Literature Review

- Abstract
- Introduction
- Problem Statement
- Types of Data used
- Conclusion

Abstract

- Point Net is a pioneer in direction of deep learning on point sets.
- There is limitation that is doesn’t capture metrics space points, so it limits the ability to recognize fine-grained patterns and generalizability to complex scenes.
- A hierarchical neural network is introduced that uses Point Net recursively called Point Net++ and obtained better result for 3D point clouds.

Introduction

- Needed Point Net ++
  - Point Net jumps directly from per-point features to global features. This results on that it doesn’t consider neighbors, local surfaces, fine details, curvature of small regions and many more.
  - Where, Point Net ++ reduces generalization and increase the ability to detect fine details Performance on segmentation.

Flow chart

- **Point Net**

  [Point1] → MLP →|

  [Point2] → MLP →|→ MAX POOL → Global Feature → Classification

  [Point3] → MLP →|

  Real examples: Learns overall length, width and shape.

- **Point Net++**

Point Cloud

↓

Sampling → Local Grouping (kNN/ball)

↓

Mini PointNet (per group)

↓

Larger Region Grouping (hierarchical)

↓

Global Feature

↓

Classification/Segmentation

Real examples: Learns Wing edges, Curvature of noise, Tail geometry etc.

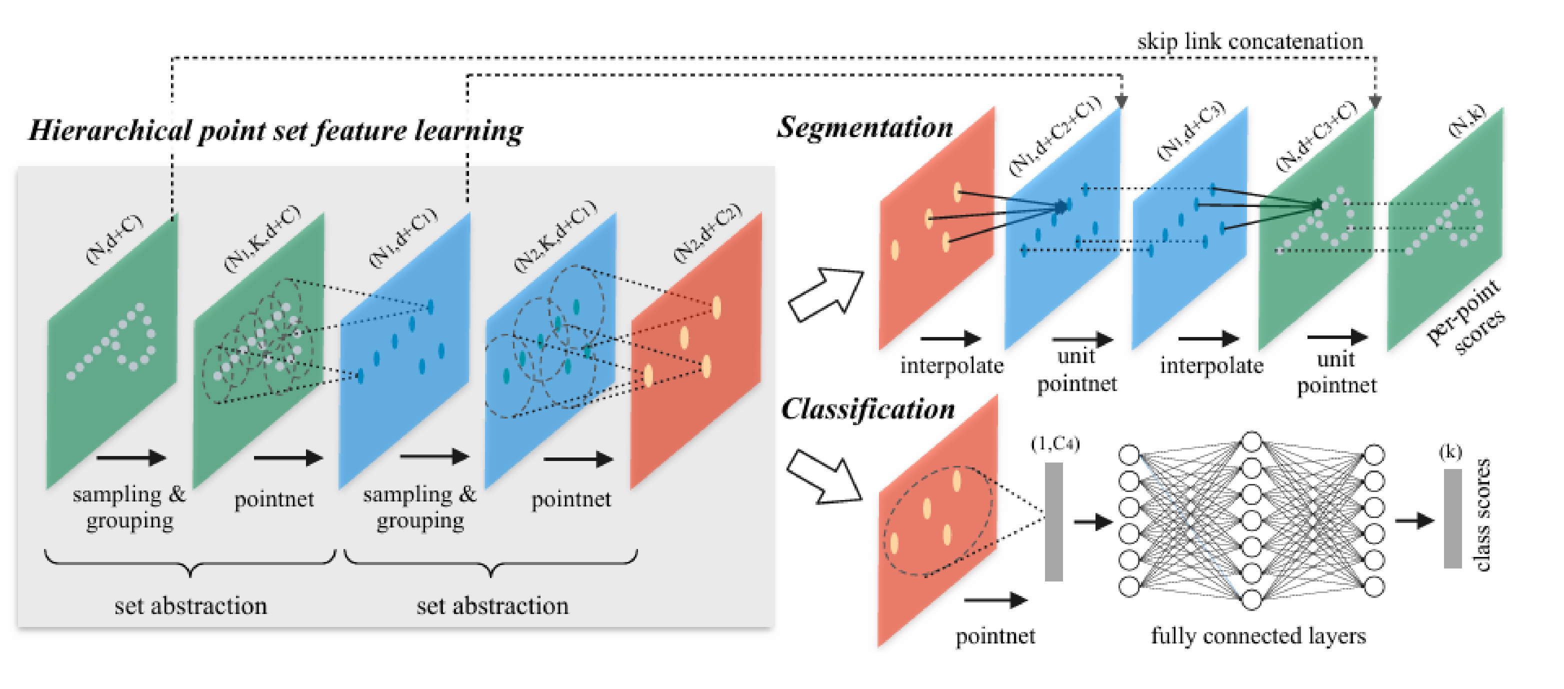

Image Source: Qi et al., 2017 (Point Net++)

Point Net++ has to address two issues:

- How do we divide the point cloud into meaningful regions?
  - ❑ Uses of the kNN, Ball query (Epsilon Ball), Voxel Grouping
- How do we learn local features for each regions?
  - ❑ Point Net is chosen as the local features extractors
    - Produce unordered point
    - robust to noise
    - Strong per point feature

- **Center point and Certain radius are chosen by a method called Farthest Point Sampling (FPS).**
  - FPS is based on:
    - ✓ Entire space of neighborhoods, evenly.
    - ✓ Neighborhoods naturally adapt to the object's geometry - dense areas from many small neighborhoods, while sparse areas get larger one.

Problem statement

- ϕ= (M, *d*)
- R^n → Euclidean Space
- M → Set of points
- d → distance metric
- Density of M may not be uniform everywhere.
- Our goal is to design a learning function ‘f’ that takes this point set as input and produces meaning full semantic information

Function of the Point Net ++ are:

- Classification: e.g finding an airplane from model Net40
- Segmentation: finding the wings, tail etc.

Hierarchical structure = PointNet++ → a member of set abstraction level. At each level, a set of points is processed and abstracted to produce a new set with fewer elements.

These abstraction levels are divided into three levels:

1. **Sampling Layers:** for given input points {x1, x2……… } we use iterative (FPS) for choosing centroid. FPS spreads the centroid evenly across the object.
   - FPS → Farthest Point Sampling
   - FPS gives Data-Dependent Sampling, meaning "if adapts the object is shape.
2. **Grouping layer:**
   - Input to this layer = N * (d + c)
   - set of centroids of size = N1 * d
   - Output size N1 * K * (d + c) K → number of points in the neighborhood of centroid points
   - K → varies across groups but the succeeding District layer is able to convert flexible number of points into a fixed length local region feature vector.

3. **PointNet Layer:**

- ❑ Each local neighborhood (group) is fed into a mini–Point Net to learn local features.
- ❑ Step by Step

40 nearby points.

1→ First subtract centroid coordinates (origin=centroid).

2→ Shared map (earn) pattern.

3→ Max pooling compresses any 40 points → a single 1 x c feature vector.

Uneven density makes learning difficulty so, PointNet++ solves this using ***density-adaptive feature learning.***

- **Multi Scale Grouping (MSG) and Multi Resolution Grouping (MRG)**

- **MSG method:** features extracted at different spatial scales (snow radius, medium radius, large radius) and concatenate them to form one multiple-scale feature vector.

- **MRG method:** It avoids heavy computation by doing two kinds of iterations passed from feature the previous layer (low resolution) and fine features (high resolution)
  - Quality of local → coarse region
  - linear detail → fine feature

Point Net ++ is set abstractions that reduces no of points at each Layer. This is good for learning, but it becomes a problem for segmentation, because segmentation needs a prediction for every original point. So, we must bring features back from the subsampled (few) points to the original (many) points. This process is called *Point feature Propagation***.**

PFP up samples "feature from fewer points → more points.

How?

- *Introduction (estimating features of missing points)*
  - *use inverse-distance weighting*
- *Skip connection:*
  - *Interpolated features.*
  - *Saved features from earlier level.*
- Unit PointNet to Clean features
  - After combining features, a small map is applied to refine each point feature vector. This ensures smooth consistent features across all points.

Types of Data Evaluated :

- Object classification is evaluated by accuracy. Semantic scene labeling is evaluated by average voxel classification accuracy following [5]. We list below the experiment setting for each dataset:
  - 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.
  - Scan Net: 1513 scanned and reconstructed indoor scenes. We follow the experiment setting in [5] and use 1201 scenes for training, 312 scenes for test.

Results

- **For MINIST Results (2D Digit Classification):**
  - Error in Point Net ++ is smallest 0.51% where, 90.7% and 91.9% percent of accuracy for shape classification. The accuracy is greater for Point Net ++ Normals than Point Net.
  - Point Net ++ normals represent surface direction at each point.
  - Point Net ++ represents coordinates only.

CONCLUSION

- PointNet++ improves PointNet by learning local + global features using a hierarchical structure.
- It captures fine geometric details and handles non-uniform point densities using MSG and MRG layers.
- Achieves higher accuracy on MNIST, ModelNet40, and ScanNet, especially when using normals.
- Overall, it is a robust, scalable, and more powerful framework for 3D point cloud classification and segmentation.

Reference

Qi, C. R., Yi, L., Su, H., & Guibas, L. J., “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space,” *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.