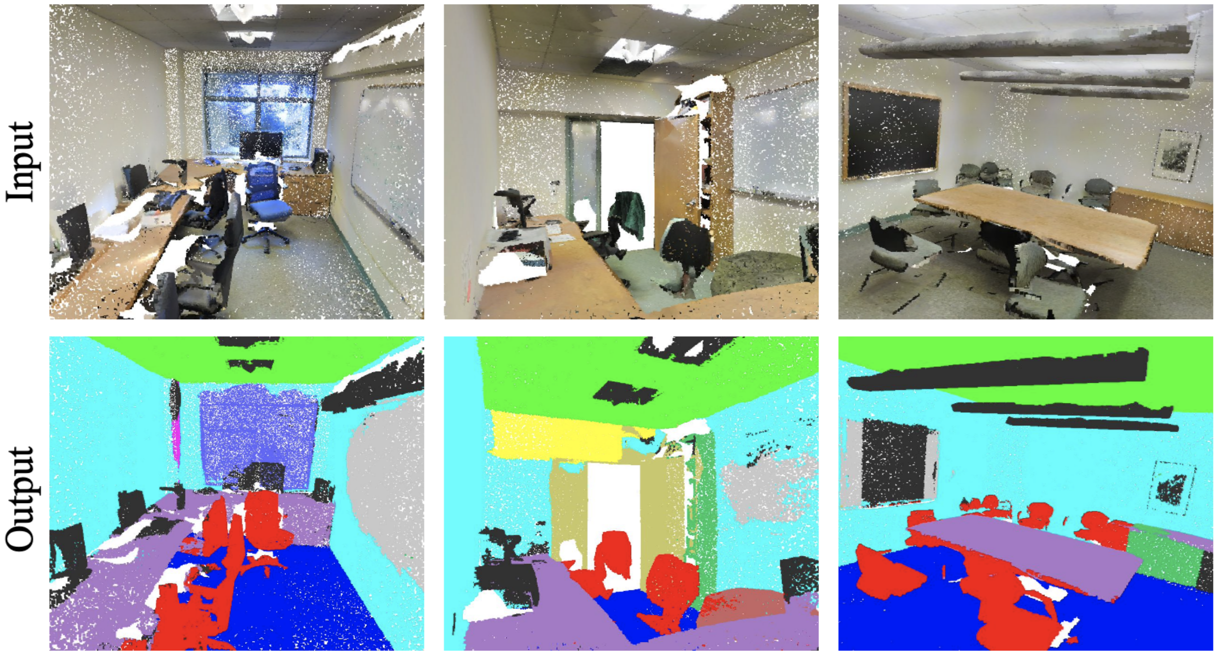
# PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

## Report

Vinayak Gupta 2nd July 2021

## Introduction

Usually, Point Cloud representations are converted to mesh or voxel representations because of their irregularity. But converting them into mesh or voxels causes unnecessary addition of data and causes issues. This paper presents a novel architecture that takes in the point cloud directly. This model allows for object classification, part segmentation, to scene semantic parsing.

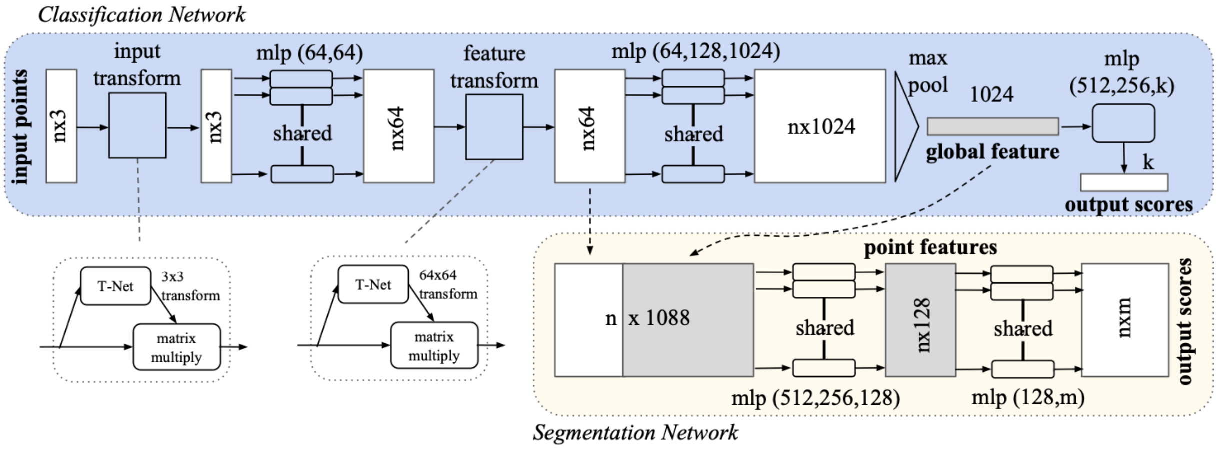


## Invariance of the model

We need to make sure that the model is not affected by the variance of the point cloud. Since in a point cloud, the points are spread irregularly it is not possible to take the points in order. Hence, if the points are taken in any order for input, the output prediction should always be the same no matter the order of points taken. Also, the model should take care of noise/missing data in the point cloud.

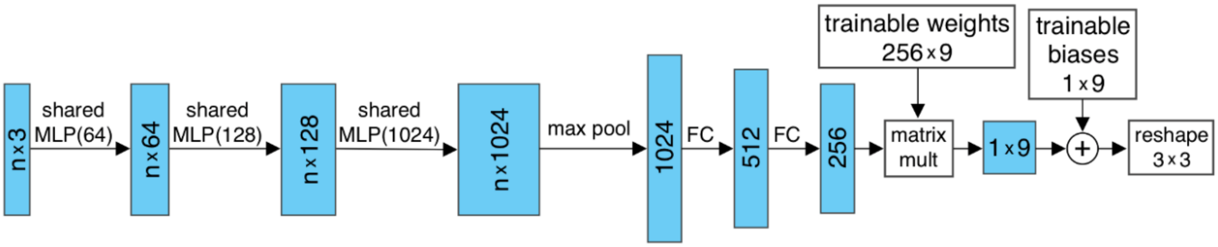
Suppose a point in the cloud is disturbed by a small amount, this disturbance should not affect the model predictions since it is common in Point Cloud representation for small noises. Since the points are not isolated and are always closely spaced in the point cloud, we should also consider the local features/information wrt to that point while making predictions.

## PointNet Architecture



The PointNet is a simple but efficient and powerful model. We will concentrate on the segmentation network more. The PointNet architecture is made up of MLPs, transformation networks and max-pooling. For Semantic Segmentation, we need to have both global and local features since we draw the boundary locally. Hence we use both global and local features in our network for the Segmentation task

To make the model invariant to the input data, we create a mini-network that predicts a 3x3 transformation grid, which is multiplied with the input matrix to get an invariant matrix. Basically think in this way: Assume the point cloud in the 3D space is rotated by an angle θ. So basically this 3x3 grid is kind of a rotational matrix. So using the input data, the mini-network predicts the appropriate 3x3 grid which rotates the point cloud in the opposite direction(rotates the point cloud by an angle -θ). So now given any rotation of the input point cloud, the model removes the rotation variance of the input point cloud using this mini neural network

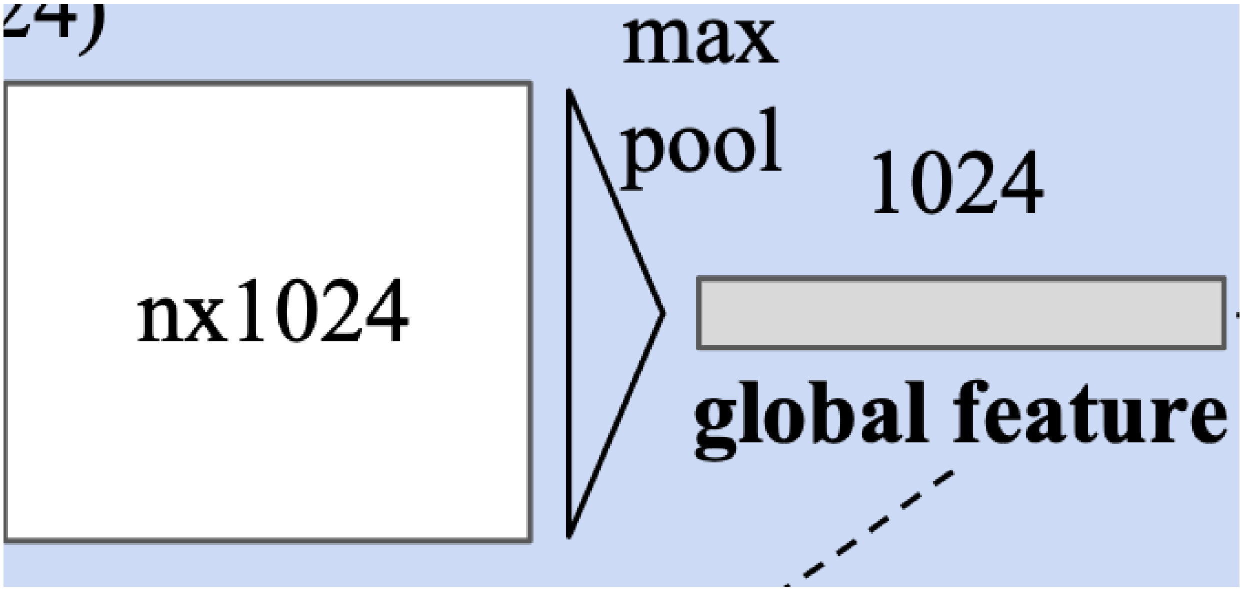


The above picture is the architecture of T-Net model

Similarly, we predict extend this method to remove the variance for the features as well by predicting a feature transformation using a similar mini-network but this time we are using a larger grid size. So to avoid overfitting, we add a regularization term. A is the feature transformation matrix

$$L_{reg} = \|I - AA^T\|_F^2,$$

The point cloud representation is an unordered representation when compared to pixels in a 2d image or voxels in a 3d volume. Hence sampling points in any order should not affect the model performance and the model should always almost predict the same output



So to tackle this problem, we use the Max-Pooling layer which basically removes the problem of unorderedness. Since we do max pooling across the layers, it really does not matter the order we sample the points from the point cloud

We concatenate the local and global features which makes optimisation faster since for Scene Segmentation, it is important for the model to be aware of the local variables as well

## Dataset

We experiment on the Stanford 3D Semantic Parsing Dataset. The dataset contains 3D scans from Matterport scanners in 6 areas including 271 rooms. Each point in the scan is annotated with one of the semantic labels from 13 categories (chair, table, floor, wall etc. plus clutter). To prepare training data, we firstly split points by room, and then sample rooms into blocks with area 1m by 1m. We train our segmentation version of PointNet to predict per point class in each block

Each point is represented by a 9-dim vector of XYZ, RGB and normalized location as to the room (from 0 to 1). At training time, we randomly sample 4096 points from each room.

## Training

No Dropouts used. The momentum for batch normalization starts with 0.5 and is gradually increased to 0.99. We use adam optimizer with an initial learning rate 0.001, momentum 0.9 and batch size 32. The learning rate is divided by 2 every 20 epochs.

Loss Function: Softmax classification loss + Regularisation loss

Accuracy Metric: IoU Score and Per-Point classification accuracy

## EndNote

To get to know more about the model, check out the paper: PointNet