

# TP - Pointnet pour la classification de nuages de points

Master ID3D - Julie Digne

30 Novembre 2023

Dans ce TP nous verrons comment reconnaître des formes géométriques, définies par des nuages de points.

## 1 Librairies - logiciels

La programmation se fait en python, en utilisant les librairies suivantes

- Numpy
- Matplotlib (pour visualiser l'évolution des fonctions de coût)
- Pytorch (pour la partie neuronale)
- scikitlearn (pour la matrice de confusion si besoin)

Un environnement conda `tp_ml.yml` se trouve sur la page du cours.

## 2 Données

Un script `prepare_data.py` est fourni et génère des données de test et de train. Vous pouvez les visualiser avec matplotlib, le label 0 correspond à des cylindres, le label 1 correspond à des parallélépipèdes rectangles, le label 2 correspond à des tores. Cela crée un dossier data divisé en train et test.

## 3 Architecture

Nous avons besoin de 2 réseaux un 'Tnet' et un 'pointnet'.

TNet qui va sortir une matrice de rotation :

- Entrée : nuage de points  $nD$  sous forme `batch_size x nD x 2048`
- 3 couches de **convolution 1D** de tailles 64, 128, 1024 suivies chacune de batchnorm puis relu
- un max par canal (`torch.max`)
- 2 couches linéaires de taille 512, 256, suivies chacune de batchnorm puis relu
- une couche linéaire passant de 256 à  $nD^2$

PointNet :

- Un Tnet avec  $nD=3$  (aligne les données)
- Appliquer la matrice à la donnée d'entrée (`torch.bmm`)
- convolution (64) + batchnorm +relu
- Tnet avec  $nD=64$  (aligne les features)
- convolution (128) + batchnorm +relu

- convolution (1024) + batchnorm +relu
- un max par canal (torch.max)
- 2 couches linéaires de taille 512, 256, suivies chacune de batchnorm puis relu
- une couche linéaire passant de 256 à 3 (3 classes) avec log\_softmax

La loss est la negative log likelihood (nll\_loss), et vous pouvez prendre une sgd avec un learning rate de  $1e - 3$

## 4 Tester sur les données test

Vous pouvez sortir les taux de vrais/faux positifs/négatifs, de bonne reconnaissance, ou la matrice de confusion (scikitlearn)

## 5 Robustesse au bruit

Rajouter du bruit aux données et tester la robustesse de l'algorithme. On pourra aussi réentraîner avec du bruit aléatoire (augmentation de données).