



ÉCOLE CENTRALE NANTES

INFO IA - PIIA MARÉES
RAPPORT

Rapport 1 - PIIA1 Marées

Élèves :

Daniel MACEDO GALEMBECK
Nicolas CONTRERAS
Alexandre THOMASSIN
Liushuangfei XIE

Enseignant :
Mr. ROUX

17 février 2023

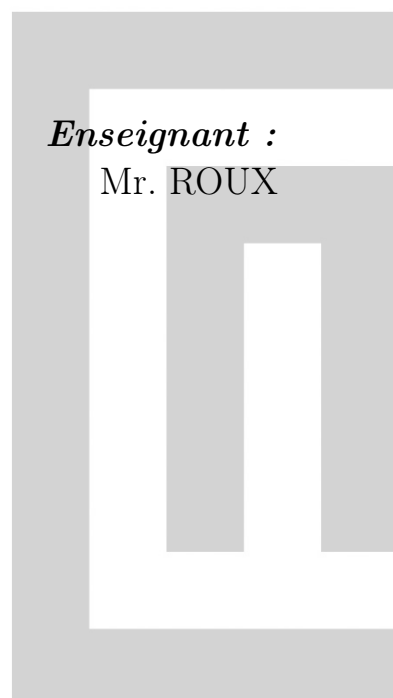


Table des matières

1	Rapport des deux semaines passées	2
1.1	Objectif final	2
1.2	État de l'art	2
1.2.1	Définitions sue IA	2
1.3	Exemples :	3
1.4	Données	4
2	Objectif des deux prochaines semaines	4
2.1	Étude du module TensorFlow	4
2.2	Mise en forme des données	4

1 Rapport des deux semaines passées

Ces deux premières semaines ont été consacré à la recherche d'exemples et de librairies pouvant aider à la réalisation du projet.

1.1 Objectif final

D'abord, on a décidé notre but final avec le projet d'une façon plus explicite : avoir un modèle qui est capable de, ayant une date dans le futur ou passe, nous retourne les horaires de la marée en ce jour.

1.2 État de l'art

Une recherche sur les concepts et définitions de l'intelligence artificielle actuelle a été effectuée, définissant non seulement l'IA elle-même, mais les 4 niveaux de complexité qu'elle peut avoir.

Comme le moyen le plus courant de produire une IA est l'apprentissage automatique, nous avons étudié les principales méthodes de construction de modèles d'apprentissage automatique.

Un résumé de cette recherche est représenté ci-dessous

1.2.1 Définitions sue IA

IA : Mettre en œuvre un certain nombre de techniques visant à permettre aux machines d'imiter une forme d'intelligence réelle (humaine).

1. **Machines Réactives :** Ces machines n'ont pas la capacité d'apprendre, ça veut dire, elles ne sont pas capables d' utiliser les expériences antérieures pour prendre des nouvelles décisions. Elles peuvent donner des sorties basées en un nombre limité d' entrées.
Ex. : DeepBlue, IA qui a gagnée contre Gary Kasparov aux échecs.
2. **Mémoire limitée :** En plus de faire ce que les machines réactives font, elles sont capables d' utiliser des données historiques pour améliorer les décisions - donc, de gros volumes sont utilisés pour entraîner les modèles.
Ex. : IA's de reconnaissance d'images.
3. **Théorie de l'esprit :** Ici on arrive dans le domaine théorique, il n'y a pas de modèles réels qui sont autant développés. Ici, en plus de la capacité d'apprendre, l'IA commence à comprendre les entités avec qui elle interagit, en détectant les émotions et désirs, par exemple.
Ex. : Assistance personnelle (comme Siri) peut voir en la voix d'utilisateur qu'il est énervé et essaie de le tranquilliser.
4. **Superintelligence artificielle :** Encore très loin de notre réalité, la SIA serait comme un superhuman, avec une capacité d'apprentissage dans quelque domaine et une mémoire incroyables. Les conséquences sont inconnues.

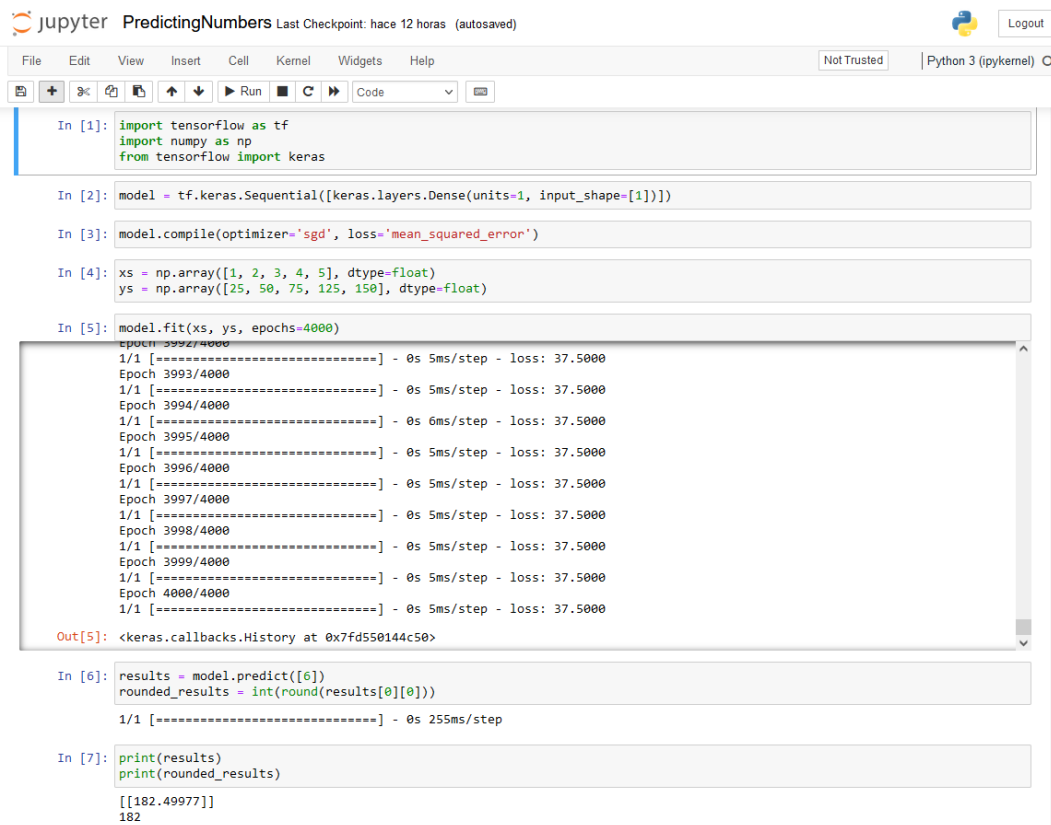
1.3 Exemples :

Nous avons cherché différents exemple de MachineLearning.
En effet, nous avons trouvé ce tutoriel de la librairie TensorFlow (https://www.tensorflow.org/tutorials/structured_data/time_series) dans lequel le but est de prédire des données météo a partir de données déjà récolté.

Ce tutoriel nous apporte plein d'informations sur :

- La préparation des données en vu de leurs utilisation
- La construction d'un modèle d'apprentissage automatique
- La répartition des données sur la phase d'entraînement et celle de test

Après avoir regarder ce tutoriel nous avons essayé d'implémenter un apprentissage très simple avec TensorFlow afin d'essayer cette librairie.



```

In [1]: import tensorflow as tf
import numpy as np
from tensorflow import keras

In [2]: model = tf.keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])

In [3]: model.compile(optimizer='sgd', loss='mean_squared_error')

In [4]: xs = np.array([1, 2, 3, 4, 5], dtype=float)
ys = np.array([25, 50, 75, 125, 150], dtype=float)

In [5]: model.fit(xs, ys, epochs=4000)

Out[5]: <keras.callbacks.History at 0x7fd550144c50>

In [6]: results = model.predict([6])
rounded_results = int(round(results[0][0]))

In [7]: print(results)
print(rounded_results)

[[182.49977]]
182

```

FIGURE 1 – Mise en œuvre du premier programme

Nous avons aussi trouvé une librairie appelée PyTides donnant les hauteurs de marrées à la date demandé, cependant après vérification cette librairie s'appuie sur des équations physiques (position du Soleil, de la Lune, ...), elle n'est donc pas intéressante dans la réalisation du projet.

1.4 Données

Après avoir analysé le jeu de données de marégraphie, nous avons constaté ce qui suit :

- Hauteurs de mer, chaque année pendant 166 ans (1846 à 2022).
 - dans les archives de 2022, seulement jusqu'au 4 août.
- Il existe des données par heure, par dix minutes et par minute.
- Il n'y a pas de données pour 1858, 1859, 1915 et les années entre]1944, 1953[.
- Les fichiers **3_1996-1.txt** et **3_1996-2.txt** ne sont pas nécessaires. On doit plutôt utiliser le fichier **3_1996.txt**.
- Il n'existe pas de données datées sur les phases de la lune.
- Toutes les données n'ont pas la même **source**.
- L'ensemble des fichiers fait 271 Mo (166 fichiers).

2 Objectif des deux prochaines semaines

2.1 Étude du module TensorFlow

Bien que nous ayons maintenant une idée un peu plus claire des types de modèles que nous pouvons mettre en œuvre, nous ne comprenons pas encore complètement comment nous pouvons le faire. Ainsi, d'après les exemples que nous avons trouvés, nous avons choisi la bibliothèque TensorFlow de Python - qui est souvent utilisée à cette fin.

L'un des objectifs des deux prochaines semaines est donc que chacun d'entre nous essaie de se familiariser avec la bibliothèque, en recherchant et en suivant des tutoriels, en lisant des articles, mais surtout en essayant de mettre en pratique les outils de la bibliothèque.

Après cette période et cette étude, nous serons idéalement en mesure de définir les nouveaux objectifs en termes de répartition des données pour l'entraînement et les tests, les tests que nous voudrions effectuer (pour des dates dans le futur ? à quelle distance dans le futur ? dans le passé ? utiliser les anciennes données pour l'entraînement et voir si elles ont la même qualité que les données actuelles ?). Ainsi que le(s) type(s) de technique(s) de ML à utiliser et la manière de les mettre en œuvre.

2.2 Mise en forme des données

Comme nous l'avons compris à travers les différentes recherches que nous avons pu faire ces deux semaines, il est essentiel de préparer ses données afin qu'elle soit utilisable par le programme.

Par conséquent, nous avons pensé qu'il serait une bonne idée de transformer les données en fichiers .csv, d'enlever la tête de chaque fichier et de convertir les temps en secondes (format unix). Nous avons aussi pensé à regrouper les fichiers par tranche de 10 ans ou 20 ans, afin de ne donner qu'un fichier de données au programme lors de l'entraînement.

L'objectif est donc d'écrire un script python afin de faire ce formatage.