3D Data Processing - $4^{th} Assignement$

Luca Sambin

1 Problem Description

The goal of this assignment was to complete the missing parts on the provided Colab notebook, that design a modified PointNet architecture that is able to extract 3D descriptors to be used for matching.

So basically, the goal of this assignment is to design a reduced version of the PointNet architecture **"TinyPointNet"**) that learns a 3D local feature descriptor from training data. The idea is to use the n-dimensional global feature learned by TinyPointNet directly as a descriptor of a locality of points (neighborhood of a point p), removing from PointNet the last MLP (multi-layer perceptron) used for classification tasks. Unlike the original PointNet where the global feature has dimensions **1024**, TinyPointNet could provide a lower dimensional (**256**) global feature. The structure of the TinyPointNet is shown in the figure below.

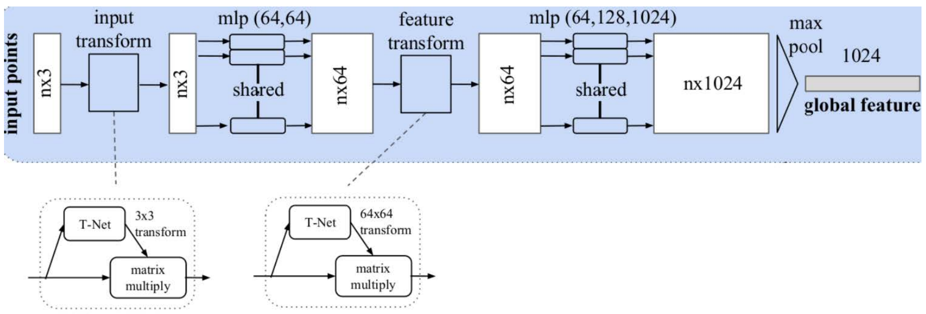

Figure 1: TinyPointNet

2 Solution Structure

To reach the goal it was necessary to complete three part of the notebook:

- ***Sample generation:*** Generates a set of positive and negative pairs of point sets. Each point set is composed of all the points included in a spherical support region with radius r (see picture below) around some point. In order to generate an anchor and a positive and a negative sample, extract a random point (anchor) and its neighborhood from the starting point cloud. The closest point of the noisy point cloud, will be selected as a positive sample, along with its neighborhood, while the negative sample will be extracted between points far from the anchor. Also normalization is performed, basically subtracting the coordinates of the three selected points (anchor, positive and negative sample) from their corresponding sets.

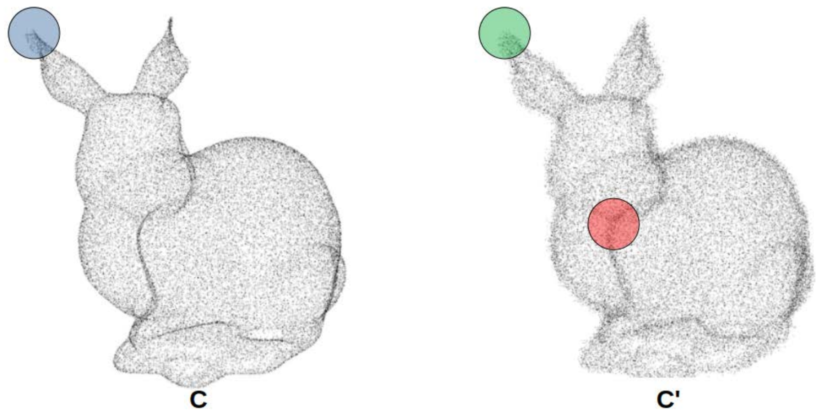

Figure 2: Sample generation

- ***Design the TinyPointNet architecture:*** follow the same structure of TinyPointNet (see the Figure 1). Finally, remember to set the dimension of the output feature to **256** instead of the original value 1024.
- ***Define the TinyPointNet loss function:*** using a Triplet loss L as a loss function. Check the figure below, where A is a reference neighborhood of points (called anchor), P is a neighborhood that represents a correct match with respect to A, while N is a neighborhood that represents a wrong match with respect to A:

$$\mathcal{L}\left(A,P,N\right) = \max(\| \mathrm{f}(A) - \mathrm{f}(P) \|_2 - \| \mathrm{f}(A) - \mathrm{f}(N) \|_2 + \alpha, 0)$$

Figure 3: Triplet Loss Function

3 Results

In this section is showed the results, which consist of matching accuracy in the test set, obtained after the training on the training set. I test the TinyPointNet descriptors with a new dataset (the test set).
The result shows a matching accuracy of **64%**. Of course, the accuracy is not so high, but this is totally expected since it was used a reduced-size train dataset, which doesn't allow to obtain high-accuracy results.
Furthermore, since the positive and the negative sample are selected randomly, the result may vary from time to time.
Finally, the figure below shows the trend of train and valid losses.

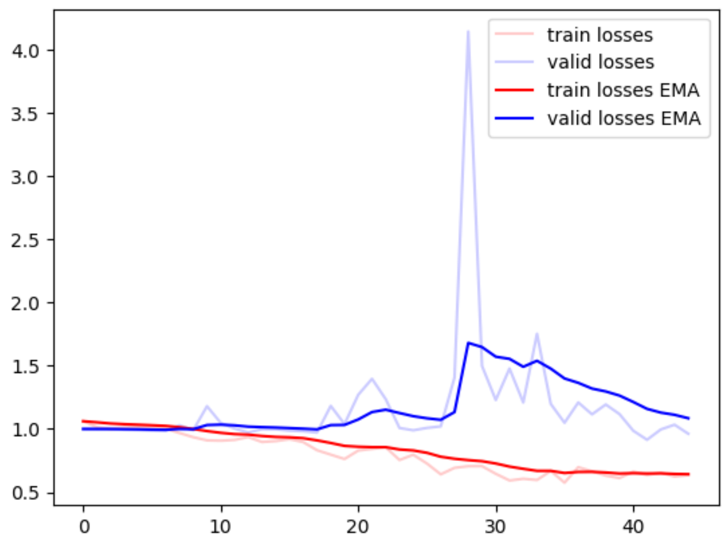

Figure 4: Train Graph