This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Explaining the PointNet: What Has Been Learned Inside the PointNet?

Binbin Zhang*, Shikun Huang*, Wen Shen†, Zhihua Wei
Department of Computer Science and Technology, Tongji University, Shanghai, China
{1830832, 1830831, 1810068, zhihua_wei}@tongji.edu.cn

Abstract

In this work, we focus on explaining the PointNet [4], the first deep learning framework to directly handle 3D point clouds. We raise two issues based on the nature of PointNet and give solutions. First, we visualize the activation of point functions to examine the issue how global features represent different classes? Then, we propose a derivative of PointNet, named C-PointNet, to generate the class-attentive responce maps to explore that based on what information in the point cloud is the PointNet making a decision? The experiments on ModelNet40 demonstrate the efficacy of our work for getting better understanding of PointNet.

1. Introduction

PointNet is a pioneer in studying deep learning on point sets. Although PointNet and its derivatives [5, 1, 8, 11] have achieved superior performance in various 3D tasks, we cannot explain their representations in a way that humans can understand due to the highly nonlinear nature of deep learning methods. Recently, a growing number of researchers have studied the interpretability of 2D deep learning methods. However, few prior works study the interpretability of deep learning on 3D point clouds.

In [4], per-point functions was visualized to show what have been detected. However, it does not invert the point functions to original point clouds, therefore, we can only see a group of irregular shapes, while can not achieve the visualization results like "the wing of the aircraft can activate a particular point function". In addition, t-SNE was used to embed point cloud global features into a 2D space and visualizes the correlation between point clouds. These work is not enough for explaining what has been learned inside the PointNet.

For this reason, we focus on exploring visual patterns hidden inside the PointNet and extracting point set regions that directly contribute the decision-making. The basic idea of PointNet is to aggregate all per-point features to a global feature and then output classification scores through a multi-layer perceptron (MLP). This basic idea of PointNet raises questions which we are interested in:

- How global features represent different classes?
- Based on what information in the point cloud is the PointNet making a decision?

To examine the first issue, we gain insights on what the learnt per-point function's detect [4] and visualize the activation of the point sets on it. To examine the second issue, we change the network structure of PointNet and make the global feature class-attentive for understanding the decision-making process of PointNet.

2. Related work

2.1. Deep learning on point sets

There has been a growing number of work in recent years in deep learning on point sets [3, 10, 4, 5, 1]. Among them, PointNet [4] is becoming a module similar to a convolution layer and is used directly as the base of many netwoks. Therefore, explaining PointNet is an appropriate breakthrough to interpreting the deep networks for 3D point sets.

2.2. Interpretability for deep learning

Zhang and Zhu [16] roughly define the scope of the interpretability for deep learning into five researh directions: visualization of CNN representations [12, 18], diagnosis of CNN representations [2, 15], disentanglement of CNN representations [14], building explainable models [6], and semantic-level middle-to-end learning [13]. Among all, visualization of CNN representations is the most direct way and is the foundation of interpretability for network. Compared with the well developed interpretability learning for deep learning on 2D data, the interpretability study of deep learning on 3D data is getting started. Some work visualizes the intermediate features by inverting features to point sets [9, 5], which helps people to understand the representations of a network. [4] use t-SNE clustering method to embed features to low-dimensional space and visualize sample correlations. [7] learns kernel correlations to represent

*Equal contribution. †Corresponding author.

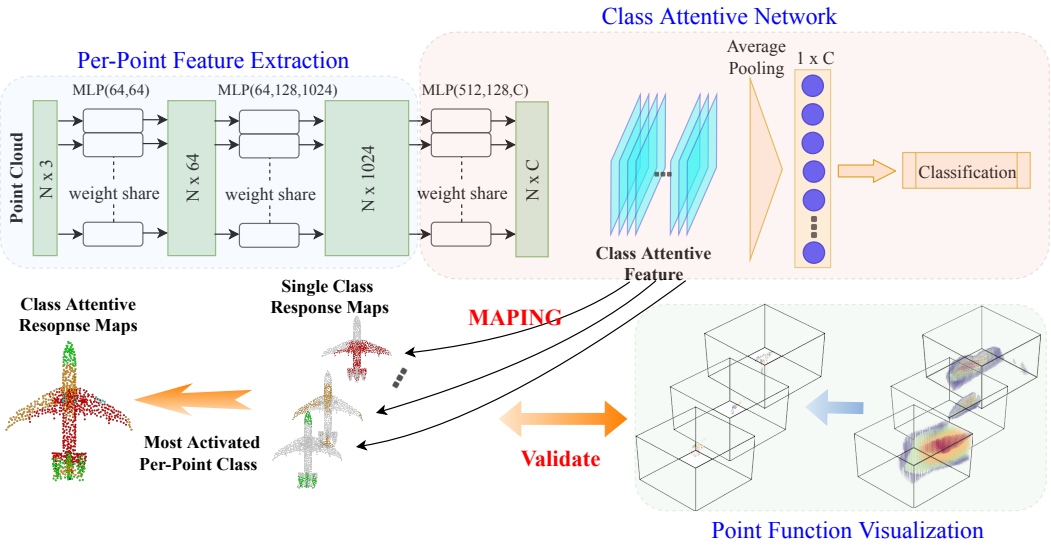

Figure 1. The procedure for explaining the PointNet, including two main parts, visualizing point function and generating class-attentive response maps. First, we visualize the point functions to show what has each point function learnt. Then, we use the proposed C-PointNet to extract the class-attentive global feature, containing per-point feature extraction and class-attentive global feature extraction. Finally, we compute the class-attentive response maps to explore the decision-making process. The class-attentive response maps can verify which point function learns the important information for classification.

complex local geometric structures, which captures various structures (such as plane, edge, corner, concave and convex surfaces). [17] proposes an unsupervised 3D point-capsule network based on the 2D capsule network [6]. By visualizing the iterations of AE training, it finds that fandomly initialized capsules gradually identify meaningful parts.

To the best of our knowledge, no previous work exists that simultaneously visualizes the PointNet representations and understands the decisions made by PointNet.

3. Explaining the PointNet

In this section, we clarify the procedure for explaining the PointNet, as Fig. 1 shows. The procedure can be summarized as follows. First, visualize the point functions (Sec. 3.1). Then, use the proposed C-PointNet to extract the class-attentive global feature, including per-point feature extraction and class-attentive global feature extraction. Finally, compute the class-attentive response maps to explore the decision-making process (Sec. 3.2).

3.1. How global features represent different classes?

This section tackles the first question: *how global features represent different classes?* We think about the question from two different sides: what each dimension of global feature have learned, and how important each dimension of global feature is.

Given an unordered point set $\{x_1, x_2, ..., x_n\}$, PointNet can be defined as a set function f that maps a point set to a vector:

$$f(\{x_1, x_2, ..., x_n\}) = \gamma(MAX\{h(x_1), h(x_2), \ldots, h(x_n)\}) \quad (1)$$

where γ and h are MLP networks.

To analysis what each dimension of global feature have learned, we visulize the activation on each point function h. Unlike [4] that directly visualizes the activation of points of any position, we only show the activation of a particular point set, which can more intuitively reflect what each dimension of global feature have learned for a specific point cloud. For point function h, we visualize the point x_i that $h(x_i) > T$, T is a threshold, to view what the dimension of global feature corresponding to h has learned.

3.2. Based on what information in the point cloud is the PointNet making a decision?

The above issue focuses on what the global feature learns, in this section, we further exploring that *based on what information in the point cloud is the PointNet making a decision?*

Inspired by [2], we modify the PointNet structure to extract class-attentive global features and generate class-attentive response maps, the class-attentive model is named C-PointNet (architecture see Fig. 1).

C-PointNet retains the per-point feature extraction part of PointNet, but change the max-pooling layer. We add a MLP to reduce the dimension of per-point features to the number of classes, and then perform global averaging pooling to generate a categorical output. Each dimension can

be thought of being associated with a particular class. The procedure of generating class-attentive response maps can be summarized as follows. First, extract the per-point class-attentive features $f_{cls} \subseteq \mathbb{R}^{n \times K}$, n and K are the numbers of points and classes respectively. Then, based on f_{cls}, compute the class-attentive map $R(\underline{P})$.

$$R(\underline{P}) = A[R_1(f_{cls}), R_2(f_{cls}), \ldots, R_K(f_{cls})]. \quad (2)$$

where $R_i(f_{cls})$ is a mapping operation that assigns the class-attentive feature values to the origin point sets, the output of which is a single class response map. $A(\cdot)$ finding the class at each point that maximizes the class-attentive feature value.

4. Experiments

4.1. Point function visualization

It has been proved in [4] that different point functions can detect different regions of a point cloud and the different points being deteced by the same point function have different activation values. In this work, we visualize the activated points in different colors according to their activation values, as Fig. 2 shows. We find that each point function can detect a range of points with different activation values. Specific to a point cloud, each point funtion can activate a certain region of the point cloud. Taking airplane as the example, different point functions can activate the wing, the head, the tail, and so on. Another finding is that the activation regions of point funtions are of great overlapping. There are totally 1024 point functions, but a point cloud only has several parts. It is possible that a part can activate a large number of point functions at the same time. The advantage of having a lot of overlapping is that when a point function fails to detect, another point function can still detect the important part. However, a large number of repeated detection generate too much redundant information, which will damage the interpretability of the network.

4.2. Class-attentive response maps

In this section, we train the proposed C-PointNet to generate class-attentive response maps and explore the desion-making process of PointNet. Our C-PointNet achieves remarkable accuracy of 88.0% on Modelnet40 dataset, only 0.7% lower than the retrained PointNet. Without affecting the classification accuracy, the proposed C-PointNet can explore the desion-making process. As shown in Fig. 3, the most points of the correctly classified point clouds are contributing to predicting the groundtruth label, while the points of the misclassified point clouds may contribute to predicting many different labels. Fig. 4 further demonstrates that a pair of point clouds with the similar shape may have the similar class-attentive response maps, which

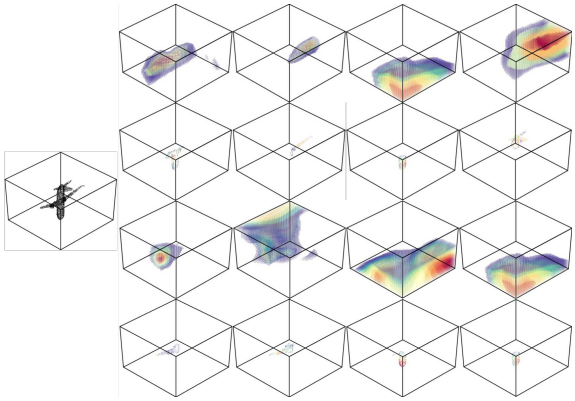

Figure 2. Point function visualization. The first and third rows are the visualization of point functions and the second and forth rows are the point clouds and their activations on the corresponding point functions. For all the points p in a point cloud, we calculate the activation values $h(p)$ of each point function. We visualize the points that $h(p) > 1$ and assign different colors depending on the activation value (red represents large activation value and purple represents small activation value).

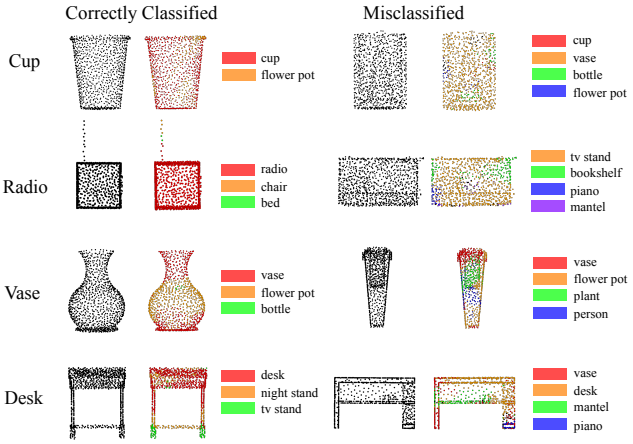

Figure 3. Class-attentive response maps. Each row shows the class-attentive response maps of a certain point cloud class. For each point cloud class, we give correctly classified and misclassified samples. We represent the original point cloud in black dots, represent the points which contribute to predicting the correct class in red dots, and represent other points that contribute to predicting wrong classes in different colors's dots.

means that the two point clouds are easily misclassified to each other.

5. Conclusion

In this work, we explain the PointNet by examining two issues to study what has been learned inside the network. We first visualize the activation of point functions to learn how the global feature of PointNet represents different

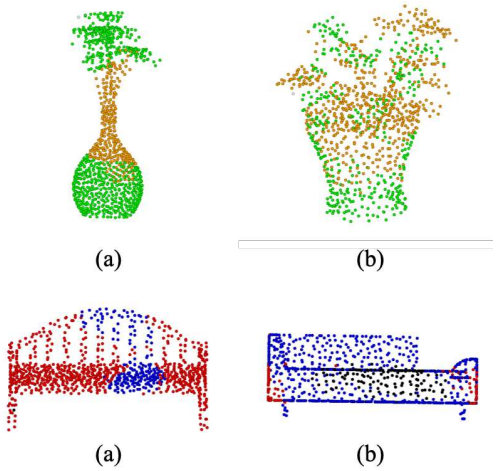

Figure 4. Example point clouds that are easy to be misclassified. (a) *vase* vs. (b) *flower pot* and (c) *bench* vs. (d) *sofa* are two pairs of point clouds that have the similar class-attentive response maps with each other, which are represented by *green* dots vs. *yellow* dots and *red* dots vs. *blue* dots respectively.

classes. Then, we propose the C-PointNet based on PointNet architecture to visualize the regions that affecting the decision-making process. Experiments on ModeNet40 indicate that our method can get better insight into the PointNet while maintain high classification accuracy. In the future, we will exploring our method with different 3D netwroks.

Acknowledgment

The work is partially supported by the National Key Research and Development Project (No. 213), the National Nature Science Foundation of China (No. 61573259) and the Special Project of the Ministry of Public Security (No. 20170004).

References

[1] Mingyang Jiang, Yiran Wu, and Cewu Lu. Pointsift: A sift-like network module for 3d point cloud semantic segmentation. *arXiv preprint arXiv:1807.00652*, 2018.

[2] Devinder Kumar, Alexander Wong, and Graham W Taylor. Explaining the unexplained: A class-enhanced attentive response (clear) approach to understanding deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 36–44, 2017.

[3] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015.

[4] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[5] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017.

[6] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866, 2017.

[7] Yiru Shen, Chen Feng, Yaoqing Yang, and Dong Tian. Mining point cloud local structures by kernel correlation and graph pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4548–4557, 2018.

[8] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2569–2578, 2018.

[9] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018.

[10] Li Yi, Hao Su, Xingwen Guo, and Leonidas J Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2282–2290, 2017.

[11] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2799, 2018.

[12] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[13] Quanshi Zhang, Ruiming Cao, Ying Nian Wu, and Song-Chun Zhu. Mining object parts from cnns via active question-answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 346–355, 2017.

[14] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. Interpreting cnn knowledge via an explanatory graph. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[15] Quanshi Zhang, Wenguan Wang, and Song-Chun Zhu. Examining cnn representations with respect to dataset bias. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[16] Quan-shi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

[17] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3d point-capsule networks. *arXiv preprint arXiv:1812.10775*, 2018.

[18] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.