Meta-PointNet

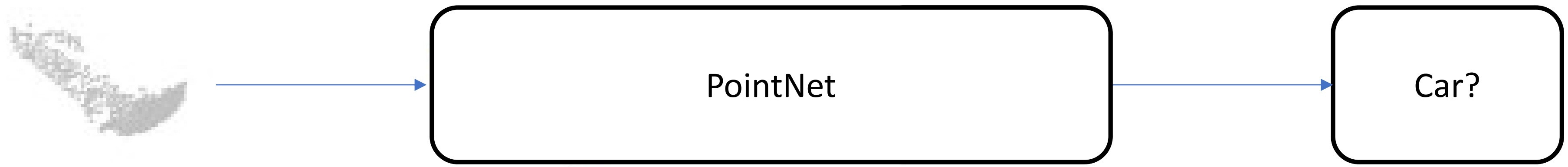
Deep learning for 3D Vision Final Project

2022.06.01

Presenter: SangYeong Jo

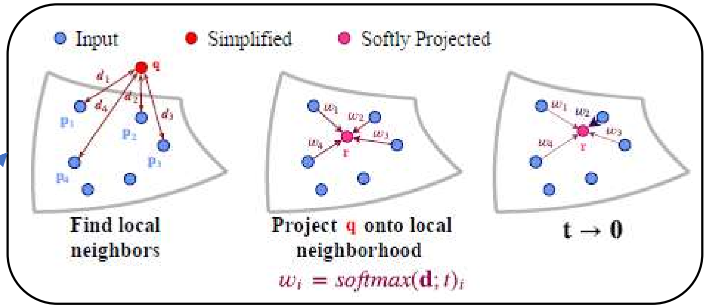
Introduction

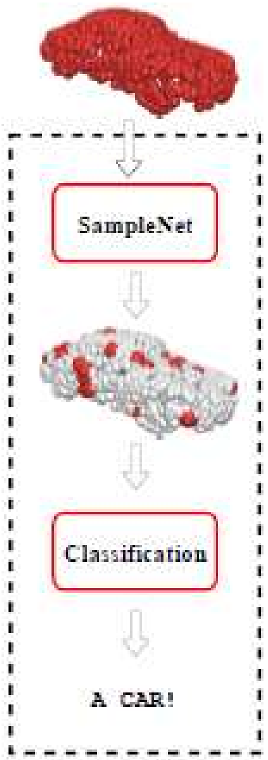
- PointNet
  - Point cloud
    - It is collected set of points from RBG-D or Lidar sensor
    - It is difficult to handle because they are individual, unrelated, and not in regular format
    - It is usually required to be trasnsformed into a 3D voxel grid
  - Input permutation-invariant model
    - Simple symmetric function to aggregate the information
    - Alignment approach for all input data

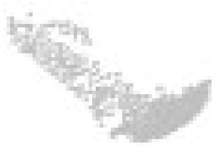


Introduction

- SampleNet
  - Philosophy
    - Reduce the size of point cloud for computational efficiency and communications cost
  - Sampling Method
    - Farthest point sampling
      - Maximal coverage of the input
      - Minimal geometric error
    - **Learnable sampling approach**







SampleNet

Sampling → PointNet → Car?

Introduction

- Model-Agnostic Meta-Learning (MAML)
  - Meta learning approach
    - Few-shot learning
    - Learn to learn
  - Inner loop
    - It learns tasks with updating weights temporally
  - Outer loop
    - It update model parameter using loss from temporal updated weights

Algorithm 2 MAML for Few-Shot Supervised Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
randomly initialize θ
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Sample K datapoints $\mathcal{D} = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ using $\mathcal{D}$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation (2) or (3)
    Compute adapted parameters with gradient descent: $\theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
    Sample datapoints $\mathcal{D}'_i = \{\mathbf{x}^{(j)}, \mathbf{y}^{(j)}\}$ from $\mathcal{T}_i$ for the meta-update
  end for
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i})$ using each $\mathcal{D}'_i$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 2 or 3
end while

Proposed Method

- Meta-PointNet (Proposed)
  - Principle
    - With fewer raw point cloud
    - Random partial samples

Experiment

- Dataset
  - 3D Point Cloud Classification on ModelNet40

- Protocol
  - Same protocol following both PointNet and SampleNet
    - Batch: 32, Learning rate: 0.01, Optimizer: Adam, decay rate: 0.7, etc.

- Comparison
  - In the scenario of fewer number of points
  - In the Scenario of shuffled points

Result

- Comparison
  - Reproducted performance with fewer number of points
  - Meta-PointNet outperforms SampleNet
  - Interesting result
    - The result with number of 512 shows better performance than 1024

Meta-PointNet

# of points	Accuracy
1024	86.5
512	86.8
256	84.1
128	83.9

SampleNet

# of points	Accuracy
1024	81.9
512	82.3
256	81.3
128	78.6

Result

- Comparison
  - Reproducted performance with shuffled point
  - Meta-PointNet outperforms SampleNet

Meta-PointNet

Epochs	# of points	Accuracy
250	1024	85.6
500	1024	85.6

SampleNet

Epochs	# of points	Accuracy
250	1024	81.3
500	1024	81.7

Thank you