



ÉCOLE CENTRALE NANTES

INFO IA - PIIA MARÉES
RAPPORT

Rapport 4 - PIIA1 Marées

Élèves :

Daniel MACEDO GALEMBECK
Nicolas CONTRERAS
Alexandre THOMASSIN
Liushuangfei XIE

Enseignant :
Mr. ROUX

17 février 2023

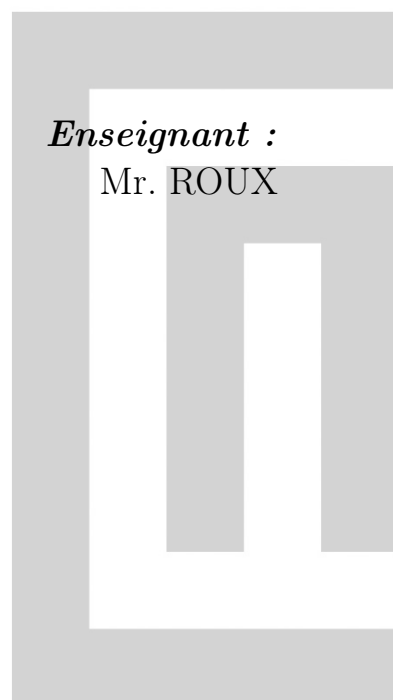


Table des matières

1	Rapport des deux semaines passées	2
1.1	PyTorch :	2
1.1.1	Données utilisées :	2
1.1.2	Modèle :	3
1.1.3	Entraînement :	3
1.1.4	Prédictions sur les données connues :	4
1.2	Amélioration des résultats à travers d'amélioration de les données	4
2	Pour les deux semaines à venir	6
2.1	Amélioration des résultats à travers d'amélioration de les données	6

1 Rapport des deux semaines passées

1.1 PyTorch :

Lors de ces deux dernières semaines, nous avons essayé de reproduire ce nous vous fais sur TensorFlow avec PyTorch.

Pour cela nous avons utilisé le module Forecasting de PyTorch, ce module apporte des méthodes pour prédire plus facilement des données périodiques.

1.1.1 Données utilisées :

Pour cela nous allons utilisé les données des années 2016 à 2022, en ne prenant que les données de Source 3.

De plus, on ajoute un index de Temps, utilisé par la librairie et nous permettant de rapporter les données à leurs temps réel. Il est préférable d'enlever la date car celle-ci n'est pas nécessaire et elle augmente significativement le nombre de paramètre du modèle.

Ce sont des données prises toutes les 10 minutes.

	Valeur	Source	Time
1	3.607	3	0
13	3.496	3	1
25	3.446	3	2
37	3.298	3	3
49	3.229	3	4
...
4034625	1.931	3	338248
4034637	1.915	3	338249
4034649	1.918	3	338250
4034661	1.942	3	338251
4034673	1.998	3	338252

FIGURE 1 – Données utilisées avec PyTorch

1.1.2 Modèle :

On définit le modèle de telle sorte avec PyTorch Forecasting :

```
1 tft = TemporalFusionTransformer.from_dataset(  
2     training,  
3     learning_rate=0.03,  
4     hidden_size=8,  
5     attention_head_size=1,  
6     dropout=0.1,  
7     hidden_continuous_size=4,  
8     output_size=7,  
9     loss=QuantileLoss(),  
10    log_interval=1,  
11    reduce_on_plateau_patience=4  
12 )
```

On obtient un modèle avec 4,6k paramètres.

1.1.3 Entraînement :

On entraîne ensuite le modèle sur des blocs de données de longueur 100 avec une prédiction de 10 données dans le futur. (Ce sont les même taille de bloc que lors de l'étude de Tensorflow afin de pouvoir comparer les résultats)

On remarque que l'entraînement est assez rapide : 5min pour 10 epochs A la fin de l'entraînement on obtient une loss de 0.05.

1.1.4 Prédictions sur les données connues :

On peut afficher les prédictions sur 3 exemples différents :

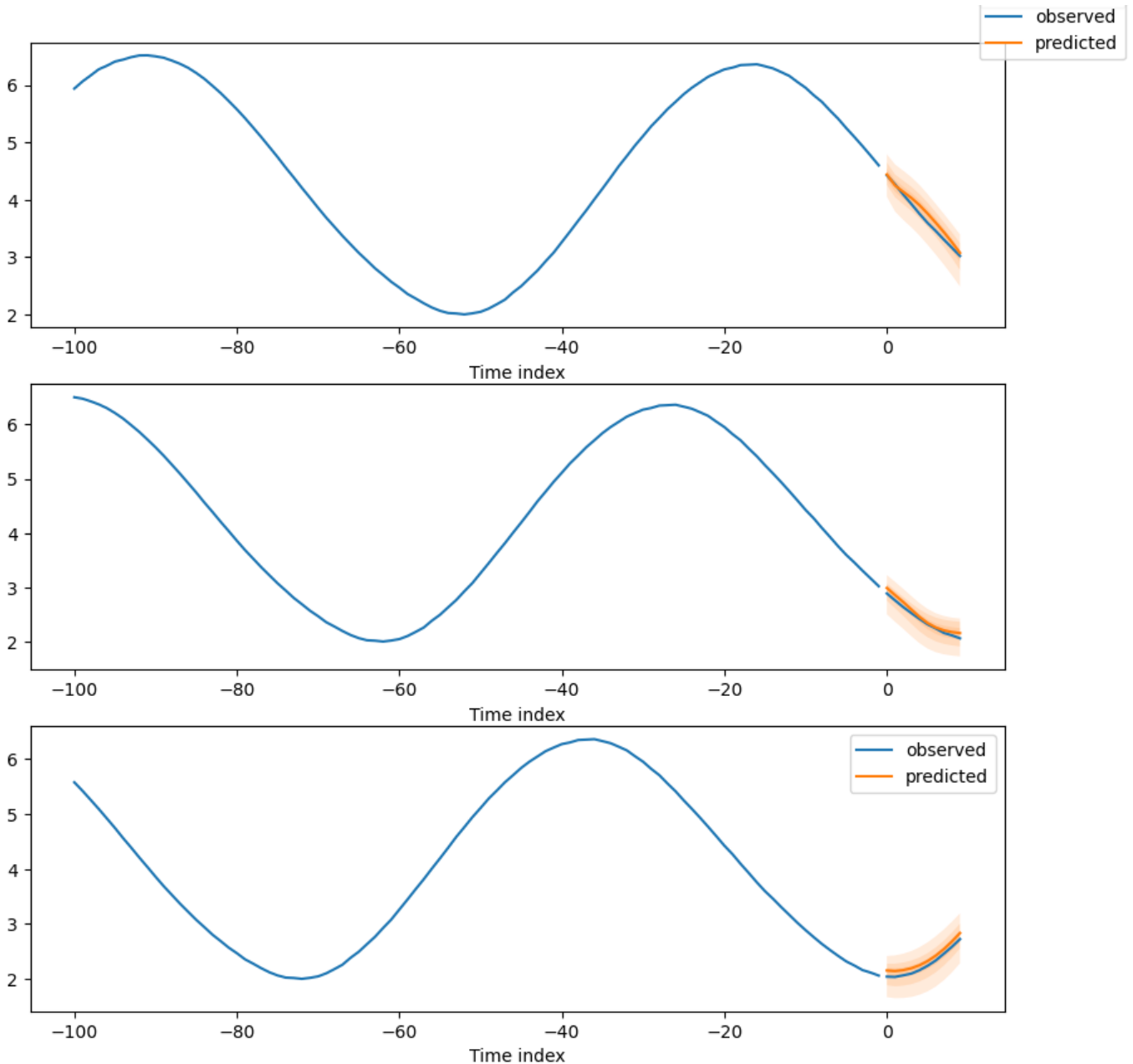


FIGURE 2 – Prédictions PyTorch

Les modèles sont quasiment aussi précis que ceux de Tensorflow cependant l'entraînement avec PyTorch est bien plus rapide.

1.2 Amélioration des résultats à travers d'amélioration de les données

Suite aux observations remarqués lors du dernier rapport - que la plupart des années ont des données manquantes ou de sources différentes (et, alors, avec qualités différentes) nous avons pensé qu'au

lieu d'essayer d'implémenter nouveaux modèles plus complexes et, théoriquement mieux, d'essayer d'améliorer les données de base.

Il y a plusieurs façons de faire cette approche, nous avons choisi de remplir les données qu'on n'a pas avec la moyenne mobile (rolling average) de les dernières mesures (7, 30 ou 200). Mais, nous faisons cette remplissage seulement avec les données réels, ça veut dire, si on a une période plus grande que 7, 30 ou 200 heures l'horaire reste sans valeurs. Nous avons pris cette décision parce que sinon nous générerions de données basés sur autres données que ne sont pas tout à fait réelles.

Avec la nouvelle serie temporelle, nous avons utilisé un modèle présenté en le dernier rapport (le SVM) et nous pouvons comparer les résultats si dessous :

	MSE	RMSE	R²	MAE	MedAE	MaxErr
0	0.134127	0.366233	0.943784	0.161769	0.065375	2.771175

FIGURE 3 – Résultats avant remplissage de données

	MSE	RMSE	R²	MAE	MedAE	MaxErr
0	0.019605	0.140018	0.991723	0.075189	0.051246	1.719642

FIGURE 4 – Résultats avant remplissage avec moyenne mobile de 7 périodes

	MSE	RMSE	R²	MAE	MedAE	MaxErr
0	0.032447	0.18013	0.986262	0.086726	0.053921	2.207363

FIGURE 5 – Résultats avant remplissage avec moyenne mobile de 30 périodes

	MSE	RMSE	R²	MAE	MedAE	MaxErr
0	0.03731	0.193157	0.984204	0.088998	0.052994	2.361699

FIGURE 6 – Résultats avant remplissage avec moyenne mobile de 200 périodes

Nous pouvons voir qu'il avait une amélioration dans tous le cas, mais principalement avec 7 périodes, nous pensons que c'est parce que comme les données ont une allure sinusoïdal, les moyennes plus grandes n'ont pas la capacité de prendre en compte ce mouvement.

2 Pour les deux semaines à venir

2.1 Amélioration des résultats à travers d'amélioration de les données

En suite aux résultats prometteurs démontrés dans la section 1.2, nous avons comme but explorer ce méthode là, en utilisant avec autres périodes temporelles, ainsi que avec modelés différents. En plus, un $R^2 > 0.99$ peut-être un cas de *overfitting*, donc, nous avons comme but aussi explorer autres méthodes pour mesurer l'erreur pour bien vérifier que les résultats sont vraiment meilleurs.