Deep Learning Based Point Cloud Transformer

Project Plan

03 October, 2021

Yuan Shuwen 3035638753

Supervisor: Dr. Dong Xu
1. Project Background
   1.1 Overview
   1.2 Tasks for 3D point cloud
   1.3 Transformer for Vision

2. Project Objective

3. Project Methodology
   3.1 Pytorch implementation of Transformer model
   3.2 Implementation of Pytorch PCT
   3.3 Classification and object segmentation

4. Project Timeline & Milestone
1. Project Background

1.1 Overview

3D point cloud is a common way to represent spatial geometric structures, and have a wide range of applications in computer vision and robotics. In recent years, the emergence and development of 3D sensors have made it increasingly convenient to obtain point cloud data, and the current strong interest in autonomous driving and intelligent robotics in academia and industry has highlighted the importance of point cloud processing and 3D scenes.

In the field of autonomous driving, 3D point cloud labelling also plays an important role. To ensure the proper driving of autonomous vehicles, Autonomous driving should first have a complete perception system to replace the driver's brain. As the degree of human intervention in autonomous driving becomes smaller and smaller, the accuracy, efficiency and comprehensiveness of the perception system to obtain information about itself and its surrounding environment become more and more demanding. Unlike 2D images, point clouds are disordered and unstructured, making it challenging to design neural networks to process them.

1.2 Tasks for 3D point cloud

There are basically four types of 3D point cloud tasks, which are object detection, object tracking, semantic segmentation, and adjustment.

- 3D point cloud object detection: locate and classify objects in a 3D point cloud by adding and fitting 3D cuboids around objects.
- 3D point cloud object tracking: add and fit 3D cuboids around objects to track their movement across a sequence of 3D point cloud frames. For example, track the movement of vehicles across multiple point cloud frames.

- 3D point cloud semantic segmentation – create a point-level semantic segmentation mask by painting objects in a 3D point cloud using different colors where each color is assigned to one of the classes you specify.

- 3D point cloud adjustment task types – Each of the task types above has an associated adjustment task type that can audit and adjust annotations generated from a 3D point cloud labeling job.

1.3 Transformer for vision

Transformer has been a great success in natural language processing and shows great potential in image processing. The output features of each word are related to all input features, so it is able to learn the global context. All operations of Transformer are parallelizable and sequentially independent. Theoretically, it can replace convolutional operations in convolutional neural networks with better generality. It has inherent permutation invariance in processing point sequences and is therefore well suited for point cloud learning. Inspired by the success of Transformer for vision and NLP tasks, based on the principles of traditional Transformer, the core idea of PCT is to avoid defining the order of point cloud data by exploiting one of the properties of the inherent order invariance of Transformer and to perform feature learning through an attention mechanism.
2. Project Objective

I want to realize the Pytorch implementation of PCT: Point Cloud Transformer, and will show how the 3D point cloud representation implemented by PCT can be applied to various tasks in 3D point cloud processing, including 3D point cloud object classification, object segmentation, and normal estimation.

3. Project Methodology

3.1 Pytorch implementation of Transformer model

The diagram below shows the overall structure of Transformer, which is mainly composed of two parts, Encoder encoder on the left and Decoder decoder on the right, and a complete task can be accomplished by such a structure. From the diagram, we can see that there are several basic units in the structure of Transformer, and the PyTorch deep learning framework will be applied to implement these basic units in the following. These include: Inputs Embedding, Mask, Self-Attention, Multi-Headed Attention, Feed-Forward Network, Layer Normalisation, etc.
3.2 Implementation of Pytorch PCT

The easiest way to modify the Transformer for point cloud use is to treat the entire point cloud as a sentence and each point as a word.

3.3 Classification and object segmentation

- Classification: Classifying the point cloud P into Nc object classes, we input the global features F into the classification decoder by two cascaded feedforward neural networks LBR, each with dropout probability 0.5, and finally a linear layer to predict the final classification score.
- Segmentation: To segment the point cloud into n parts (e.g., table tops, table legs), we must predict a part label for each point. First, connect the global features F and the pointwise features. To learn a generic model for various objects, we also encode a one-hot object category vector as 64-dimensional features and connect it to the global features. The architecture of the partitioning network is essentially the same as that of the classification network, except that the dropout is performed only on the first LBR, and we predict a point-by-point partitioning score, identifying the part label of a point also as the part label with the highest score.

4. Project Timeline & Milestone

| Oct 2022          | - Meeting with supervisor for ideation and inception  
|                  | - Brief literature review for understanding the current development of PCT |
| Nov 2022         | - Complete compiling the project plan and setting up the project web page  
|                  | - Further literature review, particularly on methodology of PCT & Transformer |
| Dec 2022         | - Study materials about deep learning  
<p>|                  | - Working on the Pytorch implementation of Transformer and Native PCT |</p>
<table>
<thead>
<tr>
<th>Month 2023</th>
<th>Tasks</th>
</tr>
</thead>
</table>
| Jan 2023  | - Prepare and deliver the first presentation  
|- Complete compiling the interim report  
 |- Deliver the preliminary implementation |
| Feb 2023  | - Implement the new design or route planning  
 |- Evaluate the new implementation or the route planning model |
| Mar 2023  | - Either seek to improve the model or explore route planning  
 |- Update the finalized project web pagea |
| Apr 2023  | - Complete compiling the final report  
 |- Deliver the finalized tested implementation  
 |- Prepare and deliver the final presentation |
Reference:


3. https://learnku.com/docs/cv-papers/1/pct-point-cloud-transformer/10601#:~:text=%E7%82%B9%E4%BA%91%E6%98%AF%E6%9F%90%E4%B8%AA,PCT%20%E7%94%9F%E6%88%90%E7%9A%84%E9%83%A8%E5%88%86%E5%89%B2%E3%80%82

