CS323: Deep Learning for Visual Computing
Reading Assignment 6

David Felipe Alvear Goyes
Student ID: 187594

12 May 2023

3D Deep Learning

This section of the assignment covers two key papers in the field of 3D Deep Learning - PointNet and DGCNN.

PointNet

- **Question 1:** How does PointNet handle the problem of permutation invariance in point cloud data, and what are the advantages of this approach?

  Answer: PointNet uses a set of 3D points as input to perform classification or segmentation. Traditional neural network architectures are capable to process ordered data, and PointNet offers a solution to process unordered data such as point cloud, without the need to convert them to another format such as voxel grids or multi-view images.

  PointNet has the de ability to permutation invariance which is the key solution for processing point cloud data. It addresses the problem by first processing independently each one of the points to compute the corresponding features, it uses a shared multilayer perceptron capable to learn a high dimensional point feature. Secondly, PointNet uses a symmetric function to produce the same result regardless of the order of their inputs, which is a key property for permutation invariance. The approach to achieve permutation invariance is to apply the symmetric function on the transformed elements of the individual points to obtain a global feature, this represents the entire point cloud and is invariant to the order of the points in the input, the PointNet can handle unordered data. In more detail, the network uses a simple max pooling operation for the symmetric function, to process the individual processed features from the shared MLP.

  The advantage of the mentioned approach is that Point Cloud can consume raw point cloud data without the need to perform voxelization or projections to 2D images. Additionally, the architecture can handle different point cloud data sizes, the shared MLP and max pooling operation can be applied to any input size. Point Cloud is less

sensitive to transformations and distortions for the property of permutation invariant. Lastly, the max pooling operation can capture the local structures to then compute a global feature representation of the input data.

- **Question 2:** Can you describe the architecture of PointNet and explain how it processes point cloud data?

  Answer: PointNet has a unique architecture to be able to process point cloud data. First, the input to the network is a set of points that represent 3D coordinates (x,y,z) and additional features such as color, or surface normal. The set of unordered data is passed to a spatial transformer network that is a mini point net network (T-Net), which applies an affine transformation to the point to align them in a standardized coordinate frame, the output of the T-Net is a matrix that is applied to the point cloud data. After the spatial transformation, each one of the transformed points is passed through a shared multi-layer perceptron (MLP) to learn a high-level representation of the points.

  The second part of the network starts with a feature transform using a T-Net, in which the output is a transformation matrix that is applied to the point features helping them to be invariant to transformations. These transformed feature points are passed to a shared MLP to learn higher-level features for each point. Finally, the architecture computes a global feature vector by applying a max pooling operation across all the points, this operation makes the network invariant to the points' input order.

  For the classification task, the global feature is passed through an MLP composed of fully connected layers to produce the output of K-label scores. For the segmentation task, the architecture performs a combination of local and global knowledge using the global feature vector and a previous point feature vector. This new feature point set containing global and local information is performed by two shared MLPs to obtain the segmented point cloud points.

- **Question 3:** How does the max pooling operation in PointNet help to aggregate information from multiple points, and how does it differ from other pooling operations in deep learning?

  Answer: Given that point cloud data are inherently unordered, the max pooling operation is important for aggregating the information from multiple points achieving the invariant permutation requirement. In detail, after the shared multi-layer perceptron extracts the features for each point independently, the max pooling operation is applied to aggregate the feature information by taking the maximum value of the features from each point, this result in a global feature vector that is the summary of the most significant features of the input.

  Max pooling operation differs from other pooling operations such as average pooling, and sum pooling, in the sense that it takes the most important (maximum) features from each point, whereas the other operations take the average or sum of the features for each point, that could miss important characteristics and not identify the most important features.

DGCNN

- **Question 1:** How does DGCNN handle the problem of non-uniform sampling in 3D point clouds, and what are the advantages of this approach?

  Answer: The Dynamic Graph Convolutional Neural Network addresses the problem of non-uniform sampling in 3D point clouds, introducing the definition of a dynamic graph. DGCNN constructs a K-nearest neighbors (K-NN) graph for each point to compute a local structure, it defines edges between the current and neighbor points. The K-NN graph is computed dynamically, which means that the edges between local points are computed on the fly for each point, then the graph can change during training as the point features are updated. The dynamic graph addresses the issue of non-uniform sampling and also adds sensitivity to the model to detect local structures in the data and capture dynamically the more relevant structures as the network learns.

  The advantage of the dynamic graph approach is that it allows the network to capture local structures allowing it to handle the non-uniform sampling problem and be more robust to noise and outliers. Additionally, the K-NN graph computation is computationally efficient compared with other methods that require pairwise comparison. Finally, DGCNN can learn features invariant to the order of the points paying attention to the local structure as well as the global structure of the point cloud data.

- **Question 2:** Can you describe the dynamic graph construction process in DGCNN, and how it captures local and global spatial relationships between points?

  Answer: Dynamic Graph Convolutional Neural Networks are designed to capture semantic characteristics over potentially long distances, using the local and global information of the input point cloud data. The process of dynamic graph construction starts with the K-nearest neighbors graph construction. For each point, the algorithm creates a graph where each point is connected to its k closest points generating edge features that describe the relationship between the point and the neighbors. The edge features can be spatial coordinates or in later layers high-dimensional vectors. The edge features are passed through a 1D convolution operation that gives a new set of feature points that takes into account the local structure of the neighbors and contains the local semantic information, the last is called Edge Convolution which can be described in equation 1. The edge convolution takes the edge feature and applies an edge function with an aggregation function to capture the global shape structure and the neighbor information (edges).

$$h_i = \max_{j \in \mathcal{N}(i)} \Phi(\mathbf{x}_i, \mathbf{x}_j), \tag{1}$$

$$\Phi(\mathbf{x}_i, \mathbf{x}_j) = \mathrm{MLP}(\mathbf{x}_i - \mathbf{x}_j, \mathbf{x}_i), \tag{2}$$

  Based on the output of the EdgeCon the point features are updated. Then, the K-NN graph can be recomputed on the new features allowing that dynamically the graph to be recomputed when the edge features are updated. Finally, after several edge

convolutions and graph update the network applies a global max pooling operation to aggregate the features from all points, similar to the PointNet implementation.

- **Question 3:** How does the dynamic graph pooling operation in DGCNN work, and how does it help to reduce the computational cost of processing large point clouds?

  Answer: Graph pooling operations are designed to reduce the size of the graph, and in the context of DGCNN are used to reduce the number of points while retaining the important features. DGCNN uses EdgeConv operations which dynamically update the edge features of the graph, with several iterations the edge features can be pooled to maintain the most important features reducing the number of K-NN computations and EdgeConv operations, which implies that having a reduced size graph will reduce the operations that can be performed, this implies the reduction of computational cost in processing large point clouds.