

TP8note

March 24, 2021

1 TP noté 25/03/2021

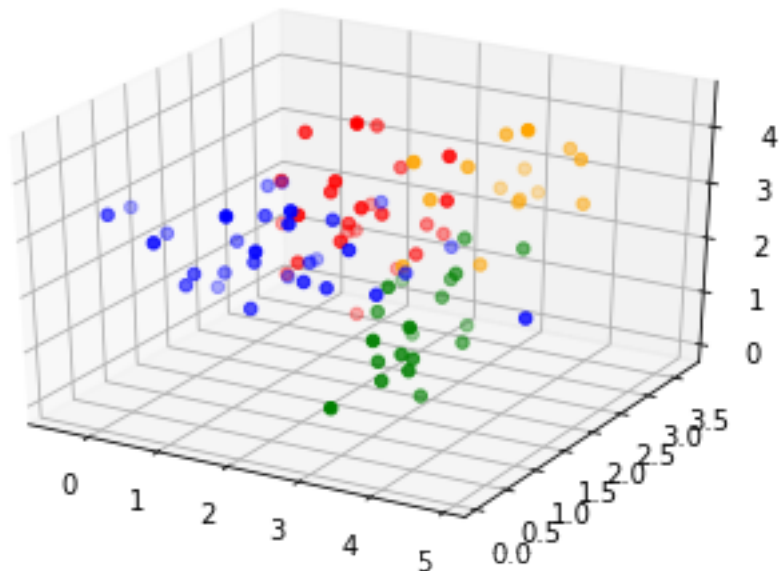
L'objectif de ce TP est de proposer un classifieur linéaire pour les données accessibles dans le fichier *data.py*. Pour accéder aux données, il suffit de recopier ce fichier dans votre répertoire de travail et d'écrire au début de votre programme:

```
[ ]: from data import *
```

L'algorithme de descente de gradient est disponible via le fichier *grad_desc.py*. Pour avoir accès à l'algorithme, il suffit de recopier ce fichier dans votre répertoire de travail et de l'importer au début d'un programme:

```
[ ]: from grad_desc import *
```

```
[1]:
```



Les données sont accessibles à partir des arrays numpy *data['input']* et *data['target']*. le vecteur input est un vecteur de \mathbb{R}^3 , la valeur de target est un entier dans l'ensemble $\{0, 1, 2, 3\}$ (4 classes).

Plus précisément, il s'agit de prévoir la valeur de *target* à partir de *input*. On pourra s'inspirer de ce qui a été fait pour les jeux de données *iris* et *digits*.

Il s'agit donc de trouver une fonction de $\mathbb{R}^3 \mapsto \{0, 1, 2, 3\}$, concordante (dans la mesure du possible) avec les données.

2 Exercices

1. Programmez une fonction qui calcule le nombre d'erreurs commises par un classifieur linéaire.
2. Programmez une fonction qui calcule une loss pour ce classifieur (basée sur la log-vraisemblance et le softmax)
3. Appliquez la descente de gradient à cette loss.
4. Indiquez la solution trouvée, et le nombre d'erreurs commises par cette solution.

3 Précision:

Vous n'utiliserez pour trouver la solution que les fichiers disponibles, et la bibliothèque numpy.

En particulier, vous n'utiliserez pas de bibliothèque du type scikitlearn ou autre.