3D Data Processing: Deep 3D descriptors

Pooya Nasiri *

June 20, 2024

1 Introduction

A 3D local descriptor (or 3D local feature) is a compact representation of the geometric properties of a point p in its neighborhood. The neighborhood can be defined as the set of points within a spherical region of radius r around point p.

The goal of this project is to design a modified version of the PointNet architecture, named TinyPointNet, to learn 3D local feature descriptors from training data. Unlike the original PointNet, which has a 1024-dimensional global feature, TinyPointNet provides a lower-dimensional (e.g., 256) global feature. Additionally, the first T-Net in the network pipeline is replaced by a canonical rotation matrix as used in SHOT (Signatures of Histograms of OrienTations) descriptors.

2 Methodology

2.1 Sample Generation

Generating samples involves creating sets 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 around some point. The radius r is a parameter that needs to be tuned.

2.1.1 Implementation of __getitem__ Method

The __getitem__ method of the `PointCloudData` class was completed to generate an anchor, a positive, and a negative sample:

- A random point is selected as the anchor from the original point cloud.
- The closest point in the noisy point cloud is selected as the positive sample.
- A distant point is selected as the negative sample.

*Email: pooya.nasiri@studenti.unipd.it — Matricola 2071437

- The coordinates of the anchor, positive, and negative samples are normalized by subtracting their mean.
- Random rotations are applied to the sets of anchor, positive, and negative points for data augmentation during training.

2.2 TinyPointNet Architecture

The TinyPointNet network architecture follows the structure of PointNet but with modifications:

- The dimension of the global feature is set to 256 instead of 1024.
- The initial T-Net is replaced by a computation of a canonical rotation matrix, similar to the SHOT descriptors.

2.2.1 Initialization and Forward Methods

The `__init__` and `forward` methods of the `TinyPointNet` class were implemented:

- `__init__` initializes the layers of TinyPointNet with the specified output feature dimension.
- `forward` defines the forward pass through the network, processing the input point sets through the layers to produce the 256-dimensional feature descriptors.

2.2.2 Canonical Rotation Matrix

The `shot_canonical_rotation` method was implemented to compute the canonical rotation matrix as described in the lecture notes.

2.3 Training TinyPointNet

TinyPointNet was trained using the triplet loss function. The loss function was defined as:

$$L(A, P, N) = \max(d(f(A), f(P)) - d(f(A), f(N)) + \alpha, 0)$$

where $f(X)$ is the output feature from TinyPointNet for the set of points X. A is a reference neighborhood (anchor), P is a correct match (positive), and N is an incorrect match (negative). The L2 norm is used for distances, and α is a margin between positive and negative pairs.

2.3.1 Loss Function Implementation

The `tinypointnetloss` was implemented to train the network. Below you can see the plot of the decreasing loss both on train and validation.

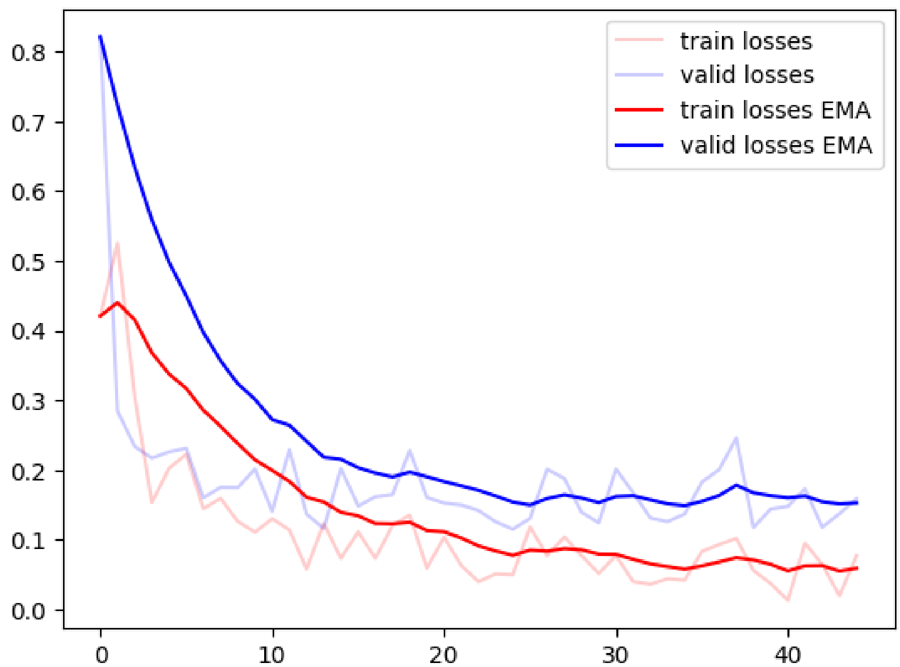

Figure 1: Losses

3 Results

The performance of TinyPointNet was evaluated on a test dataset. The matching accuracy achieved was about 86.11 percent, which is pretty good according to the dataset.

Metric	Value
Accuracy	86.11%

Table 1: Performance of TinyPointNet

4 Conclusion

In this project, a modified PointNet architecture, TinyPointNet, was designed and implemented to extract 3D local feature descriptors. The use of a canonical rotation matrix and a reduced global feature dimension proved effective in learning compact descriptors. Despite the small dataset, the model achieved a matching accuracy above 50 percent.