Continual Learning for 3D Point Cloud

Kodavanti Rama Sravanth* [1] , Dupati Srikar Chandra [1] & Dr. P.K.Srijith [1]

[1]Indian Institute of Technology, Hyderabad

Abstract

- Point cloud data represents a three-dimensional environment comprising individual data points & it is obtained via LIDAR, it has various applications in Vision,Robotics, AR.
- Robotics,AR rely on real-time processing of point cloud data. As new data points are continuously generating, ML models need to adapt to stream of data flow by forever learning of data.
- This work does continual learning on point cloud which keeps the model up to date. We propose novel approach using knowledge distillation by addressing the problem of catastrophic forgetting.

Methodology

- PointNet architecture is trained incrementally, where the model learns new classes while retaining knowledge of previously learned classes. During each incremental training phase, we introduce new classes and update the model's logits accordingly.
- Crucial part is applying the feature distillation at multiple layers there by guiding the student model by the teacher model.Here we apply distillation at three different layers.

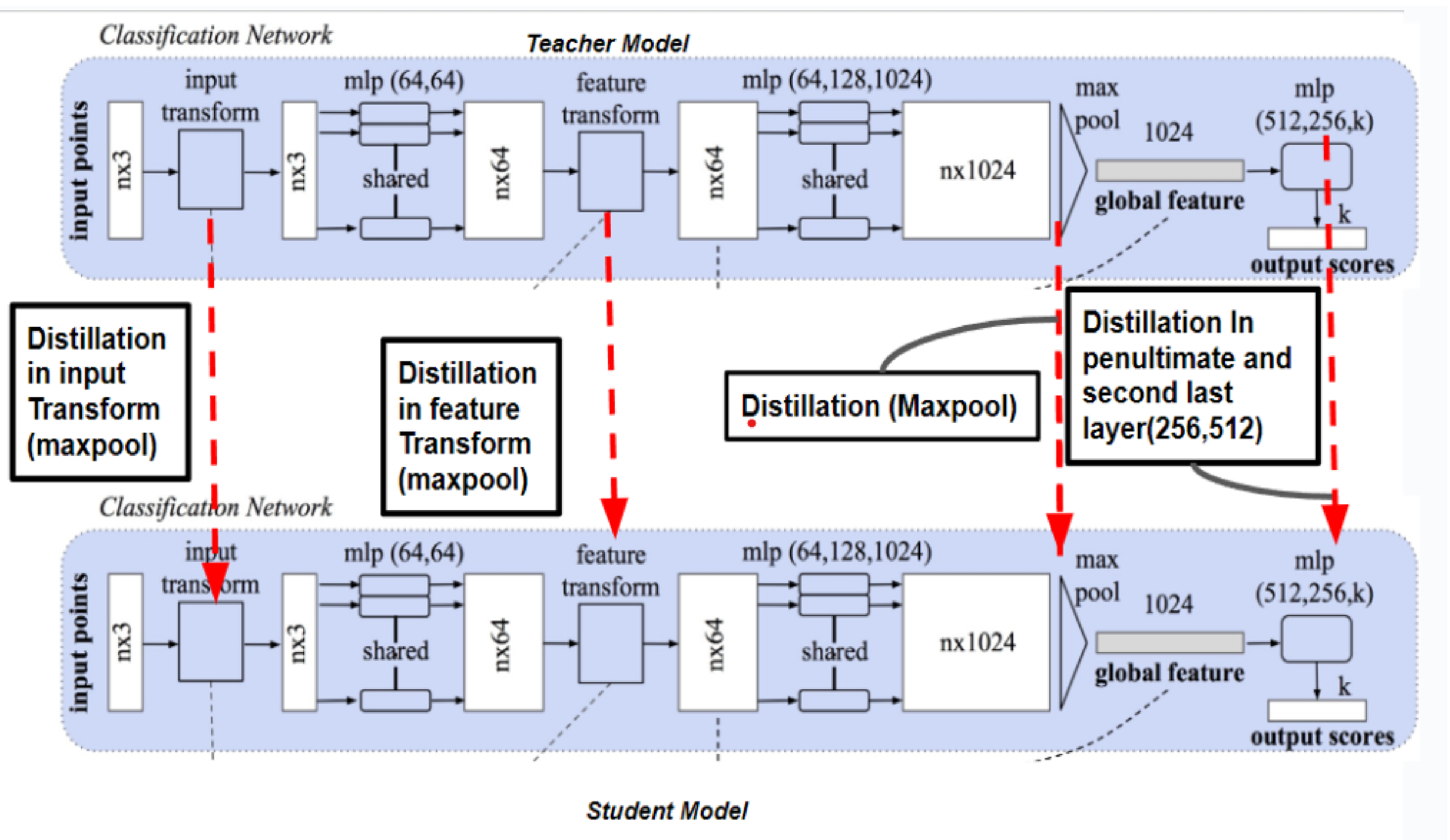


Figure 1. Feature Distillation between teacher & student models

- Training is done through two-fold approach, One is the distillation loss between teacher & student model, While the later is the loss of student model on new classes. These 2 terms preserves trade off between catastrophic forgetting & incremental learning.

Algorithm 1 Continual Learning with Distillation on Point Cloud

Input: Initial dataset D_{t-1}, dataset for task t D_t
Output: Trained model f_t
Initialize the model parameters θ using D_1
for $t = 2$ **to** k **do**
  while *stopping criteria not met* **do**
    Sample a batch from D_t
    Compute penultimate layer features for the batch: $z_{t-1} = \text{penultimate\_layer\_features}(f_{t-1}(x_{t-1};\theta))$
    Compute penultimate layer features for the incremental data: $z_t = \text{penultimate\_layer\_features}(f_t(x_t;\theta))$
    Compute the feature distillation loss: $L_{FD} = \text{feature\_distillation\_loss}(z_{t-1}, z_t)$
    Compute the classification loss using new class logits: $L_{class} = \text{classification\_loss}(\hat{y}_t, y_t)$
    Compute the total loss: $L_{total} = L_{class} + \lambda \cdot L_{FD}$
    Update the model parameters using backpropagation: $\theta \leftarrow \text{backpropagation}(L_{total})$
Procedure: Predicting class label for a new point cloud x_{test}
Compute the model output: $y_{test} = f_t(x_{test};\theta)$
Select the class label with the highest probability: $\hat{y} = \text{argmax}(y_{test})$
return Trained model f_t

Figure 2. Algorithm for incremental continual learning using knowledge distillation

Loss Function

- Loss function contains two terms one is for distillation & another for the loss of student model while learning the new tasks. The trade-off between these 2 terms is given by the parameter α

$$\text{loss} = \text{PointNet loss} + \alpha * \text{Distaillation loss} \tag{1}$$

$$\text{PointNet loss}(\text{outputs}, \text{labels}, m3*3, m64*64, \theta) = -\log(p(y)) + \theta\left(\frac{\|\text{diff3x3}\|_2^2 + \|\text{diff64x64}\|_2^2}{bs}\right) \tag{2}$$

$$\text{distillation loss} = \frac{1}{N}\sum_{i=1}^{N}(z_{t-1}^{(i)} - z_t^{(i)})^2 \tag{3}$$

Optimization Problem

$$\text{minimize } \theta_t\,(L_{\text{PointNet}}(\hat{y}, y, \theta_t) + \alpha \cdot L_{\text{distillation}}(f_{t-1}(x_{t-1}), f_t(x_t), \theta_{\text{student}})) \tag{4}$$

Experimental Setup and Results

- Point Net architecture was trained on Modelnet10 & Modelnet40 datasets which contain 10,40 classes respectively.
- Dataset has been divided into 5 tasks each task consits of 2 classes.Also we have divided dataset into classes whose number of data points are in the same range. Which overcomes catastrophic forgetting
- Also experimented with the training the classes with equal number of data points.The results are as follows:

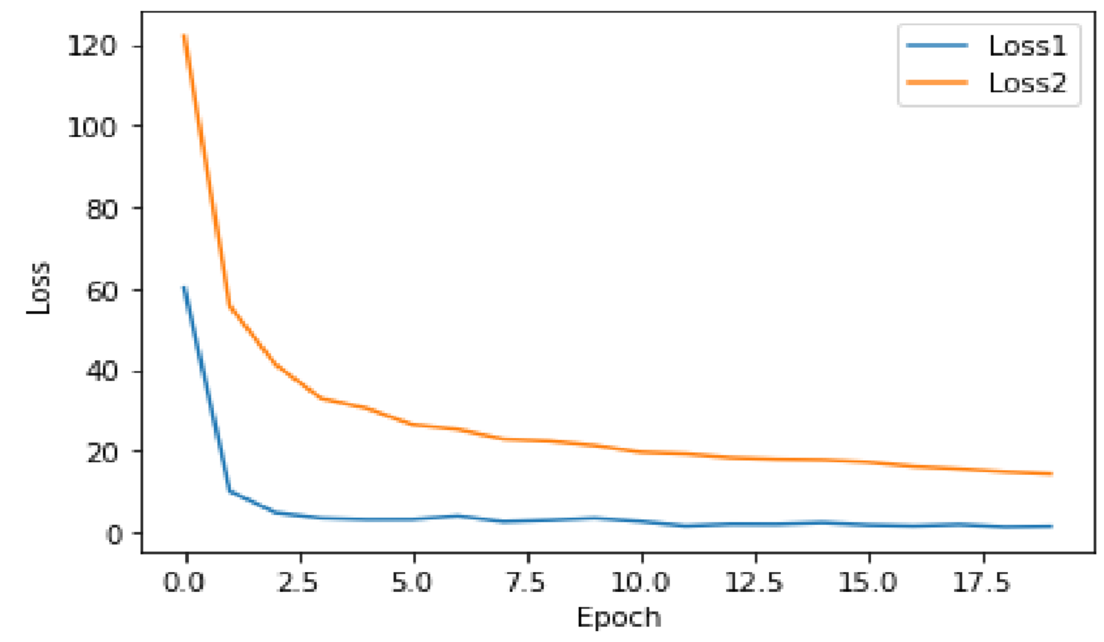


Figure 3. Variation of overall loss function with epochs

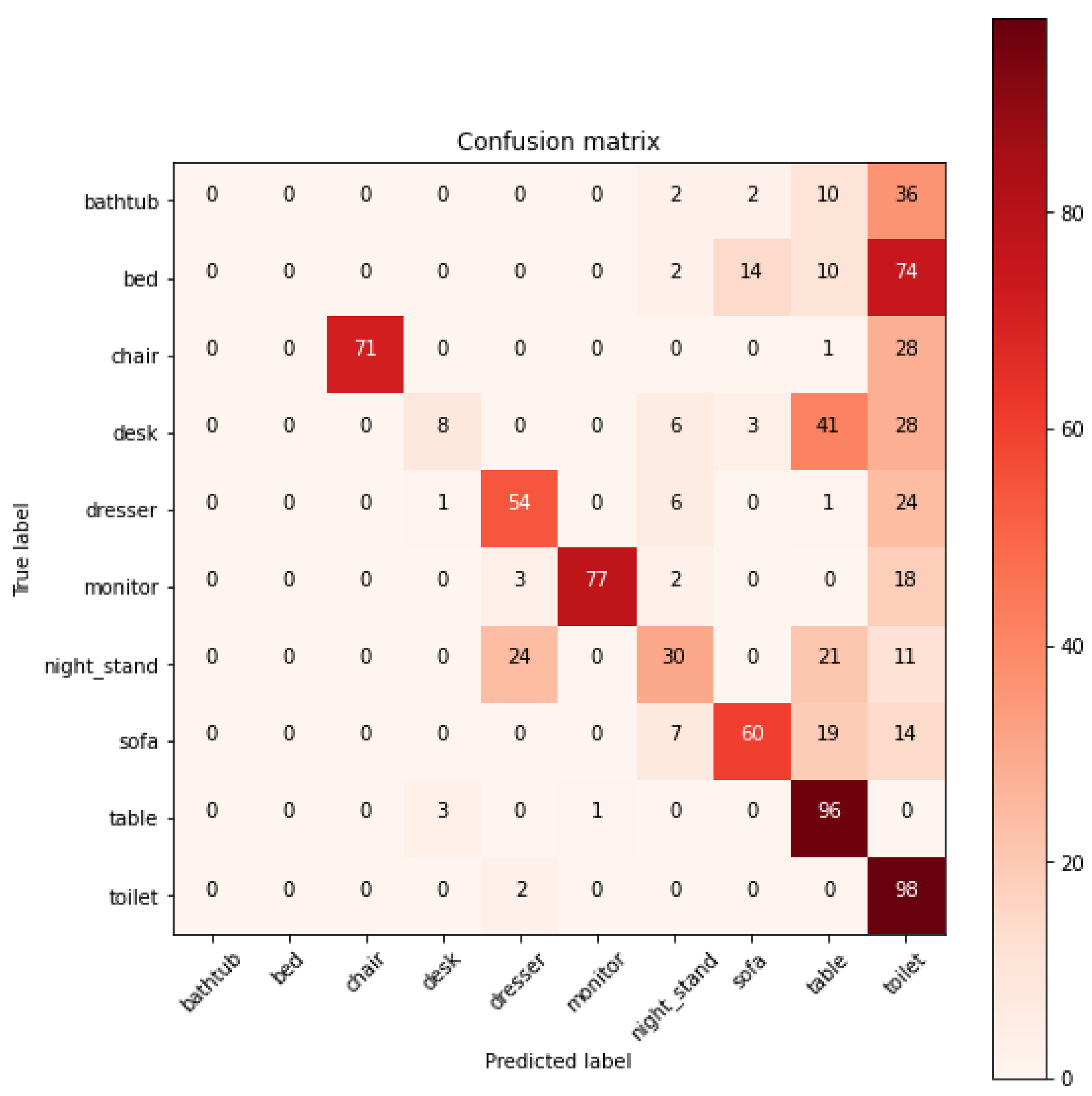
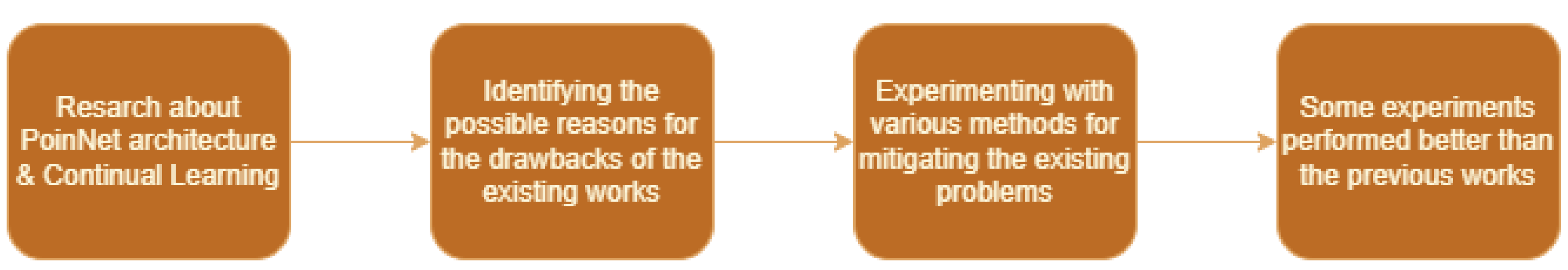


Figure 4. Confusion Matrix with $\alpha = 20$ & learning rate = 0.001

Conclusions & Future Work

Here from the results we can say that model is predicting correct the previous classes while learning the new classes, although there are some instances where model is forgetting this need to addressed by experimenting with other different data setups & implementing new techniques of knowledge distillation.

Timeline of the project



References

[1] Charles R. Qi, Hao Su, "**PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation**" *CVPR*, 2021
[2] Matthias De Lange., Rahaf Aljundi, "**Continual learning survey: Defying forgetting in classification tasks**" *IEEE Multimedia*, Ver.3, 2021.
[3] Maciej Zamorski, Michał Stypułkowski, "**Continual learning on 3D point clouds with random compressed rehearsal**" *CVPR*, 2022

Email: CS20BTECH11027@iith.ac.in