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

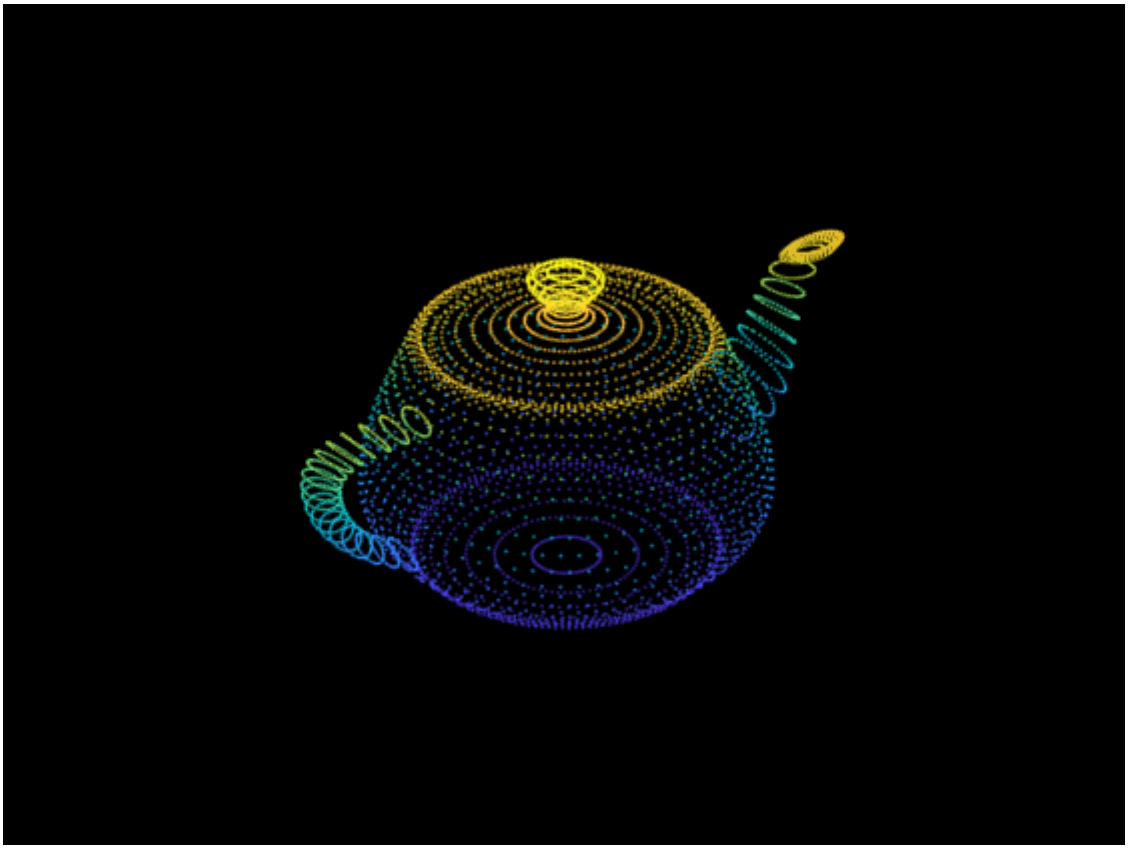
What is the pointcloud?

Pointcloud는 3차원 공간 상에 퍼져있는 여러 포인트의 집합을 의미한다.
3D 공간 정보를 읽어들이기 위해 Lidar Sensor(좌), RGB-D Camera(우) 등을 사용한다.

Point Cloud 데이터는 2D 이미지 데이터와 달리 정형화 되어있지 않고, Unordered한 특성을 가진다.

2D 데이터의 경우 정해진 격자 구조의 형태 안에 정보가 저장되지만, Point Cloud 데이터는 3D 공간 상의 수많은 점들을 순서없이 기록하는 방식으로 데이터가 저장된다. 이러한 데이터는 딥러닝 모델에 있어 상당히 치명적으로 다가온다.

Sparse한 Point Cloud Data



2D 이미지 데이터가 정해진 격자에 Pixel 값이 모두 존재하는 Dense한 특성을 지녔다면, Point Cloud 데이터는 매우 Sparse한 성질을 가지고 있다. Point에 비해 빈 공간이 훨씬 더 많은 Point Cloud의 특징 때문에 데이터는 그 크기에 비해 얻을 수 있는 유의미한 정보가 부족해짐. 전처리를 통해 데이터를 정형화하더라도 동일하게 가지고 있는 성질.

이러한 데이터로 인공지능 모델이 학습 될 경우 유의미한 정보를 거의 얻기 힘들고, 학습 난이도만 올라가게 된다.
이로 인해 3D 인공지능 모델은 좋은 성능을 보여주지 못했다.

이후 Point Clod 데이터를 활용한 3D 인공지능 모델 연구는 Point Cloud 데이터가 가진 성질의 한계를 극복하는 방향으로 수행되었다. 단순히 Point Cloud 데이터를 전부 해석하는 것이 아니라, Point Cloud 데이터로부터 유의미한 정보를 추출하는 방식으로 연구가 이루어졌다. 그 결과 PointNet을 시작으로 Point Cloud로부터 유의미한 정보를 추출하는 등 좋은 성과를 보이기 시작했음.

ABSTRACT

“Point Cloud is an important type of geometric data structure”
“Due to its irregular format, most researchers transform such data to regular 3D voxel grids or collections of images”
“This, however, renders data unnecessarily voluminous and causes issues.”

- 객체로부터 3차원 기하학적 정보를 얻기위해 Point Cloud를 채택했으나,
- Point Cloud의 Sparse하고 Unordered한 특성으로 인해 딥러닝 모델 적용에는 한계가 분명했음

- 기존 연구에서는 Point Cloud 데이터를 Voxel 또는 Image 집합으로 변환시키려는 시도가 많았다.
- 하지만 이렇게 인위적으로 Format을 바꾸면, Quantization artifact (Point 사이 빈 공간)을 발생시키고 결국 데이터가 불필요하게 Volumionous하다는 한계점은 여전했다.
  → 데이터의 용량이 쓸데없이 커짐

“ In this paper, we design a novel type of neural network that directly consumes point clouds, which well respects the permutation invariance of points in the input”

- 본 논문에서는 Permutation-Invariant한 새로운 네트워크를 제안해 Point Cloud를 Input으로 사용하도록 했다.

“it shows strong performance on par or even better than state-of-the-art (SOTA)”

- 본 논문이 처음 출시된 2017년 당시 3D Computer Vision에서 가장 성능이 좋은 모델이였음

Instruction

“Since point clouds or meshes are not in a regular format, most researchers typically transform such data to regular 3D voxel grids or collections of images (e.g, views) before feeding them to a deep net architecture.”

“This data representation transfor- mation, however, renders the resulting data unnecessarily voluminous — while also introducing quantization artifacts that can obscure natural invariances of the data.”

- 2D-Image는 Matrix로 얻어지기 때문에 Regular Format을 가지지만, 3D-Point는 Irregular format으로 얻어진다.
- 이를 해결하기 위해 3D Voxel grids or Collections of images로 Format을 변환시킨 뒤 네트워크 입력으로 넣었으나 Abstract에서 앞서 말했다 싶이 문제는 여전함.

"For this reason we focus on a different input representation for 3D geometry using simply point clouds. - and name out resulting deep nets PointNets"

- 따라서 본 논문은 Point Clouds를 사용한 3D Geometric input representation, PointNets을 제안한다.

"Point clouds are simple and unified structures that avoid the combinatorial irregularities and complexities of meshes, and thus are easier to learn from."

*"The PointNet, however, still has to respect the fact that **a point cloud is just a set of points** and therefore **invariant to permutations of its members**, necessitating certain **symmetrizations in the net computation**. Further **invariances to rigid motions** also need to be considered."*

- Point Clous는 Mesh처럼 Point 조합으로 이루어진 것이 아니기 때문에 데이터를 직접 다루기에는 쉬울 지 몰라도, 단순한 Points의 집합이기 때문에 Permutation-Invariant 해야하며, 연산 과정에서 반드시 Symmetry함을 고려해야 한다.

"In the basic setting each point is represented by just its three coordinates(x,y,z). Additional dimensions may be added by computing normals and other local or global features."

- 각 Point는 (x, y, z) 좌표를 가지고, $Normals$와 $Local\ Features$, $Global\ Features$ 연산에 따라 차원이 추가됨

---

*"Key to our approach is the use of **a single symmetric function, max pooling.** Effectively the network learns a set of optimization functions/criteria that select interesting or informative points of the point cloud and encode the reason for their selection. The final fully connected layers of the network aggregate these learnt optimal values into the global descriptor for the entire shape as mentioned above*

*(**shape classification**) or are used to predict per point labels (**shape segmentation**)”*

- symmetric function network of Max Pooling Layerfmf 하나 채택
  - Point Cloud 중 관심 영역을 선택하는 함수를 최적화해주는데 효과적인 네트워크 역할
- 네트워크 마지막에는 FC Layer를 연결
  - 최적화된 값을 Global Descriptor와 연결시켜, Classification 또는 Segmentation 작업을 수행

“More interestingly, it turns out that our network learns to summarize an input point cloud by a sparse set of key points,”

- PointNet은 적은 수의 key points만으로 입력 Point cloud를 Summarize하도록 학습을 진행함

“The key contributions of our work are as follows:”

- 3D에서 Unorders Point Sets을 사용하는데 최적화된 새로운 모델을 제안
- PointNet이 3D shape Classification을 수행하도록 훈련될 수 있음을 보여줌

PointNet은 3D Object Classification/Segmentation 같은 Task에서 좋은 성능을 보여준 모델.

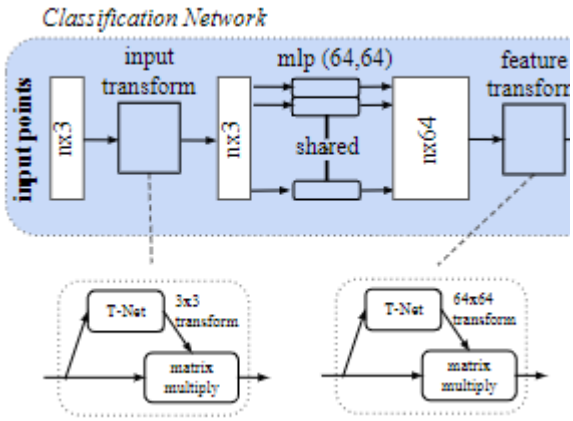
Method

- Properties of Point Sets
  - Unordered
    - Point Set은 Irregular한 특성 때문에 N개의 Point Set이 있다면, $N!$ permutation에 대해서 모두 같은 결과가 나오도록 Network를 구성해야 한다.
  - Interaction among Points

        - Point는 Nearby Points와도 Meaningful한 Subset을 가진다.
        - Network model은 Nearby Points들의 Local Structure, Combinational Interaction을 학습해야 한다.
    - Invariance under transformations
        - Roate, Translate 등과 같은 Transformations로 인해 Point에 대한 값이 달라져서는 안된다.

- PointNet Architecture

> **PointNet Structure’s key three Points**
>
> 1. the Max pooling layer as a **symmetric function** to aggregate information from all the inputs
> 2. a local and global information combination structure
>   a. Segementation Task를 위해선 **Global Feature과 Local Feature이 Concatenate**(결합)되어야 한다.
> 3. two joint alignment networks that align both input point and point features
>   a. Point Cloud에 Transformation이 일어나도 Target Task의 결과가 달리지 않아야한다.
>   b. 이를 위해 T-Net이라는 Mini Network를 사용했다.
>   - T-Net의 구조는 Max-Pooling과 FC-Layer의 결합 형태



- Affine Transformation matrix를 predict하고 이 transformation을 Input Point에 적용
- Feature Extraction matrix 부분도 있긴함

- Theoretical Analysis
  - Universal Approximation
    - Neural Network Continuous Set Function의 Approximation ability
      - Set Function의 Continuity로 인해 작은 Perturbation은 큰 변화를 일으키지 않음.
      - Hausdorff Distance에 대한 Continous Set function을 고려하면 쉽게 이해 가능
  - Bottleneck Dimension and stability
    - PointNet의 Expressiveness는 Max-Pooling layer의 Dimension K에 영향을 크게 받는다.
    - 이 부분은 이해 못함.. ㅋㅋ

⇒ 위 두 Theorem을 통해 Model이 Perturbation, Corruption에 어느정도 Robust함을 보여줌.

⇒ 이를 통해 PointNet은 Sparse key point set만으로 Shape을 Summarize하는 것을 잘 학습했음을 알 수 있다.

Limitation

1. 입력 데이터의 크기에 제한이 있어, Large Scale의 3D Point cloud data 처리에는 한계가 있음
2. Rotate, Translate 등의 Transformation에 대해 Invariant하지만, 지나친 변형에 대해서는 그렇지 못함
3. 입력 데이터를 처리하기 위해 각 Point의 좌표 정보를 사용하는데, 공간 정보를 완전히 보존하지 않는다.
    a. 입력 데이터가 물체의 표면을 나타내는 경우, 물체의 회전 및 이동에 따라 Point의 좌표가 변한다.
    b. PointNet은 물체를 동일한 Class로 인식하기는 하지만 실제 모양과는 다른 결과를 출력할 수 있음.
    c. 이는 PointNet++를 통해 개선되었다.

Next Paper

VoxelNet : Input Data를 3D Voxel Grid로 분할하여 처리하는 모델

Hard Things to understand

- Affine Transformation Matrix, Hausdorff Distance, Continuous Set Function, MLP (Multi-Layer Perception network
- Joint Alignment Network

Related Work

Point Cloud Features

Point features는 특정 변환 (transformation)에 불변하도록 설계되어 있음.

Deep Learning on 3D Data

Volumetric CNNs **:** 3D Convolutional neural networks를 voxelized shapes에 최초로 적용시킨 사례

- voxel : volume + pixel → 3D pixel
- 억지로 voxelized 시키려다보니 volumetric representation이 제한되고, 3D Convolution 연산에 연산량이 매우 많았음

Multiview CNNs, Spectral CNNs, Feature-based DNNs 등 다양한 시도가 있어왔음

Deep Learning on Unordered Sets

“From a data structure point of view, a point cloud is an unordered set of vectors.”

“Not much work has been done in deep learning on point sets”

Problem Statement

“*We design a deep learning framework that directly consumes unordered point sets as inputs.”*

- Unordered point sets를 Input으로 받는 딥러닝 프레임워크를 설계했다.
- Point cloud는 A set of 3D Points로 표현 가능
  - Each Points $P_i|i = 1, ..., n$ where is a vector of its (x, y, z) plus extra feature channels (color, normal)

“Our proposed deep network outputs kscores for all thekcandidate classes”

“Our model will outputn×mscores for each of thenpoints and each of themsemantic sub-categories.”

Model은 Classification Task를 위한 구조와 semantic Segmentation Task를 위한 구조로 구성되어 있다.

- 자세한 건 모델을 보면서 확인 가능

Deep Learning on Point Sets

Properties of Point Sets in R^n

> **Three Main Properties:**
>
> 1. **Unordered**
> 2. **Interaction among points**

3. **Invariance under transformations**

1. Unordered

“a network that consumesN3D point sets needs to be invariant toN! permutations of the input set in data feeding order.”

Point Cloud는 Pixel array처럼 순서가 있는 것이 아니라 Points의 집합이기 때문에 특정한 순서가 존재하지 않는다.

- Data가 들어가는 순서는 상관이 없음을 의미하기도 함.

만약 Network가 N개의 3D Point Sets를 Input으로 이용한다면, N!개의 permutations(순열)에 대해서도 Invariant한 Featur를 학습해야한다.

2. Interaction among Points

Points는 공간에서 거리 정보를 가지고 있기 때문에 Points는 독립적인 것이 아닌 이웃하는 Point 간의 관계는 유의미하다.

그렇기 때문에 모델은 Local Structure from nearby points를 잘 파악할 수 있어야 한다.

However, PointNet은 Local Structure을 잘 파악하지 못함

3. Invariance under transformations

Geometric object로서 Point Set은 Rotate, Translate 등 어떤 변형에도 Invariant한 Representation을 학습해야한다.

즉, Points가 변한다고 해서 Classification 분류 값이나 Segmentation 값이 변하면 안된다.

PointNet Architecture

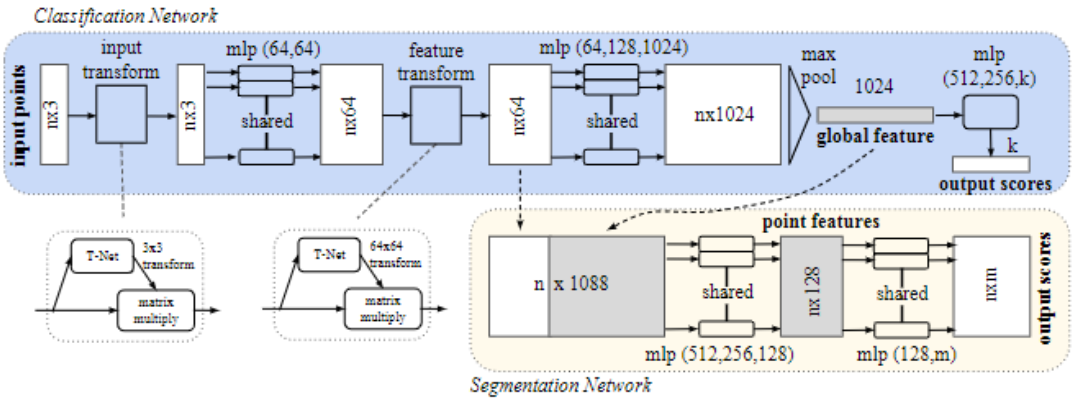


Figure 2. **PointNet Architecture.** The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. "mlp" stands for multi-layer perceptron, numbers in bracket are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net.

Global Feature

- Classification Network에서 Symmetric fuction Max-pooling을 통과하고 나온 Output
- Classification을 하기 위한 Feature

Local Feature

- Classification Network에서 중간에 Featur Transform을 거쳐서 나온 Output

Max Pooling Layer

- Interesting 또는 Informative Point를 고르는 기준을 학습하고 그 기준에 대한 근거를 Encoding한다.

Fully-Connected Layer ⇒ CAM (Classification Activation Map)에 대한 개념 이해가 필요함

- Max Pooling 결과로 나온 값을 모아서 Global Descriptor로 만든다.
- Global Descriptor는 Entire Shape이나 per point labeling에 사용

Network

- Input으로 들어온 Point Cloud를 sparse set of key points로 Summarize하는 것을 학습한다.
- PointNet은 Outlier 또는 Data Missing 등과 같은 Input Point의 perturbation에 Robust하다.

> **PointNet Structure's key three Points**
>
> 1. the Max pooling layer as a symmetric function to aggregate information from all the inputs
> 2. a local and global information combination structure
>   a. Segementation Task를 위해선 Global Feature과 Local Feature이 Concatenate(결합)되어야 한다.
> 3. two joint alignment networks that align both input point and point features

1. **the Max pooling layer as a symmetric function to aggregate information from all the inputs**

Input permutation에 Invariant하기 위한 3가지 전략이 있음

1. **Canonical Order로 Input을 Sorting**

   "While sorting sounds like a simple solution, in high dimensional space there in fact does not exist an ordering that is stable w.r.t.point perturbations in the general sense."

   - Feature가 고차원에 있기 때문에 Point perturbation에 대해 안정적인 순서는 존재하지 않는다.

   "Therefore, sorting does not fully resolve the ordering issue, and it's hard for a network to learn a consistent mapping from input to output as the ordering issue persists."

- 그래서 이 전략으로는 힘들다는 얘기

2. **Input을 Sequence로 간주학, RNN에 Training**

   "The idea to use RNN considers the point set as a sequential signal and hopes that by training the RNN with randomly permuted sequences, the RNN will become invariant to input order."

   - 실제로는 Sequence가 없는 PointSet이지만 N! 순열에도 Invariant한 Feature를 뽑아내기 위해, N! sequence input data를 만들어서 RNN을 돌리자는 얘기

   "While RNN has relatively good robustness to input ordering for sequences with small length (dozens), it's hard to scale to thousands of input elements, which is the common size for point sets"

   - 기존 RNN은 짧은 길이의 Sequence를 다루기 때문에 Ordering 문제에 Robust했으나, 수 천개의 Input element를 가지는 Point Sets에게는 부적합하다.

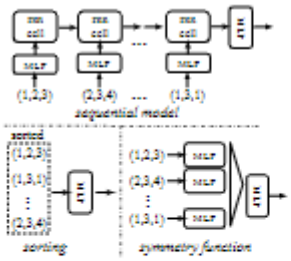
   모든 Permutation을 고려한다면, RNN같은 모델이 Invariant할 수 있지만, N! sequence와 같은 Large scale에 적용 가능성을 보장할 수 없으며, 적용하더라도 PointNet보다는 성능이 구림

3. **Simple symmetric function을 사용해보자**

   Symmetric function : $f(x_1, x_2)$가 symmetric function이면 $f(x_1, x_2) = f(x_2, x_1)$

   - Symmetric function을 이용하면, N개의 vector를 입력 받았을때, Input Order에 변하지 않는 새로운 vector를 출력 가능
   - Symmetric function을 이용하면, 함수 전체를 수식으로 나타낸 General function을 symmetric하게 근사할 수 있고, 결국 permutation-invariant하게 변함.

⇒ Symmetric function으로 Max Pooling을 사용한다. (실제로 성능도 좋음)



	accuracy
MLP (unsorted input)	24.2
MLP (sorted input)	45.0
LSTM	78.5
Attention sum	83.0
Average pooling	83.8
Max pooling	**87.1**

Figure 5. **Three approaches to achieve order invariance.** Multi-layer perceptron (MLP) applied on points consists of 5 hidden layers with neuron sizes 64,64,64,128,1024, all points share a single copy of MLP. The MLP close to the output consists of two layers with sizes 512,256.

2. a local and global information combination structure
    - Segementation Task를 위해선 Global Feature과 Local Feature이 Concatenate(결합)되어야 한다.

3. two joint alignment networks that align both input point and point features
    - For : Semantic labeling of point cloud has to be invariant if point cloud undergoes certain geometric transformation.
    - 두 개의 미니 Network인 T-NET을 사용하는 과정이 포함.
        - Input Transform : Feature Extraction 전에 전체 Input set을 Canonical Space에 정렬
            - Affine Transformation Matrix를 Predict해서 해당 Matrix를 Inputr Points와 행렬 곱을 진행함
            - 마찬가지로 permutation-invariant를 위한 과정임
        - Feature Transform : Extend to alignment of feature space
            - Feature를 Align할 수 있는 Feature Transformation Matrix를 Predict한다.