Contenu

LISA : Laser Interferemoter Space Antenna est un observatoire spatial dont le lancement est prévu en 2034 qui permettra la détection directe d'ondes gravitationnelles indétectables par les interféromètres au sol tels que LIRGO ,VIRGO.

LISA doit être capable de détecter un grand nombre de signaux(onde gravitationnelle) provenant d'environs 60 millions de système binaires galactiques.

Objectif : caractériser ces systèmes binaires galactiques

Pour atteindre le but , **il s'agit simplement d'estimer les paramètres des systèmes binaires galactiques en étudiant les ondes gravitationnelles produites.**

Dans la suite de mon travail, j'utiliserai des méthodes d'apprentissage automatique en particulier les réseaux de neurones pour l'estimation de ces paramètres binaires galactiques.

Pour faire de l'apprentissage automatique, nous avons besoin d'une base d'exemple pour l'entrainement de notre modèle. Ainsi, les données qui m'ont été fournir sont celles du simulateur lisacode.

Nous appellerons onde gravitationnelle par forme d'onde et nous les noterons simplement FO.

Nous allons implémenter deux types de réseaux de neurones à savoir le perceptron multi-couches (MLP) et les réseaux de neurones convolutifs (CNN)

1. Les données
    a. Base d'apprentissage

Un élément essentiel de l'apprentissage automatique est de disposer de données d'apprentissage nombreuses et de qualité.

Notre base d'exemple est constitué de N=40 0000 FO observé sur 256 fréquences dans 3 canaux (X,Y,Z) .
Concrètement, une donnée est constituée d'une **forme d'onde**, de **256 fréquences** et d'un **canal**.

    b. Traitement des données

Pour une meilleure estimation des paramètres , nous devons faire des traitement préalable sur les données .

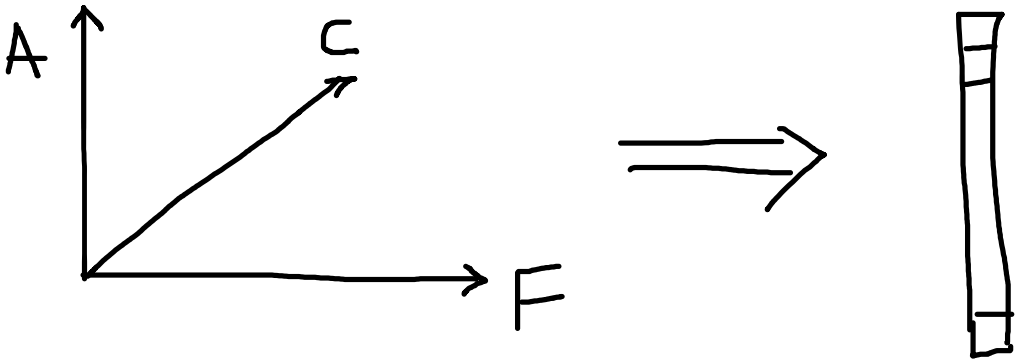
Cf : Pre_processing.ipynb

2. Split data

I. Le MLP

1. Les données d'entrées

En entrée du réseau de neurones, nous devons avoir un **vecteur.** Au départ, chaque FO est un tableau de 5*2827* (#c) (voir fichier pre_processing.ipynb) qu'il faut transformer en vecteur de taille **14 135*(#c).**



A : désignant l'ensemble des 5 sous FO générer aléatoirement suivant une loi uniforme de la forme d'onde i.
F : l'ensemble des fréquences allant de 1 à 2827
C : le type de canal d'observation de 1 à 3.

Ainsi une entrée X est un vecteur de taille **14 135*(#c)..**

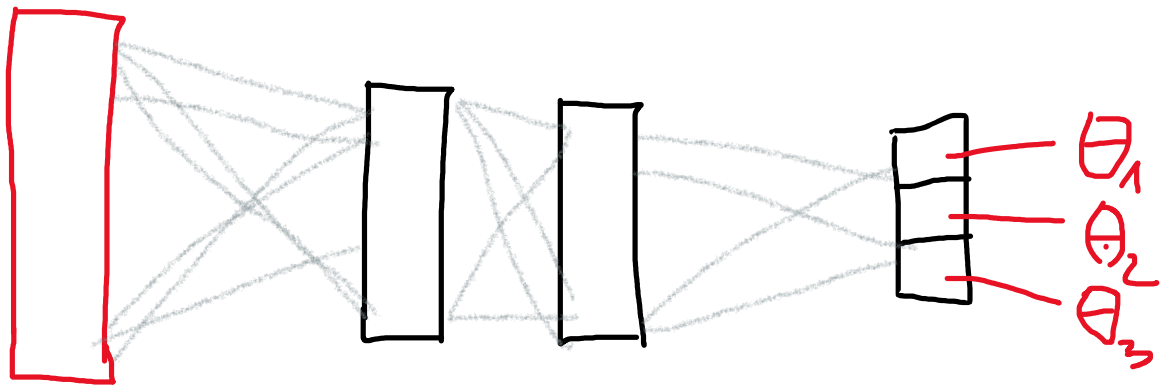
2. Le réseau

On va construire un réseau de neurones qui produira une fonction F : **R** (14 135*(#c)) vers **R**(3) [#3 le nombre de paramètres à estimer]. L'architecture est composée de **trois couches** En entrée , nous avons un vecteur de taille **14 135*(#c)** .

- La première couche (cachée) est constituée de p neurones
- La deuxième couche cachée constituée également de p neurones
- Les deux couches cachées auront pour fonction d'activation la fonction **RELU**
- La couche de sortie est composée de 3 neurones avec la fonction d'activation **linéaire**

Nous cherchons donc la fonction F elle que $F(X_i) \simeq Y_i$ pour nos données transformées (X_i, Y_i), $i = 1,...,40\,000$.

X_i *:* désigne ici nos données les FO (le tableau numpy *data*) et Y_i *:* les paramètres associés (tableau numpy *params*)



3. Programmation

Cf : fichier mlp_with_regressor.ipynb