TP - Pointnet for point cloud classification

Master ID3D - Julie Digne

21 Novembre 2024

The goal of this lab is to recognize geometric shapes defined by point sets.

1 Libraries

- Numpy
- Matplotlib (for loss visualization)
- Pytorch
- scikitlearn (for computing the confusion matrix)

A conda environment tp_ml.yml is available on the course webpage.

2 Data

A prepare_data.py script is provided and generates test and train data. You can visualize them with matplotlib, label 0 corresponds to cylinders, label 1 corresponds to rectangular parallelepipeds, label 2 corresponds to tori . This creates a data folder divided into train and test.

3 Architecture

We need two networks a 'Tnet' and a 'pointnet'.

TNet outputs a matrix of rotation:

- Input: point set nD in the form: batch_sizexnDx2048
- 3 layers of convolution 1D with size 64, 128, 1024 followed by batchnorm and relu
- max per canal (torch.max)
- 2 linear layers of sizes 512, 256, followed each by batchnorm and relu
- A linear layer going from 256 to nD*nD

PointNet:

- Tnet with nD=3 (for data alignment)

- Apply the rotation to the input data(torch.bmm)
- convolution (64) + batchnorm +relu
- Tnet with nD=64 (features alignment)
- convolution (128) + batchnorm +relu
- convolution (1024) + batchnorm +relu
- A max per canal (torch.max)
- 2 linear layers of sizes 512, 256, followed by batchnorm then relu
- a linear layer going from 256 to 3 (3 classes) with log_softmax

The loss is the negative log likelihood (nll_loss), the optimizer is a SGD with a learning rate of $1e-3$

4 Testing

Evaluate the number of true/false positive and negative or the matrix of confusion (scikitlearn).

5 Robustness to noise

Add noise to data and test the robustness of the detection. A possibility is also to retrain with additional random noise (data augmentation).