1 Key Concept of HW

1.1 Background

1.1.1 Point Cloud Data

Point clouds are sets of 3D points that represent the shape and structure of objects or scenes in the form of unordered point sets. Due to the unordered property of the point cloud, traditional networks cannot directly use the point cloud data to train.

1.1.2 PointNet

PointNet was introduced by Charles R. Qi et al. in a 2017 paper titled "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation." It is a neural network architecture that directly takes raw point clouds as input and processes them without any pre-processing or feature extraction steps. PointNet is capable of learning meaningful features from point clouds, such as local and global geometric information, and can be used for tasks such as 3D object classification, semantic segmentation, and point cloud generation.

1.2 Core Concept of Paper

1.2.1 Make the Network Permutation Invariant

The idea of extracting a global feature is essential when doing classification and segmentation tasks because the input data cannot reflect the global information. The main challenge with processing point cloud data is that it is unordered. We want the same output features regardless of the order of the input point cloud. To have a permutation-invariant network, we need a symmetric function for unordered input. To handle the above problem, we apply shared MLP on the individual points in the input point cloud to expand the dimension of input. And we achieve permutation-invariant by applying max-pooling operations to aggregate information across the points.

1.2.2 Joint Alignment Network

The Joint Alignment Network (T-Net) can eliminate the influence of translation, rotation, and other factors on point cloud classification and segmentation. The input point cloud passes through this network to obtain an affine transformation matrix. Then, this matrix is multiplied by the original point cloud to obtain a new nx3 point cloud. The same method is applied to the feature space to ensure the invariance of features.

1.2.3 Concatenating data and global feature during segmentation

Different from object classification which just needs global feature information, part segmentation needs both global and local feature. To do local and global information aggregation, we use a method similar to U-Net. PointNet aggregates local and global features simply by concentrating the global feature and local feature of each point in the hidden layer of MLP. With this modification, the network can predict each point category relying on global semantics and local geometry.

1.3 How HW engages with the concept

1.3.1 Understand Point Cloud Data

The data we download is ModelNet10. It consists of two parts: vertives and faces.

- We first display the contents in the data file.
- We then let students to write some functions to pre-process the data. We first sample point-cloud from faces of the object. Then we do normalization, rotation, and adding noise.

1.3.2 Make the Network Permutation Invariant

- Firstly, we want students to understand shared MLP in the network. The shared MLP is the same as doing 1d convolution with kernel size 1. So we first let student implement the sharedMLP layer, and compared it with torch.conv1d package.
- We then let students the forward pass of Transform network and PointNetClassifier network so that they have a clear understanding of the whole network.

1.3.3 Joint Alignment Network

- Let students implement the forward pass of T-net in the notebook to understand the architecture of T-Net.
- In order to constrain the matrix learned by T-Net to be a rotation matrix, a penalty term is added to the loss of the module and has a coefficient alpha. Let students tune the alpha value and observe the accuracy change due to the alpha. This can help students understand the loss design and the intuition for the proposal of T-Net in PointNet.
- Let students write the code for loss function to get a better understanding of what the network tries to learn: classification error + regularization term.

1.3.4 Concatenating data and global feature during segmentation

This part is similar to the U-net structure we learned in class. Since the segmentation gives each point a class label, so we need both local and global information when doing segmentation.

- We let students implement the stack operation that concatenates previous hidden layer outputs and the global feature.

2 Homework without Solution

2.1 Introduction

PointNet is a deep learning model for point cloud processing, which is a type of 3D data representation commonly used in computer vision and robotics applications. Point clouds are sets of 3D points that represent the shape and structure of objects or scenes in the form of unordered point sets.
PointNet was introduced by Charles R. Qi et al. in a 2017 paper titled "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation." It is a neural network architecture that directly takes raw point clouds as input and processes them without any pre-processing or feature extraction steps. PointNet is capable of learning meaningful features from point clouds, such as local and global geometric information, and can be used for tasks such as 3D object classification, semantic segmentation, and point cloud generation.
The PointNet architecture uses multi-layer perceptrons (MLPs) and max pooling operations to process individual points in the point cloud, as well as symmetric functions to aggregate the features of all points into a global feature vector that captures the overall structure of the point cloud. PointNet can be extended with additional modules, such as PointNet++ for hierarchical feature learning or PointNet-based convolutional neural networks (CNNs) for local feature extraction.
PointNet has been widely used in various applications, such as autonomous driving, robotics, virtual reality, and augmented reality, due to its ability to process raw point cloud data directly and learn meaningful features from unstructured 3D data.
The structure of the PointNet is shown in Figure 1. Please follow the instructions in this notebook. You will build and train PointNet on Point Cloud classification and segmentation dataset.

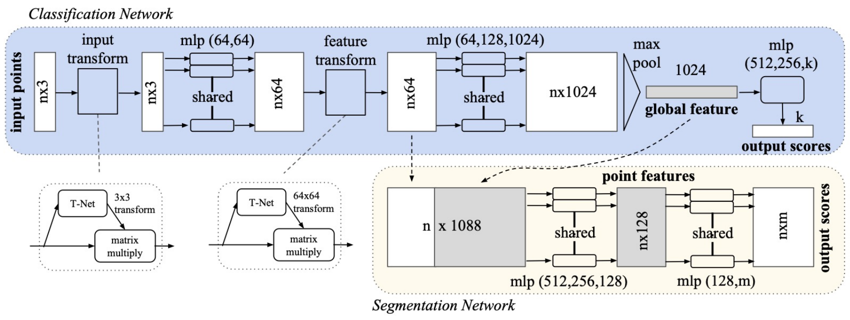

Figure 1: PointNet

2.2 Analysis Problem

A point cloud is a discrete *set* of data points in space. Because of this set-valued nature of point clouds, concepts from graph neural nets are often relevant in their processing.
For the first problem, we deal with 2D point cloud to get start. In Figure 1, we consider a simple network to process a 2d point cloud $X = \{x_i\}_{i=1}^{n} \in \mathbb{R}^{n\times 2}$, where n is the number of points. The original features of each point are its horizontal and vertical coordinates.

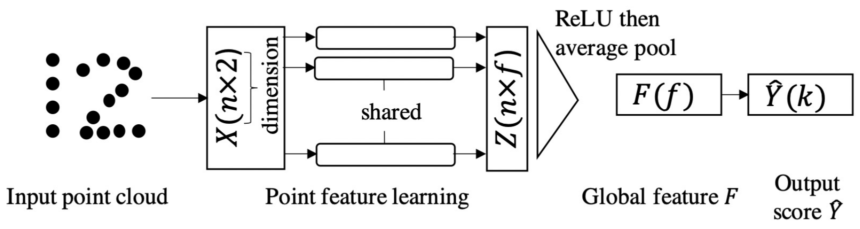

Figure 2: 2d point cloud processing network.

For example, an input point cloud with ground-truth of digit 1 could be represented as $\begin{bmatrix} 0 & 4 \\ 0 & 3 \\ 0 & 2 \\ 0 & 1 \end{bmatrix}$ where each row is a different point in the point cloud.
(a) The point feature learning module in Figure 1 learns f-dimensional features for each point separately. Specifically, it learns (shared) weights $W_1 \in \mathbb{R}^{2\times f}$ to get hidden layer outputs $Z = XW_1$. We then apply the nonlinear activation function element-wise and then use average pooling to yield the f -dimensional global feature vector $F \in \mathbb{R}^f$. Suppose we swap the first two points of the input point cloud X, i.e. $x_1, x_2, ..., x_n$ to $x_2, x_1, ..., x_n$. Show that the global feature F will not change.
Note: In reality, the network here is permutation invariant, as changing the ordering of the n input points in X will not affect the global feature F .

2.3 PointNet Classifier

(a) Finish the coding part in the colab notebook . Fine tuning the hyper-parameters to do the training

i Feel free to tune the alpha, and observe the output transform matrix of t-net and visualize the item in dataset before and after applying transform matrix, identify the similarities and differences between the input item and item applied transform matrix. What effect do you think alpha have? And when alpha is large what t-net is doing? Do you think whether alpha value can improve model performance?

ii Observe the confusion matrix, which objects have classification accuracy that is significantly higher than the others? Which objects have classification accuracy that is significantly higher than the others? Briefly explain why.

2.4 Understanding the structure and data dimension

The T-Net can get a learned Transform Matrix that transforms an object to a canonical position. The structure of this model can be seen as a small Point-Net. To better understand the model, we list the layers in T-Net, and we represent the shared-MLP in conv1d() way. B is the batch size of the data, and N is the number of points in an object. Furthermore, the brackets, like(B×N) define a vector $\in \mathbb{R}^M, (M = B \times N)$ dimension, rather than a matrix A$\in \mathbb{R}^{B\times N}$

	Input	Output
covn1d[64]	B× 3× N	
conv1d[128]		
conv1d[1024]		B× 1024× N
MaxPool	B× 1024× N	
Flatten		(B×1024)
Linear[B×9]	(B×1024)	
View layer[B,3, 3]		B×3×3

2.5 PointNet Segmentation

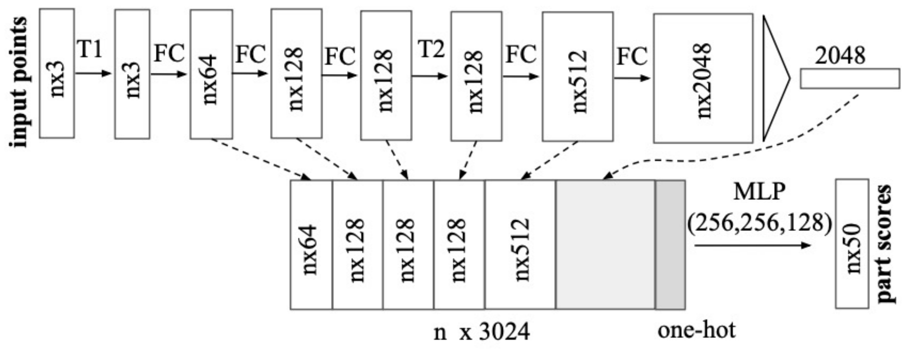

Figure 3: Segmentation PointNet

Finish the coding part in the colab notebook and answering the following questions.
(a) Observe the segmentation output of the module, what part of the object has the lowest segmentation accuracy? Briefly explain why.
(b) Briefly describe the difference between classification PointNet and segmentation PointNet. Briefly explain why segmentation PointNet is designed this way.

3 Homework with Solution

3.1 Introduction

PointNet is a deep learning model for point cloud processing, which is a type of 3D data representation commonly used in computer vision and robotics applications. Point clouds are sets of 3D points that represent the shape and structure of objects or scenes in the form of unordered point sets.
PointNet was introduced by Charles R. Qi et al. in a 2017 paper titled "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation." It is a neural network architecture that directly takes raw point clouds as input and processes them without any pre-processing or feature extraction steps. PointNet is capable of learning meaningful features from point clouds, such as local and global geometric information, and can be used for tasks such as 3D object classification, semantic segmentation, and point cloud generation.
The PointNet architecture uses multi-layer perceptrons (MLPs) and max pooling operations to process individual points in the point cloud, as well as symmetric functions to aggregate the features of all points into a global feature vector that captures the overall structure of the point cloud. PointNet can be extended with additional modules, such as PointNet++ for hierarchical feature learning or PointNet-based convolutional neural networks (CNNs) for local feature extraction.
PointNet has been widely used in various applications, such as autonomous driving, robotics, virtual reality, and augmented reality, due to its ability to process raw point cloud data directly and learn meaningful features from unstructured 3D data.
The structure of the PointNet is shown in Figure 1. Please follow the instructions in this notebook. You will build and train PointNet on Point Cloud classification and segmentation dataset.

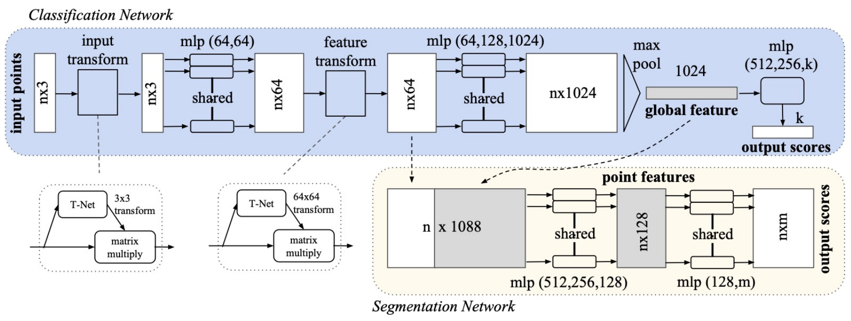

Figure 4: PointNet

3.2 Analysis Problem

A point cloud is a discrete *set* of data points in space. Because of this set-valued nature of point clouds, concepts from graph neural nets are often relevant in their processing.
For the first problem, we deal with 2D point cloud to get start. In Figure 1, we consider a simple network to process a 2d point cloud $X = \{x_i\}_{i=1}^{n} \in \mathbb{R}^{n\times 2}$, where n is the number of points. The original features of each point are its horizontal and vertical coordinates.

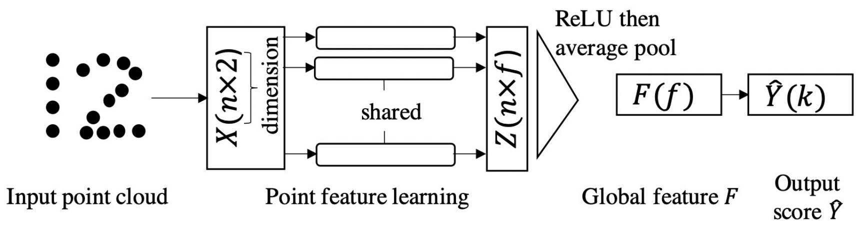

Figure 5: 2d point cloud processing network.

For example, an input point cloud with ground-truth of digit 1 could be represented as $\begin{bmatrix} 0 & 4 \\ 0 & 3 \\ 0 & 2 \\ 0 & 1 \end{bmatrix}$
where each row is a different point in the point cloud.
(a) The point feature learning module in Figure 1 learns f-dimensional features for each point separately. Specifically, it learns (shared) weights $W_1 \in \mathbb{R}^{2\times f}$ to get hidden layer outputs $Z = XW_1$. We then apply the nonlinear activation function element-wise and then use average pooling to yield the f -dimensional global feature vector $F \in \mathbb{R}^f$. Suppose we swap the first two points of the input point cloud X, i.e. $x_1, x_2, ..., x_n$ to $x_2, x_1, ..., x_n$. Show that the global feature F will not change.
Note: In reality, the network here is permutation invariant, as changing the ordering of the n input points in X will not affect the global feature F .
Solution: Expand X row-wise, derive X_1:

$$Z = XW_1 = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} W_1 = \begin{bmatrix} x_1W_1 \\ x_2W_1 \\ \dots \\ x_nW_1 \end{bmatrix} \tag{1}$$

The global feature F is the average all row vectors of Z:

$$F = \frac{1}{n}\sum_{i=1}^{n} ReLU(x_iW_1) \tag{2}$$

We can see that swapping the order of x_1, x_2 will not affect F .
Note: Actually, permuting the order of points $x_1, x_2, ..., x_n$ will not change F as it is a sum of all input point coordinates transformed by a weight matrix. Additionally, ReLU is invariant to the permutation of the order of input points.

3.3 PointNet Classifier

(a) Coding solution of classifier.ipynb: colab notebook . Fine tuning the hyper-parameters to do the training

i Feel free to tune the alpha, and observe the output transform matrix of t-net and visualize the item in dataset before and after applying transform matrix, identify the similarities and differences between the input item and item applied transform matrix. What effect do you think alpha have? And when alpha is large what t-net is doing? Do you think whether alpha value can improve model performance?

Solution: Alpha is the regularizer in the loss function, larger alpha means we inject inductive bias into the model, makeing the transform matrix outputed by T-net be close to orthogonal matrix. With large alpha, T-net output an orthogonal transform matrix. Applying transform matrix, rotating the item in 3d-coordinate system. By tuning different alpha from 0.0001 to 10, we can find that alpha doesn't help model performance much.

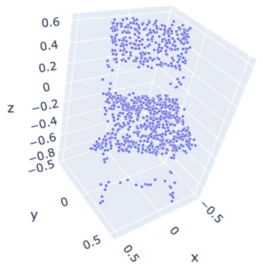

Figure 6: Chair before transform

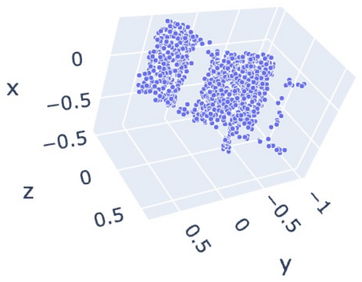

Figure 7: Chair after transform

ii Observe the confusion matrix, which objects have classification accuracy that is significantly higher than the others? Which objects have classification accuracy that is significantly higher than the others? Briefly explain why.

Solution: Chair, monitor and table have classification accuracy that is significantly higher than the others. Dresser and night stand have classification accuracy that is significantly lower than the others. This is because chair, monitor and table are significantly different from other objects in the dataset, thus they are easier to classify. While dresser and night stand are similar to each other, thus they are harder to classify.

3.4 Understanding the structure and data dimension

The T-Net can get a learned Transform Matrix that transforms an object to a canonical position. The structure of this model can be seen as a small Point-Net. To better understand the model, we list the layers in T-Net, and we represent the shared-MLP in conv1d() way. B is the batch size of the data, and N is the number of points in an object. Furthermore, the brackets, like(B$\times$N) define a vector $\in \mathbb{R}^M, (M = B \times N)$ dimension, rather than a matrix A$\in \mathbb{R}^{B\times N}$

	Input	Output
covn1d[64]	B× 3× N	B× 64× N
conv1d[128]	B× 64× N	B× 128× N
conv1d[1024]	B× 128× N	B× 1024× N
MaxPool	B× 1024× N	B× 1024 × 1
Flatten	B× 1024 × 1	(B×1024)
Linear[B×9]	(B×1024)	(B×9)
View layer[B,3, 3]	(B×9)	B×3×3

3.5 PointNet Segmentation

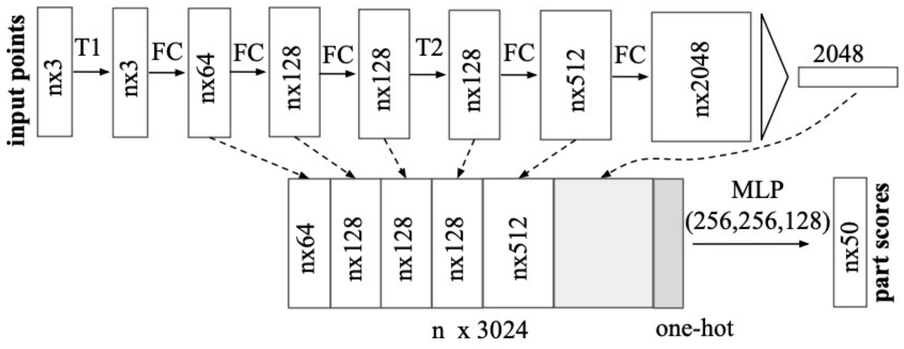

Figure 8: Segmentation PointNet

Coding solution of segmentation.ipynb: colab notebook and answering the following questions.
(a) Observe the segmentation output of the module, what part of the object has the lowest segmentation accuracy? Briefly explain why.
Solution: The tail part of the plane is inseparable from the plane body in most case. This is because they don't have clear borders, which is hard for the module to segment.
(b) Briefly describe the difference between classification PointNet and segmentation PointNet. Briefly explain why segmentation PointNet is designed this way.
Solution: Feature dimension before max pool is 2048 in Segmentation PointNet while in Classification PointNet is 1024. What's more, Segmentation PointNet concatenate every output of the mlp layer with the global feature to form the feature feeded to the final segmentation mlp. This is because segmentation task needs more information than classification task. The concatenate operation is using the U-Net way of thinking to preserve local feature while generating global feature. Higher feature dimension is to keep more information after max pool.