{m}\sum_{i=1}^{m}\sum_{i=1}^{n} L\big(\hat{y}(t), y(t)\big)$$

Où :

- La fonction de coût L, évalue les distances entre les valeurs réelles et prédites pour un pas de temps ;
- m est la taille de l'ensemble d'apprentissage ;
- θ le vecteur des paramètres du modèle.

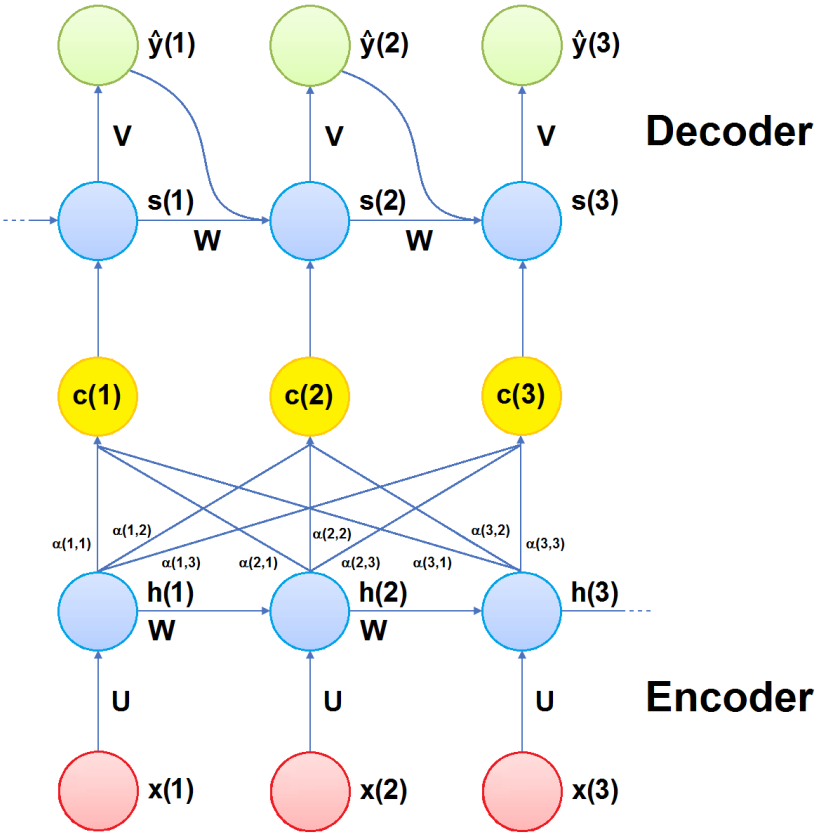
Il faut ensuite minimiser cette fonction J, c'est pour cela qu'on utilise « The Forward Propagation » & « The Backward Propagation ». On reproduira cette action un certain nombre de fois afin d'atteindre un certain nombre d'itération.

The Forward Propagation → Avec des paramètres fixes U, W et V, les données sont propagées dans le réseau et à chaque instant t, on calcul $\hat{y}(t)$ en utilisant les formules définies précédemment. A la fin, la fonction de perte est calculée.

Back propagation → Les gradients de la fonction de coût sont calculés par rapport aux différents paramètres, puis un algorithme de descente est appliqué afin de les mettre à jour. Les gradients à chaque sortie dépendent à la fois des éléments du même pas de temps et de l'état de la mémoire au pas de temps précédent.

d) La méthode Attention Mechanism

L'Attention Mechanism est une évolution du modèle encodeur-décodeur, développé afin d'améliorer les performances sur de longues séquences d'entrée.
L'idée principale est de permettre au décodeur d'accéder sélectivement aux informations de l'encodeur pendant le décodage. Le précédé étant de construire un vecteur de contexte différent pour chaque pas de temps du décodeur, en le calculant en fonction de l'état caché précédent et de tous les états cachés de l'encodeur, en leur attribuant des poids entraînables.
De cette façon, le mécanisme d'attention attribue une importance différente aux différents éléments de la séquence d'entrée, et accorde plus d'attention aux entrées les plus pertinentes.



Partie Encodeur :

À chaque pas de temps, la représentation de chaque séquence d'entrée est calculée en fonction de l'état caché du pas de temps précédent et de l'entrée actuelle. L'état caché final contient toutes les informations

codées des représentations cachées précédentes et des entrées précédentes.

$$h(t) = F\big(w_h(t-1) + \cup x(t)\big)$$

La différence entre le mécanisme d'attention et le modèle codeur-décodeur classique est qu'un vecteur de contexte différent c(t) est calculé pour chaque pas de temps t du décodeur.
Ce vecteur de contexte est l'état caché final produit par la partie encodeur, et représente l'état caché initial pour le décodeur.

Pour le vecteur de contexte c(t) pour le pas de temps t :

$$c(t) = \sum_{j=1}^{T} \alpha(j,t)h(j)$$

Où :

$$\alpha(j,t) = \frac{e^{(e(j,t))}}{\sum_{j=1}^{m} e^{(e(j,t))}}$$

$\alpha(j,t)$ permet d'estimer l'importance de l'entrée du pas de temps j afin de décoder la sortie du pas de temps t.
Avec :

$$e(j,t) = v_a \, tanh\big(U_a s(t-i) + w_{ah}(j)\big)$$

Partie Décodeur :

Le vecteur de contexte c(t) est ensuite transmis au décodeur, qui calcule la distribution de probabilité de la prochaine sortie possible. Cette opération de décodage est effectuée pour chaque pas de temps présents dans l'entrée.
Ensuite, l'état caché actuel s(t) est calculé selon la fonction d'unité récurrente, en prenant comme entrée le vecteur de contexte c(t), l'état caché s(t-1) et la sortie y^(t-1) du pas de temps précédent :

$$s(t) = F\big(s(t-1), \hat{y}(t-1), c(t)\big)$$

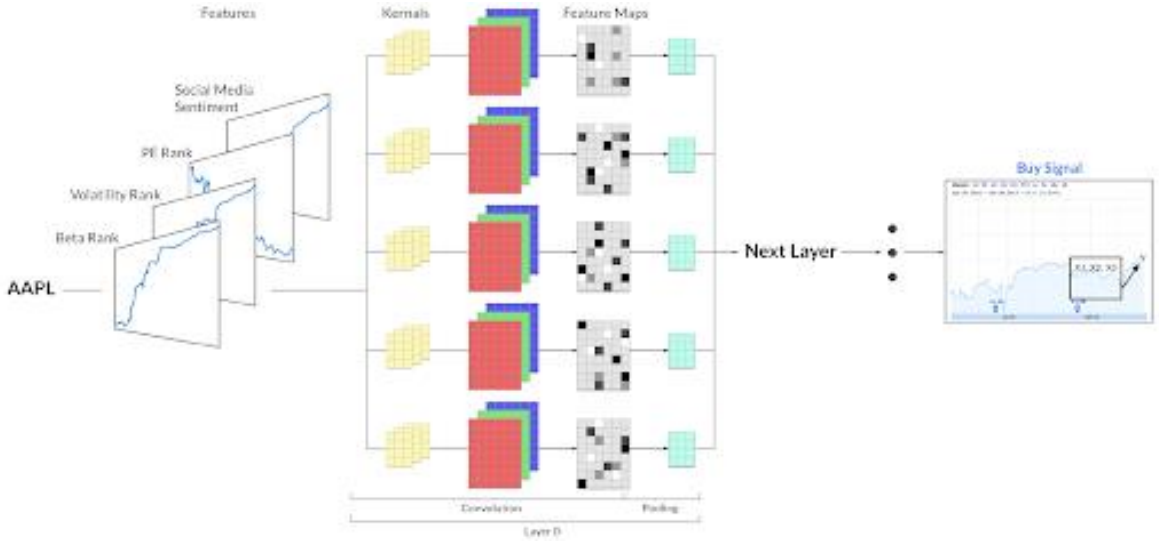
Ainsi ce mécanisme est capable de trouver les corrélations entre différentes parties de la séquence d'entrée et les parties correspondantes de la séquence de sortie.

*

II – Application en finance de marché

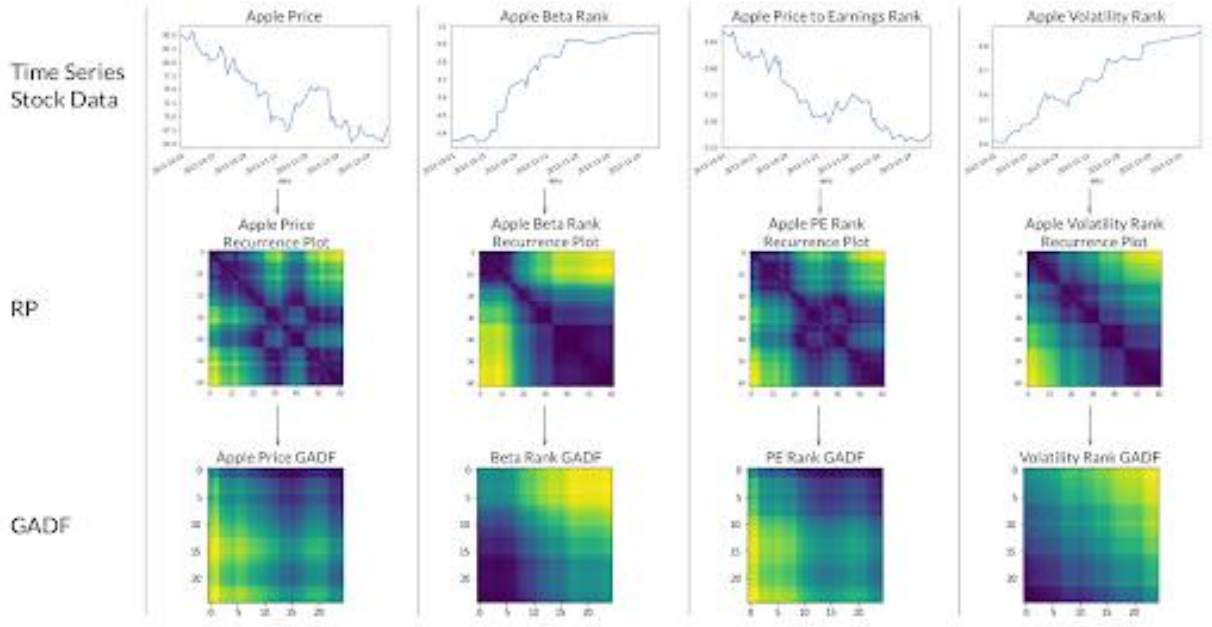
Après avoir expliqué les Réseaux de neurones récurent, la méthode Attention Mechanism et les CNN, nous pouvons aborder la prévision d'action boursières. Nous nous sommes inspirés de deux articles de Paul Wilcox, qui détaille l'utilisation des CNN sur la prévision des stocks.

Expliqué plus haut, les CNN sont utilisés pour la classification es images, on peut donc procéder ainsi avec des images de graphiques de séries chronologiques pour une action imminente des prix.



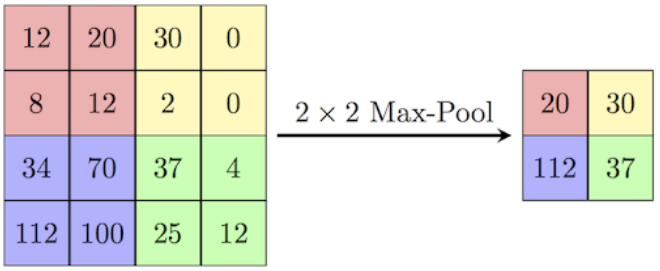
a) Inconvénients

Les images ci-dessus montrent bien qu'il est possible de mettre en entrée une multitude de canaux d'images de séries temporelles (volatilité, tendance, PE rank) comme une photo d'oiseau décomposé en trois canaux (RGB). Le résultat est un signal d'achat/vente. Une autre technique est tout aussi efficace mais avec des images bidimensionnelles, celles-ci sont plus riches.



Lorsque des comparaisons ont été faites avec d'autres processus, ils se sont rendu compte que la technologie était certes innovatrice mais moins performante qu'un autre classificateur traditionnel d'apprentissage automatique.

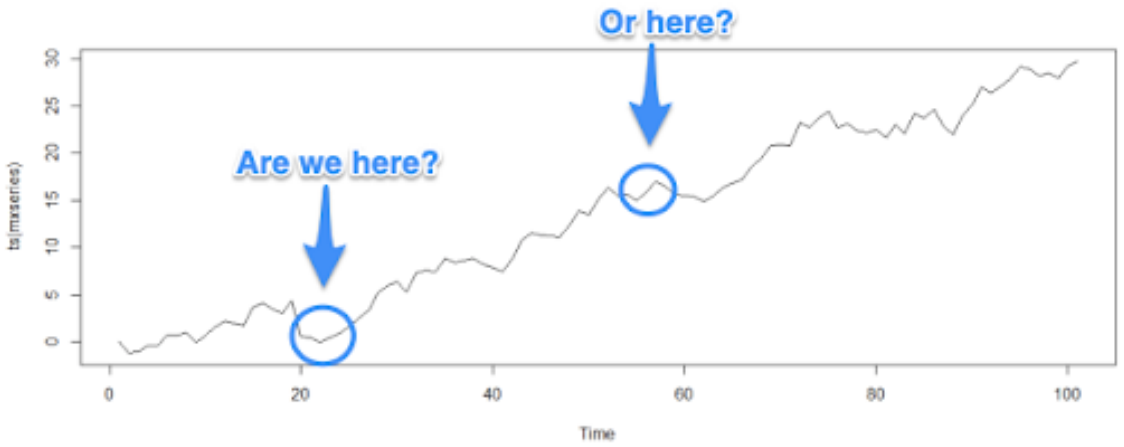
Son principal défaut est la perte d'informations importantes dû à une méthode de représentation d'approximation : Max Pooling. Elle permet de réduire les caractéristiques des échantillons dans une image afin d'effectuer un traitement plus rapide. Pour simplifier, cette méthode représente plusieurs neurones avec une seule valeur basée sur le neurone détenant la plus forte intensité scalaire, cette généralisation est à la source des pertes d'informations critiques.



On rappelle que les CNN sont hautement sollicités dans la reconnaissance d'image (voiture autonome), le fait que la mise en commun maximale soit aussi au cœur de cette méthode il serait donc faisable de corrompre le réseau en envoyant des informations biaisées et donc tromper la classification des pixels. Un danger qu'il faudrait à tout prix éviter lorsqu'une voiture autonome doit reconnaître des formes qui se ressemblent fortement.

Si on prend 2 images de chiens, l'une parfaitement normal et l'autre modifié (tête à l'envers ou pates sur le dos) le CNN les identifie comme 2 chiens car il ignore les relations proportionnelles contextuelles contenue dans une image.

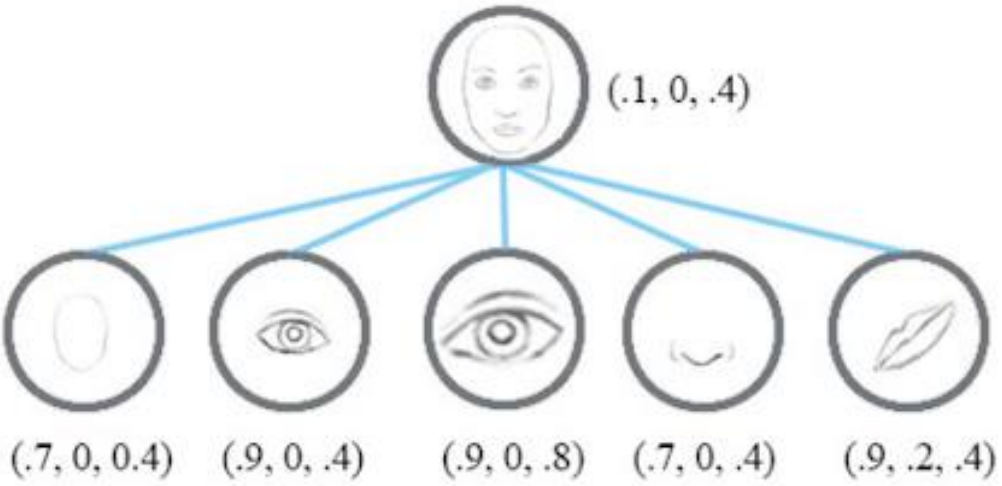
Dans le cas de la prévision des séries chronologiques, s'il y a une mauvaise classification des images graphes ou bidimensionnelles il pourrait avoir du mal contextualisé si la tendance (hausse ou baisse) est en début ou fin d'un cycle. Il considèrerait donc 2 inflexions de tendances haussières identique.
Le problème est que la représentation des données internes d'un réseau de neurones convolutifs ne prend pas en compte les hiérarchies spatiales importantes entres les objets simples et complexes.



b) Amélioration

Le 26 octobre 2017, le célèbre Geoffrey Hinton et son équipe ont introduit un nouveau type de réseau neuronal basé sur les ***capsules,*** ils ont aussi publié l'algorithme « **Dynamic routing between capsules** » (lien dans la bibliographie). Capsule Network, aka CapsNet, se base sur le regroupement des neurones et la formation du groupe (capsule) en une seule entité, les CNN classique entraîne les neurones uns-à-uns.

Les capsules ont des propriétés qui dépendent des objets traités (taille, positions, teinte etc) et ses attributs sont au sein d'un vecteur. Le principe de poids est le même que celui expliqué dans la partie théorique vu précédemment. Dans le cas des capsules, la longueur du vecteur de sortie correspond à la probabilité qu'une des caractéristiques soit présente dans un niveau supérieur. Les valeurs des longueurs doivent être comprises entre 0 et 1.



Un réseau de capsule est identique à un cadre de couche hiérarchiques avec des entités décomposées en sous-entités. Lorsqu'une photo est classée, un chemin se forme afin d'identifier les caractéristiques avec la

probabilité la plus élevée parenté à une couche supérieur. Le **Dynamic routing between capsules** permet de créer ce chemin optimal, il commence parle les niveaux inférieurs pour terminer sa sortie à l'image.

L'article propose deux tests pour évaluer la faisabilité d'utiliser CapsNet aux données boursières. Le premier test s'est fait avec des fonctions chronologique sinusoïdale, transformé en image bidimensionnel. Les résultats se sont avérés probant, le backtest affiche des résultats cohérent.

Dans le deuxième test, ils ont produit le même schéma que dans le premier, mais du bruit a été ajouté aux données. Comme on l'a expliqué, le bruit est un facteur aléatoire, cela pourrait donc donner deux tracés de récurrences avec le même segment d'onde sinusoïdale mais une finalité différente (achat ou vente). Mais encore une fois, le backtest prouve que même avec du bruit les résultats sont bons.

Le processus de Hinton nous aide surtout à détecter nos cours d'action mais il manque une information qui permet de définir l'état actuel. Le comportement de la série amène à cette information. L'exemple ci-dessous nous montre des modèles de réversion moyenne dans une graphique de série chronologiques, qui se ressemble mais restent différents en fonction des étapes précédentes. Il est donc important que le réseau discerne bien les séquences et retienne que les informations importantes.

Pour déduire un modèle prédictif, le modèle tente de trouver des prédictions avec les données historiques et regarde si ses résultats sont cohérents. C'est ce que l'architecture du transformateur avec mécanisme d'attention fait, il détecte les séquences similaires. Il est donc capable d'extraire ce qui est important en évaluant chaque pic de courbes en combinaison avec d'autres pics.
Dans la suite de l'étude nous essayons de mettre en pratique via l'utilisation de code open source les techniques abordée brièvement dans ce papier, afin de comparer un CNN vs CapsNet, et essayer de faire des prédictions avec CNN vs CapsNet vs LSTM.

Conclusion

Dans ce devoir, nous avons donc pu découvrir les séries temporelles. Pour ce faire, nous avons pris connaissance des différentes composantes des séries temporelles qui les différencies des autres séries de données plus classique.

Nous avons pu voir différentes méthodes applicables aux séries temporelles, ces méthodes sont plus ou moins efficaces en fonction des cas d'utilisation.

Comme le RNN et le CNN sont les techniques d'apprentissage dite profond les plus connues pour la prévision des séries temporelles, car elles permettent de faire des prédictions fiables sur les séries temporelles dans de nombreux cas d'application. Malgré cela, on a pu constater que RNN et CNN souffre d'un problème de gradient évanescent lorsqu'on les applique à de longues séquences.

Mais aussi le modèle encodeur-décodeur qui est la technique la plus courante pour les problèmes de correspondance séquence-séquence où les séquences d'entrée diffèrent en longueur des séquences de sortie. Cependant, l'utilisation de cette méthode entraine une baisse de performance non négligeable.

C'est pour cela que les méthodes : LSTM et mécanisme d'attention ont été développé, pour pallier les défauts cités précédemment.

LSTM a été créé pour atténuer le problème du gradient de fuite des RNN grâce à l'utilisation de portes qui régulent le flux d'informations dans la chaîne de séquences.

Le mécanisme d'attention est une évolution du modèle encodeur-décodeur, a quant à lui été développé pour pallier la baisse de performance du modèle encodeur-décodeur en présence de longues séquences, en utilisant un vecteur de contexte différent pour chaque pas de temps.

Finalement, ce devoir nous a permis de découvrir ces différentes méthodes d'un point de vue théorique mais également en termes d'application au domaine de la finance de marché.