CS-GY 9223G Project Proposal: 3D Point Cloud Classification using Deep Learning

Jiaying Li
New York University
Tandon School of Engineering
jl10919@nyu.edu

Zili Xie
New York University
Tandon School of Engineering
zx979@nyu.edu

1. PROBLEM STATEMENT

Point clouds are important datasets that represent objects or space, which are of great significance in 3D modeling, mapping and reconstruction. Each point represents the X, Y, and Z geometric coordinates of a single point on an underlying sampled surface and sometimes plus extra feature channels such as color. The classification of 3D objects has always been a hot research topic in 3D computer vision. Given hundreds to thousands of three-dimensional coordinate points, how to automatically find the category of such a geometric body among many categories is a complicated problem. This problem is mainly different from ordinary image classification problems in the following aspects.

- First, our input has changed from a four-dimensional image matrix of size $n{*}H{*}W{*}c$ to a three-dimensional vector set of length n, where n is the number of input points. Also, the relationship between 3d points are much subtler than the relationship between 2d pixels.
- For geometric objects, their corresponding categories should remain unchanged under certain geometric transformations, such as rotation, translation, scaling and folding in space. Therefore, we cannot naively judge the category of objects merely through some single perspective.
- Finally, the order of the points of the geometry does not affect the category of the geometry, which means that a 3D object composed of n points, its points can be arranged in at most $n!$ ways, and given any of these $n!$ permutations our classifier should give the same classification result.

2. PRELIMINARY LITERATURE SURVEY

Many classification methods for three-dimensional objects have been proposed by researchers in recent years. In fact, because a geometric body can have many different manifestations, the classification method will be very different according to the different forms of representations. An intuitive method is multi-view input learning. Multiple 2D rendered images or projections on a two-dimensional plane can be obtained from the 3D model. Each 2D image is trained through its own CNN network and then aggregated for pooling. Then set CNN for feature extraction, and finally output the classification of items.[3]

In other studies, 3D objects are discretized into uniform-sized voxel grids, and CNN is directly applied to the three-dimensional objects.[4, 1] Two-dimensional CNN and three-dimensional CNN are used together to learn the properties of geometry, and the outputs of multiple CNNs are fused before generating prediction results.

Finally, it is the learning network pointNet that directly uses the point cloud as input without special processing on the input geometry.[2]

3. DATASETS

- ModelNet is a CAD model database collected and created by the Princeton ModelNet project. The data is manually filtered by human workers from the statistics in the SUN database to decide whether each CAD model belongs to the specified cateogries. There are ModelNet10 datasets and ModelNet40 datasets which are both subset of the project containing respectively 10 and 40 categories. ModelNet40 is also the primary dataset used for training pointNet and the multi-view CNN classifier.
- Sydney Urban Objects Dataset (SUOD) is a dataset containing the 3D models of a variety of common urban road objects across classes of vehicles, pedestrians, signs and trees. The SUOD is the main dataset used by Voxnet project.

4. MODELS

- Multi-View CNNs. The structure of multi-view CNN contains two different CNN networks. First, multiple 2D images are obtained by rendering the input 3D cad model from multiple angles. Each 2D image is trained through a shared CNN network, and then all the data go through a pooling layer for aggregating information from different views, And then set CNN for feature extraction, and finally output the classification of items.

- Voxnet. Voxnet is similar to the 2D CNN network structure: it consists of 2 convolutional layers, 1 maximum pooling layer and 2 fully connected layers. However, before further processing, the point cloud will be transformed into an input-voxel grid. It uses Occupancy Grid Map to represent the occupation probability of each voxel in space (probability of Occupied state), and then feed the modified data to the model as input for training. The output will be a sub-vector of the output class.

- PointNet. The structure of PointNet is very simple, it can directly input point cloud data, and then get the result of classification/segmentation. The model performs independent processing on each point of the disordered point cloud. The key structure in the network is a single symmetric function maximum pooling, which integrates the information of each point in the point cloud. The framework of pointNet consists of two types of micro structures: T-Net, MLP. T-Net is a micro network, which is used to generate an affine transformation matrix to normalize the rotation and translation of the point cloud. This transformation/alignment network is a miniature PointNet. The second structure is n perceptrons that process the point cloud and features. It is capable for increasing the dimension of the point cloud to 64 or 1024.

- All the models listed above will be implemented using Pytorch and Tensorflow.

5. EXPECTATION

- The expected output of this project would be well-trained deep neural network models that has high performance on 3D object classification. Models will be verified on both indoor and outdoor scenes.

- We will compare the performance between PointNet, MultiView CNNs and Voxnet on the ModelNet indoor dataset and Sydney Urban Objects Dataset (SUOD) outdoor dataset in the experiment.

- We will visualize some results of the models output to directly compare their 3D reconstruction ability. Models are supposed to summarize the shape of the object by a sparse set of key points.

- We will use confusion matrix to measure the performance of each model on 3D multi-object classification task. Also we would like to compare their abilities with the overall accuracy and the average accuracy among classes.

6. LOSS FUNCTION

- For this project, we are solving multiclass classification task. Therefore, we use Cross Entropy Loss as our loss function.

- Cross-entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label. So predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high loss value. A perfect model would have a log loss of 0.

- In binary classification, where the number of classes M equals 2, cross-entropy can be calculated as:

$$-(ylog(p) + (1-y)log(1-p))$$

  if $M > 2$, which means for the multiclass classification, we calculated a separate loss for each class label per observation and sum the result.

$$-\sum_{c=1}^{M} y_{o,c} log(p_{o,c})$$

- In the implementation, the loss can be described as:

$$loss(x, class) = -log(\frac{exp(x[class])}{\sum_j exp(x[j])})$$

$$= -x[class] + log(\sum_j exp(x[j]))$$

7. DATA VISUALIZATION

- We have create function to display animated rotation of meshes and point clouds, which will provide us good understanding of the point cloud data.

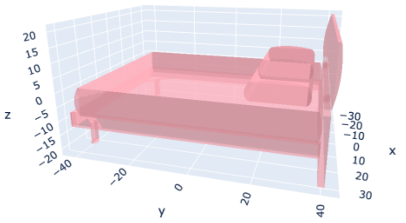

Figure 1. Visualization of the point cloud data in mesh format. This is a point cloud data in ”bed” class

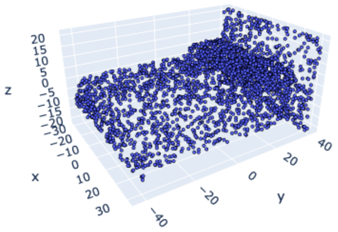

Figure 2. Visualization of the original point cloud data in point cloud format.

8. DATA PREPROCESS

- Sampler. We have write a sampler function to sample points on the surface uniformly for each point cloud data. For PointNet, we chose to sample 1024 points per cloud as in the paper.
- Data augmentation. We perform random rotation of the whole point cloud and add random noise to the points, so as to improve models robustness.
- Normalization. We perform normalization on the points that on the same axis(x, y, z).

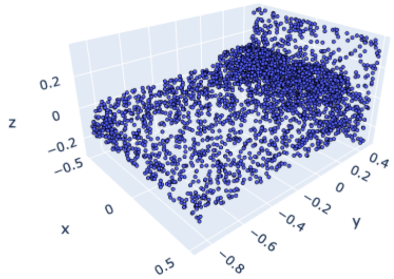

Figure 3. Visualization of the point cloud data after normalization. As we can see, the range of three axes(x, y, z) have changed after the normalization.

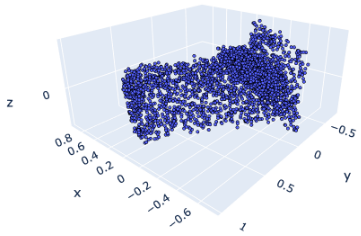

Figure 4. Visualization of the point cloud data after data augmentation. As we can see, more noise appears and it has been rotated

9. TRAINING

- We use gradient descent algorithm in Mini-batches Learning way to train our models.
- We use Adam optimizer for PointNet.
- Learning rate is set as 0.001.
- Batch size is 32 for training process and 64 for vaildation.
- PointNet is trained for 15 epochs.

10. PRELIMINARY RESULTS

- We have finished creating function for loading 3D point cloud data from ModelNet dataset.
- For data preprocessing, we have designed and achieved doing samping, data augmentation, and normalization on 3D point Cloud data.
- We have successfully created visualization for the 3D point cloud, which will help a lot on our analysis on different models performance and also provided us a way to double-check on the correctness on data preprocessing.
- We have built up the training functions and validation functions for the model training.
- We have built up the functions to create confusion matrix and create loss/accuracy curves, which will be used in model performance analysis.
- We have built up PointNet model and successfully trained it on ModelNet dataset and got it performance statics.
- We recorded the accuracy and loss curves for the model in the training and validating process. And we also create the confusion matrix to check model performance on multiple classification task. There are two visions of the confusion matrix, one is normalized and another one is non-normalized.
- For PointNet, it performs well on multiple classification task on ModelNet dataset. As we can see form the loss and accuracy curves, the loss of PointNet is continually decreasing during the training process on training data. Meanwhile, the accuracy of PointNet is continually increasing on the validation data. And from its confusion matrix we can see, PointNet has correctly predicted most of the data. However, PointNet seems perform not very robustly on the class desk and dresser.

11. FUTURE WORK

- So far we have finished training and evaluating PointNet model on ModelNet. In the next couple of weeks, we need to train and evaluating VoxNet and Nulti-View CNN on ModelNet.

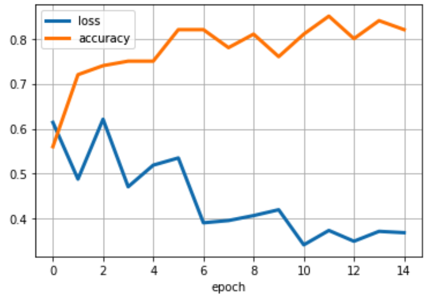

Figure 5. Loss and Accuracy curves of PointNet

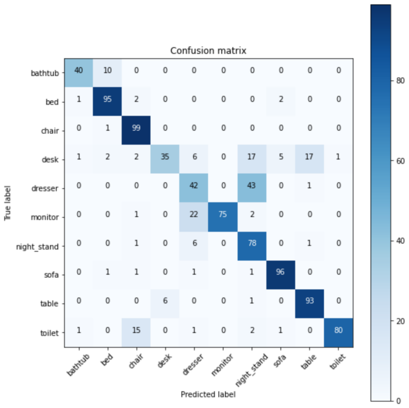

Figure 6. Non-normalized Confusion Matrix of PointNet.

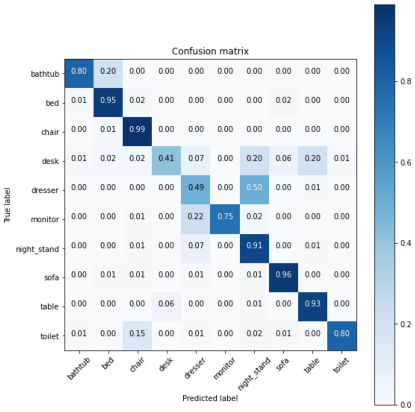

Figure 7. Normalized Confusion Matrix of PointNet.

- After we get the performance of Multi-View CNN, Voxnet and PointNet, we need to do analysis on these difference and try to figure out the reasons behind this and purpose some instructive ideas like in what situation would be better to use which model to performance 3D multiple classification task.

References

[1] M. Nießner A. Dai M. Yan C. R. Qi, H. Su and L. Guibas. Volumetric and multi-view cnns for object classification on 3d data., 2016. In Proc. Computer Vision and Pattern Recognition(CVPR), IEEE. 1

[2] Charles R. Qi* Hao Su* Kaichun Mo Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation., 2017. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 1

[3] H. Su M. Aono B. Chen D. Cohen-Or W. Deng H. Su S. Bai X. Bai et al. M. Savva, F. Yu. Shrec'16 track large-scale 3d shape retrieval from shapenet core55, 2016. 1

[4] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition, 2015. In IEEE/RSJ International Conference on Intelligent Robots and Systems. 1