

# L'Étude de la Prédiction des Incertitudes dans les Séries Temporelles

Cauê Caviglioni

Benjamin Richards

## I. L'INTRODUCTION

Le but de ce projet est d'étudier la précision des incertitudes dans les séries temporelles à l'aide des réseaux neuronaux. Plusieurs méthodes peuvent être utilisées pour atteindre cet objectif. Cependant, dans le cadre de notre projet, nous nous concentrons uniquement sur les modèles autorégressifs d'ordre 1 AR(1), qui supposent que les données suivent un processus de la forme suivante :

$$x_t = \theta_1 x_{t-1} + \theta_0 + \epsilon_t$$

où  $x_t$  représente la valeur à l'instant  $t$ ,  $\theta_1$  et  $\theta_0$  sont des paramètres du modèle, et  $\epsilon_t$  est un terme d'erreur stochastique.

Ce modèle montre que chaque point,  $x_t$ , dépend uniquement du point précédent,  $x_{t-1}$ , d'une constante, et d'un terme de bruit. Par conséquent, le problème de prédiction peut être représenté comme un problème de régression linéaire. Sur le plan des réseaux neuronaux, estimer les paramètres de cette relation linéaire est relativement simple. Toutefois, l'objectif de ce projet est d'estimer également l'erreur type associée aux prédictions, c'est-à-dire la variabilité des prédictions autour de la tendance. L'estimation de l'erreur type est un défi moins évident. Pour relever ce défi, nous prenons trois approches, où chaque approche dépend de la hypothèse que nous faisons sur le comportement de l'erreur type :

- L'estimation paramétrique des valeurs prédites ainsi que l'écart type constante.
- L'estimation paramétrique des valeurs prédites ainsi que l'écart type non constante.
- L'estimation non paramétrique des valeurs prédites ainsi que l'écart type non constante.

Toutes les implémentations des réseaux neuronaux pour ce projet sont en pyTorch.

## II. LA PRÉVISION DES COEFFICIENTS DE LA RÉGRESSION LINÉAIRE AVEC UNE VARIANCE CONSTANTE

Dans la première approche, nous supposons que l'erreur type est constante tout au long de la série temporelle. Pour cela, nous maintenons une architecture de réseau simple, en mettant l'accent sur la conception de la fonction de coût. Nous utilisons la fonction de vraisemblance maximale pour une distribution gaussienne comme fonction de coût. Cette approche est justifiée par l'hypothèse classique en régression linéaire, selon laquelle les résidus suivent une loi normale. En conséquence, les valeurs observées peuvent être considérées

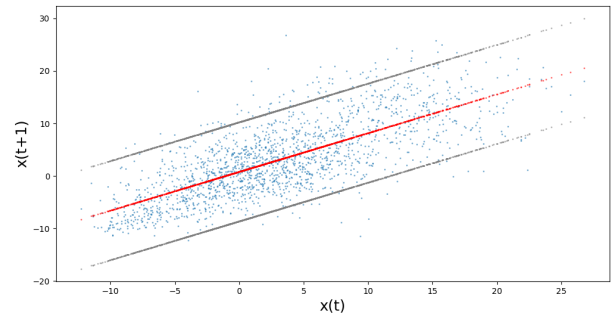
comme tirées d'une distribution gaussienne dont la moyenne correspond aux prédictions du modèle et dont l'écart type est constant. Cette hypothèse permet d'exprimer la probabilité des observations données par le modèle comme suit :

$$P(x_t | \hat{x}_t, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_t - \hat{x}_t)^2}{2\sigma^2}\right)$$

où  $x_t$  est la valeur observée,  $\hat{x}_t$  est la valeur prédite, et  $\sigma$  est l'écart type. Nous voyons clairement que l'écart type ne dépend pas du temps. En prenant le logarithme négatif de cette probabilité pour une série temporelle entière, nous obtenons la fonction de coût utilisée pour entraîner le réseau :

$$\begin{aligned} \mathcal{L} &= \prod_{i=1}^n f(x_i | \hat{x}_i, \sigma^2) \\ &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_i - \hat{x}_i)^2}{2\sigma^2}\right) \\ &= -\frac{n}{2} \ln(2\pi\sigma^2) - \frac{1}{2} \sum_{i=1}^n \frac{(x_i - \hat{x}_i)^2}{\sigma^2} \end{aligned}$$

Pour implémenter le réseau, nous avons utilisé un modèle linéaire avec seulement une entrée et une sortie. Dans cet exemple, nous avons implémenté notre fonction de coût pour inclure la variance comme paramètre de l'entraînement. L'entraînement fournit les coefficients de la ligne droite ainsi que la valeur constante de la variance. Le tracé ci-dessus montre les résultats. La ligne rouge représente la ligne droite, et les lignes grises sont calculées comme la ligne droite plus ou moins  $1.96\sigma$ , respectivement.

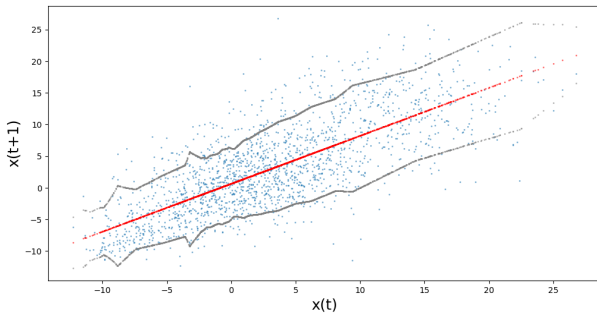


### III. LA PRÉVISION DES COEFFICIENTS DE LA RÉGRESSION LINÉAIRE AVEC UNE VARIANCE NON CONSTANTE

Nous allons maintenant implémenter un modèle visant à obtenir une estimation de la fonction de variance ainsi que les coefficients de la ligne droite. Cependant, il est nécessaire de réécrire la fonction de perte pour l'adapter à ce cas spécifique. En effet, la manipulation de la fonction de vraisemblance est plus complexe et nécessite certains ajustements, car la variance dépend du temps :

$$\begin{aligned}\mathcal{L} &= \prod_{i=1}^n f(x_i | \hat{x}_i, \sigma^2(i)) \\ &= \prod_{i=1}^n \frac{1}{\sqrt{2\pi\sigma(i)^2}} \exp\left(-\frac{(x_i - \hat{x}_i)^2}{2\sigma(i)^2}\right) \\ &= -\sum_{i=1}^n \frac{1}{2} \ln(2\pi\sigma(i)^2) - \frac{1}{2} \sum_{i=1}^n \frac{(x_i - \hat{x}_i)^2}{\sigma(i)^2}\end{aligned}$$

Puisque  $\sigma$  dépend du temps, nous ne pouvons pas le prendre en dehors de la somme pour la réduire en constante. Par conséquent, il y a deux sommes maintenant dans la fonction de perte. Dans cette approche, nous avons utilisé un modèle linéaire pour les coefficients de la ligne droite. Pour estimer la variance, nous avons utilisé un MLP à deux couches. La première couche a une entrée de taille 1 et une sortie de taille 256, tandis que la deuxième couche a une entrée de taille 256 et une sortie de taille 1. La fonction d'activation ReLU est appliquée après chaque couche. Ainsi, la variance n'est plus un paramètre de la fonction de coût. L'entraînement produit les coefficients de la ligne droite ainsi qu'une série de valeurs représentant la variance en fonction du temps. Le tracé ci-dessus montre les résultats. Nous observons que la variance n'est pas constante; cependant, les lignes représentant la variance en gris ne reflètent pas fidèlement la variance réelle. En particulier, dans certaines régions où les données s'éloignent davantage de la moyenne, les lignes de variance ne s'étendent pas en conséquence.



### IV. L'ESTIMATION DES QUANTILES

Dans cette approche, nous visons à estimer la variance de manière non paramétrique. Contrairement aux démarches précédentes, nous mettons de côté l'objectif d'estimer une droite de régression. À la place, étant donné que l'estimation

de la droite de régression nécessite une approche paramétrique, nous nous concentrons sur l'estimation de plusieurs quantiles, notamment le 50<sup>e</sup> quantile (médiane) ainsi que le 95<sup>e</sup> quantile. Pour ce faire, nous utilisons la fonction de perte pinball, qui est présentée ci-dessous. Cette fonction permet d'estimer directement les quantiles en minimisant une perte asymétrique, ajustée en fonction du quantile cible.

$$\begin{aligned}L_\tau(y, z) &= (y - z)\tau & \text{si } y \geq z \\ &= (z - y)(1 - \tau) & \text{si } z > y\end{aligned}$$

$y$  représente la valeur réelle,  $z$  la quantile prédite, et  $\tau$  le quantile ciblé. Pour mieux comprendre le rôle de  $\tau$  dans cette fonction de coût, il est important de noter qu'en pratique, on calcule la somme de toutes les pertes associées aux valeurs  $y_i$  sur l'ensemble des données. Cette somme reflète le coût total et guide l'optimisation du modèle.

$$\begin{aligned}\text{Perte totale}(z) &= \sum_{i=1}^n L_\tau(y_i, z) \\ &= \tau \sum_{i: y_i \geq z} (y_i - z) + (1 - \tau) \sum_{i: y_i < z} (z - y_i)\end{aligned}$$

On observe ici que  $\tau$  joue un rôle central en pondérant différemment les erreurs selon leur position par rapport à  $z$ . Les erreurs situées au-dessus de  $z$  sont pénalisées par un facteur de  $\tau$ , tandis que celles situées en dessous de  $z$  sont pondérées par un facteur de  $1 - \tau$ . Par exemple, lorsque  $\tau = 0,95$ , les valeurs de  $y_i$  inférieures à  $z$  sont associées à un coût plus élevé, ce qui pousse l'optimisation à augmenter la valeur de  $z$  afin de réduire la perte totale. Cela reflète notre intuition : pour estimer correctement la quantile 95<sup>e</sup>,  $z$  doit être ajusté de manière à se situer parmi les valeurs les plus élevées de l'ensemble de données. Dans ce projet, nous avons estimé le 5<sup>e</sup>, le 50<sup>e</sup>, et le 95<sup>e</sup> quantiles. Pour ce faire, le réseau implémenté utilise un MLP à deux couches pour chaque quantile. La première couche a une entrée de taille 1 et une sortie de taille 64. La deuxième couche a une entrée de taille 64 et une sortie de taille 1. La fonction d'activation softmax est appliquée entre les deux couches. L'entraînement produit trois séries de valeurs, chacune représentant les valeurs du quantile donné en fonction du temps. Le tracé ci-dessus montre les résultats. Nous observons que les lignes des quantiles suivent bien la forme attendue. Par exemple, lorsque les données s'éloignent du centre, les quantiles s'étendent. Il semble également que la médiane se situe au milieu des données.

