PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space

Sikai Zhong

February 14, 2018

COMPUTER SCIENCE

Table of contents

PointNet

Property of Point Cloud

- It is unordered,networks must be invariant to permutations;
- Points are not isolated and must have neighbor information;
- It must has invariance under transformations because all points are part of a rigid object;

Symmetry Function for Unordered input

A symmetry function takes n vectors and outputs a result that is invariant to the order of a set;

$+$ and * are operator candidates for symmetry function;

Pipeline of PointNet

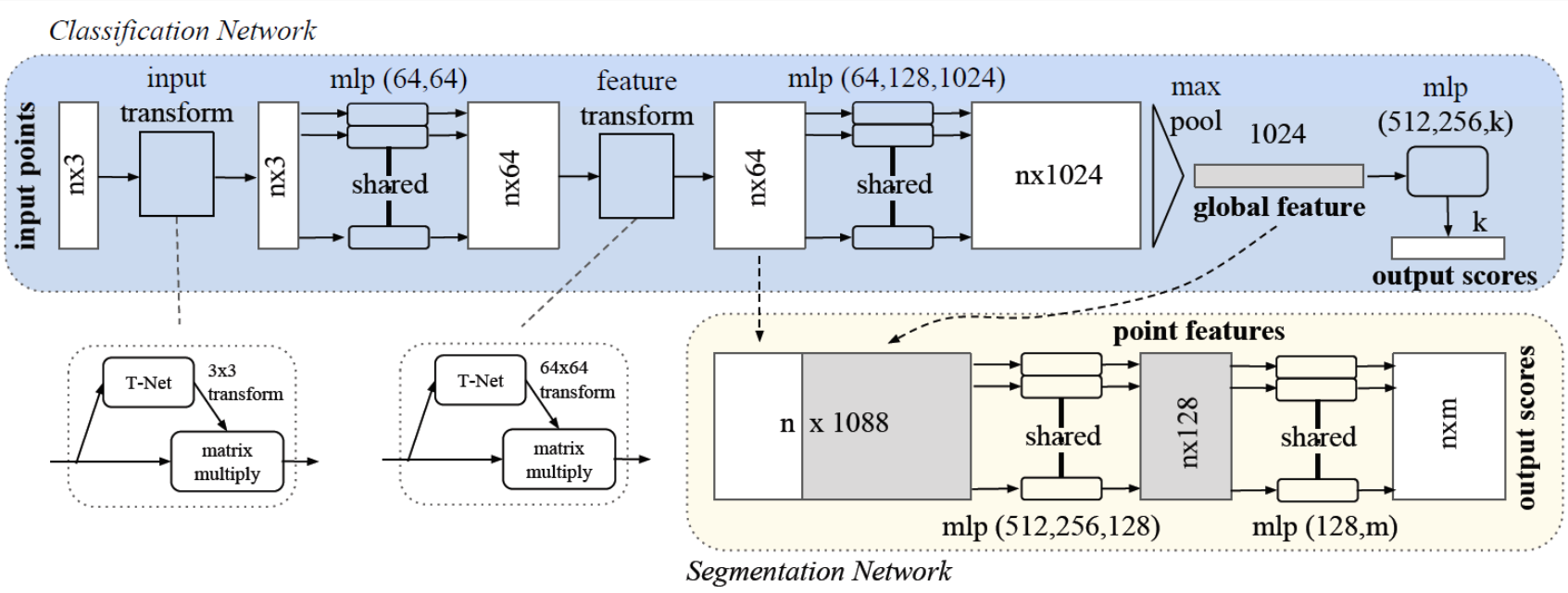


Figure 1: The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. mlp stands for multi-layer perceptron, numbers in bracket are layer sizes. Batch norm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net.[1]

Pipeline of PointNet

$$f(x_1, x_2, ..., x_n) = \gamma(MAX_{i=1,...,n}\{h(x_i)\}); \quad (1)$$

Where γ and h are multi-layer perception(MLP) networks.

PointNet

- PointNet can not capature local structure induced by the metric space;
- Local structure is very important to the success of convolutional architectures;

PointNet++

General Ideal of PointNet++

1. Partition the set of points into overlapping locally regions using farthest point sampling (FPS) algorithm;
2. Capture the local features using PointNet;
3. Group the features into bigger unit and calculate the higher features again;

Sampling Layer

1. Given input points $\{x_1, x_2, ...x_n\}$;
2. Using FPS to get a subset $\{x_{i_1}, x_{i_2}, ...x_{i_m}\}$ such that x_{i_j} is the most distant point to the result of the points as centroids;

Grouping Layer

1. Given input points of size of $N \times (d + C)$ and the coordinates of a set of centroids of size $N' \times d$ (d is the dimension of points, C is the dimension of features)
2. The outputs are groups of points of size $N' \times K \times (d + C)$;
3. K varies from different levels;

Grouping Layer

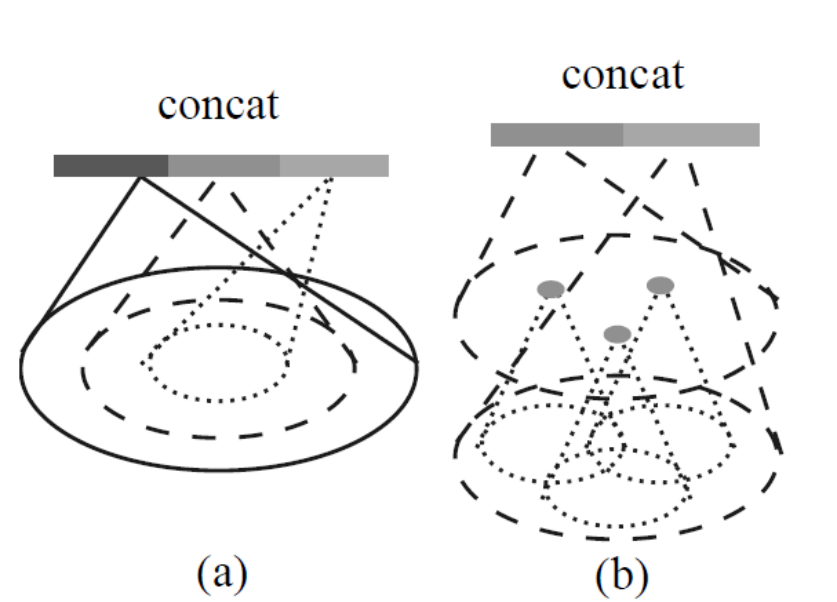


Figure 2: (a) Multi-scale grouping (MSG) (); (b) Multiresolution grouping (MRG)(features of a region at some level Li is a concatenation of two vectors.

Grouping Layer

- MRG:Features at different scales are concatenated to form a multi-scale features,computationally expensive;
- MSG: One vector (left in figure. 10) is obtained by summarizing the features at each subregion from the lower level L_i using the set abstraction level. The other vector (right) is the feature that is obtained by directly processing all raw points in the local region using a single PointNet).

PointNet Layer

1. the input are N' local regions of points with data size $N' \times K \times (d + C)$; K is the number of points in one region;
2. Output data size is $N' \times (d + c')$

Pipeline of PointNet++

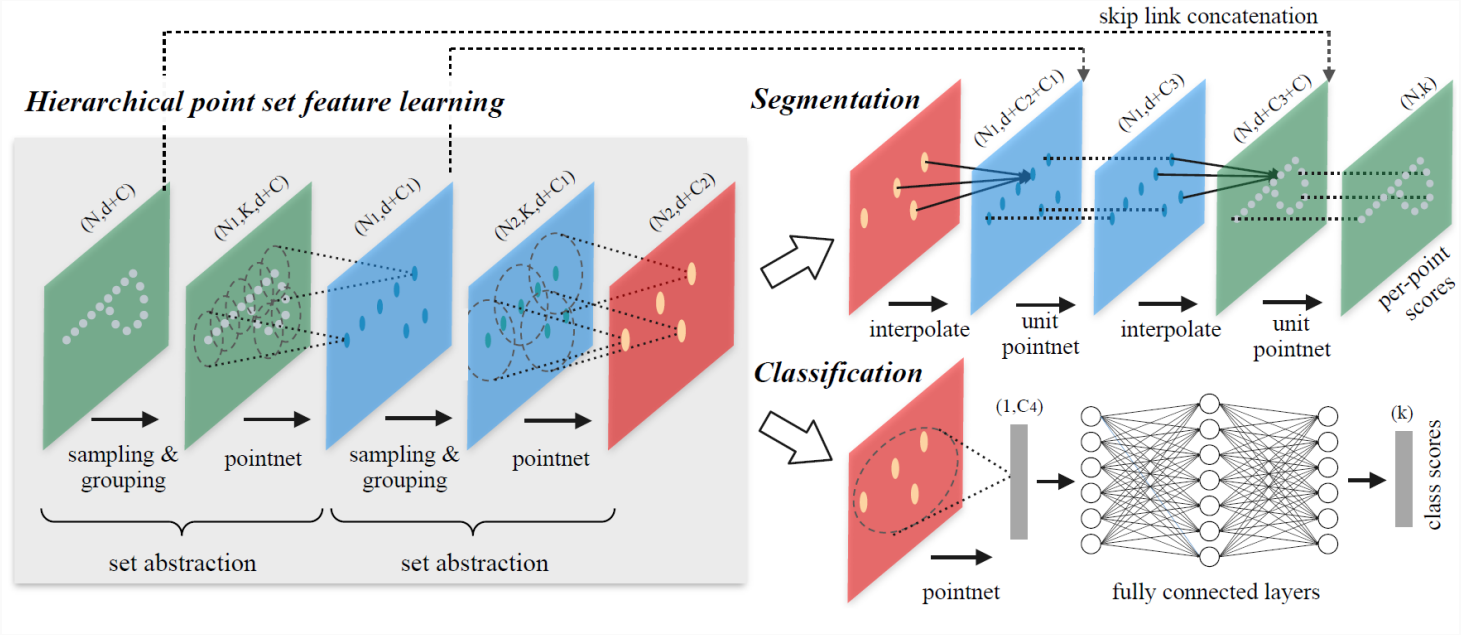


Figure 3: Illustration of our hierarchical feature learning architecture and its application for set segmentation and classification using points in 2D Euclidean space as an example. Single scale point grouping is visualized here[2]

Experiments

Datasets

- MNIST : Images of handwritten digits with 60k training and 10k testing samples.
- ModelNet40: CAD models of 40 categories (mostly man-made). We use the official split with 9,843 shapes for training and 2,468 for testing.
- SHREC15 : 1200 shapes from 50 categories. Each category contains 24 shapes which are mostly organic ones with various poses such as horses, cats, etc. We use five fold cross validation to acquire classification accuracy on this dataset.
- ScanNet : 1513 scanned and reconstructed indoor scenes. We use 1201 scenes for training, 312 scenes for test;

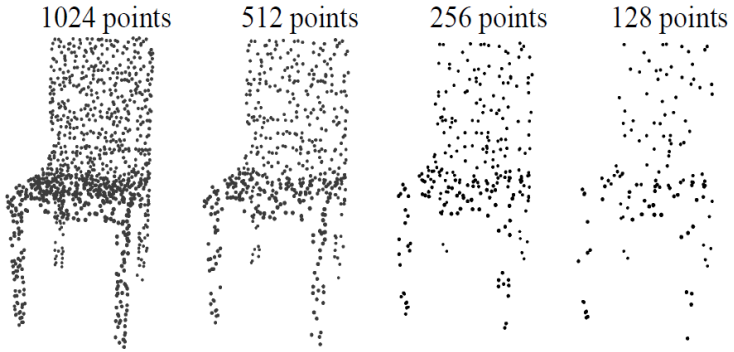
Classification

Method	Error rate (%)
Multi-layer perceptron [24]	1.60
LeNet5 [11]	0.80
Network in Network [13]	**0.47**
PointNet (vanilla) [20]	1.30
PointNet [20]	0.78
Ours	0.51

Table 1: MNIST digit classification.

Method	Input	Accuracy (%)
Subvolume [21]	vox	89.2
MVCNN [26]	img	90.1
PointNet (vanilla) [20]	pc	87.2
PointNet [20]	pc	89.2
Ours	pc	90.7
Ours (with normal)	pc	**91.9**

Table 2: ModelNet40 shape classification.



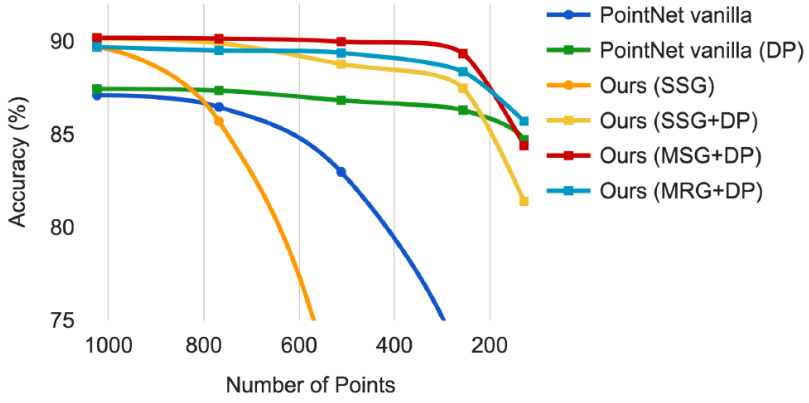


Figure 4: Left: Point cloud with random point dropout. Right: Curve showing advantage of our density adaptive strategy in dealing with non-uniform density. DP means random input dropout during training; otherwise training is on uniformly dense point

Point Set Segmentation for Semantic Scene Labeling

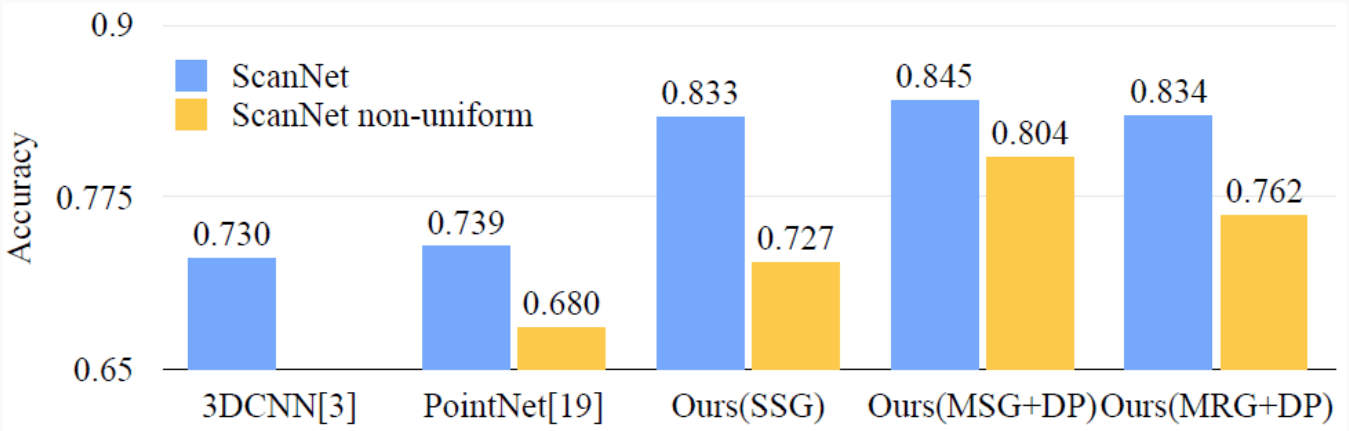


Figure 5: Scannet labeling accuracy

References i

C. R. Qi, H. Su, K. Mo, and L. J. Guibas.
Pointnet: Deep learning on point sets for 3d classification and segmentation.
Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 1(2):4, 2017.

C. R. Qi, L. Yi, H. Su, and L. J. Guibas.
Pointnet++: Deep hierarchical feature learning on point sets in a metric space.
In *Advances in Neural Information Processing Systems*, pages 5105–5114, 2017.