Weight Loss for Point Clouds Classification

FangYuan Huang¹, Cheng Xu¹,*, XiaoHan Tu¹ and SiQi Li¹
¹College of Computer Science and Electronic Engineering, Hunan University, 410082 Changsha, China
* xucheng@hnu.edu.cn

Abstract. With the widely applying of 3D sensors such as LiDAR and RGBD camera in robots and driverless technology, 3D point clouds classification has also achieved some development. PointNet[1] is a competitive method in point clouds classification due to its fast speed and well performance in real applications. However, when dealing with easy-classified samples overwhelming in 3D object classification, the PointNet shows incompatibility due to its gradients are mainly determined by these easy-classified samples. In order to optimize the gradients in PointNet, we designed the SFL(Sigmoid Focal Loss) function for 3D object classification to instead of the standard softmax cross-entropy loss function. The proposed SFL function will automated refined the weights of the hard-classified samples in training, so that the new neural network can focus on these hard-classified samples to avoid the unbalance of the classifier. Our results show that the new loss function can achieve the best result in classification tasks that directly processes point clouds.

1. Introduction
Since the achievements with deep learning on image, researches on 3D point clouds have also gradually shifted from low-level geometric feature extraction to high-level semantic understanding (point clouds classification, recognition and semantic segmentation). However, the process that the research on deep learning methods of unordered point clouds is slow compared to deep learning for 2D image perception field. The reasons can be categorized into three aspects:
• Unordered. The orders of 3D point clouds are different although the same object uses different equipment or position scanning. It is affected by the collection point clouds equipment and the coordinate system.
• Sparseness. The points of Lidar are very sparse in scenes such as that of robots and autopilots. It is extremely difficult to make the high-level semantic perception based on point clouds due to this sparseness.
• Limited. The data structure of the point clouds is a set of points composed of some 3D point coordinates. The essence is the low-resolution re-sampling of the geometric shape of the 3D world. Thus, it only can provide one-sided geometric information.

There are few previous researches on deep learning of point clouds. PointNet is the groundbreaking work based on directly processes point clouds. The basic idea is the deep network effectively learns a set of optimization functions that select the information points of the point clouds and encodes the reason for its selection. However, PointNet does not handle easy-classified samples overwhelming in the 3D object classification and its gradient is dominated by easy-classified samples. It leads to a result that is not good enough to converge.
To address these problems, we propose SFL (Sigmoid Focal Loss) instead of traditional softmax cross-entropy loss. We show that our loss function achieves better performance in the classification tasks for directly processing point clouds.

The key contributions of our work are as follows:

- We propose the loss function SFL (Sigmoid Focal Loss). It is applied to deal with the gradient that is easily dominated by easy-classified samples in training.
- We then re-construct a new PointNet through the new loss function as show in Figure 1. We show that the new network model in the classification tasks has a better result than the traditional cross-entropy loss function.
- We provide experiments and analysis on the efficiency, and stability of our method.

**Figure 1**: Classification Network. The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The loss function consists of SFL and regularization term.

2. Related work

2.1. A. Feature Representation with 3D data

According to different data representations of 3D data, the existing 3D shape feature representation methods can be divided into four types: the hand-crafted 3D data, the 3D to 2D projection, the 3D voxel and directly processes 3D data.

The handcrafted 3D data: [2, 3] first extracted the low-level features on the 3D shape, and used these features as input of the deep neural network to learn the high-level feature representation. The drawback of this method is that it still relies on the selection of handcrafted features and the optimization of parameters. Thus losing the advantage of deep learning and failing to fundamentally overcome the problems of handcrafted features.

The 3D to 2D projection: [4, 5] projected a 3D shape into a 2D image and then the network architecture with superior performance in the 2D image can be fully utilized. However, the transformation process changes the local and global structure of the 3D shape, resulting in a decrease in feature discrimination. In addition, projecting a 3D shape loses a large amount of structural information.

The 3D voxel: [6, 7] treated 3D shapes as a probability distribution in a 3D voxel grid, thereby expressing it as a binary or real-valued 3D tensor. The advantage of this type of method is that 3D voxels retain the 3D shape information, which could improve the discriminating power of the features. But the proportion of voxels that the 3D shape occupies is high which makes the voxelization results sparse.

Directly processes 3D data: [1] approximated the symmetry function with a MLP network and Max-Pooling, and obtained point-sequential and sensitive feature representations through training. This method is successfully applied to 3D object classification. Our model proposes further improvements based on this approach.
2.2. Loss function for deep learning

Loss function is to quantify how much the prediction deviates from its true result [8]. The smaller the value of loss function is, the better the model fits. Hinge loss [9] function is applied for "maximum interval" classification especially for SVM. Softmax loss function [10, 11] shows well classification performance. Huber loss [12] is proposed to enhance the robustness of the squared loss function for noise (or outliers). The focal loss [13] is designed for dense object detection, and the one-stage detector can achieve the accuracy of the two-stage detector without affecting the original speed. We are inspired by the idea of focal loss and propose SFL (Sigmoid Focal Loss) for point clouds classification tasks.

3. Classification network

Our classification network has three key modules as show in Figure 1: two joint alignment networks that align both input points and point features, the max pooling layer and the new loss function.

We predict an affine transformation matrix by a mini-network (T-net in Figure 1) and directly apply this transformation to the coordinates of input points. The mini-network itself resembles the big network and is composed by basic modules of point independent feature extraction, max pooling and fully connected layers.

The max pooling layer as a symmetric function to aggregate information from all the points. We use a simple symmetric function to aggregate the information from each point.

However, transformation matrix in the feature space has much higher dimension than the spatial transform matrix, which greatly increases the difficulty of optimization. We therefore reconstruct the loss function.

\[
\text{Loss} = \text{SFL} + L_{\text{reg}}
\]

\[
L_{\text{reg}} = \| I - AA^T \|_F^2
\]

Where \( L_{\text{reg}} \) is used to constrain the feature transformation matrix and \( A \) is the feature alignment matrix predicted by a mini-network. We find that by adding the regularization term, the optimization becomes more stable and our model achieves better performance.

4. SFL (Sigmoid Focal Loss)

The SFL is designed to address easy sample overwhelming. We introduce the SFL from cross-entropy loss.

\[
\text{CE}(p, y) = -\frac{1}{n} \sum_x [y \log p + (1-y) \log (1-p)]
\]

\( p \in [0, 1] \) is the probability of the network output for each sample and \( y \) is the ground-truth class. The cross-entropy cost function has two properties: Non-negative (our goal is to minimize the cost function); When the true input \( p \) is close to the target label \( y \), the cost function is close to 0. (eg, \( y = 0, p \sim 0; y = 1, \text{when } p \sim 1, \text{the cost is close to 0})

4.1. Focus parameter \( \gamma \)

We reshape the cross-entropy function to reduce the weight of easy-classified samples, so that the training is concentrated on hard-classified examples. We introduce a hyperparameter \( \gamma \) is to balance the weight.

\[
\text{SFL}(p, y) = -\frac{1}{n} \sum_x [y (1-p)^\gamma \log p + (1-y) p^\gamma \log (1-p)]
\]

When a sample is correctly classified, \( p \rightarrow 1 \), then the factor \( (1-p) \rightarrow 0 \); otherwise when \( p \) is small, the factor \( (1-p) \) is close to 1. The weight of easy-classified samples is reduced. An increase of \( \gamma \) can enhance the influence of the modulation factor. The modulation factor reduces the loss contribution of the easy-classified samples.
4.2. Weight parameter $\alpha$
We also use the $\alpha$-balanced variant of the cross-entropy loss. The weight parameter $\alpha \in [0, 1]$ is for \( y=1 \), \( 1-\alpha \) for others. In practical applications, $\alpha$ is generally set as the inverse of the class frequency or as a hyperparameter. We found that it can improve a little bit in the classification precision.

$$SFL(p, y) = -\frac{1}{n} \sum_{x} \left[ \alpha y (1 - p)^{\gamma} \log p + (1 - \alpha)(1 - y) p^{\gamma} \log(1 - p) \right]$$

(5)

4.3. Activation function sigmoid
We use the sigmoid function for the final layer of the network. In general, softmax is used as the final output of the network with cross-entropy loss function to solve multiple classification problems as the probability of the prediction category. But here we use sigmoid function in the loss because it's just a binary classification in each sample for one category. In addition, using sigmoid activation is more straightforward.

$$p = \frac{1}{1 + e^{-x}}$$

(6)

5. Experiment
In this section, we will discuss the experimental setups to evaluate the results of our approach compared to the softmax cross-entropy loss function. We also analyze the efficiency and the stability of the new model in 3D point cloud classification.

5.1. Result comparison to the softmax cross-entropy loss
We evaluated the performance of our method on the ModelNet40. There are 12311 CAD models from 40 man-made objects categories, which include 9843 for training and 2468 for testing.

The SFL introduces two new hyperparameters; the focusing parameter $\gamma$ and the balance parameters $\alpha$. Good performance is based on the empirical values, $\gamma=2.0$. When $\gamma = 2.0$, we compared the overall accuracy of SFL and softmax cross-entropy loss as show below.

| $\alpha$ | accuracy overall | accuracy avg. class |
|---------|------------------|---------------------|
| Softmax CE $^a$ | 89.2 | 86.2 |
| 0.30     | 89.5 | 86.5 |
| 0.35     | 89.0 | 86.4 |
| 0.40     | 89.0 | 86.2 |
| 0.50     | 89.4 | **87.0** |
| 0.60     | 89.1 | 86.0 |

$^a$ Softmax cross-entropy loss

We find the best $\alpha$ for $\gamma$ as show in Table 1. We find the best performance is $\gamma = 2.0$ with $\alpha = 0.5$ for all experiments (we tested $\alpha \in [0.3, 0.6]$). Compared with softmax cross entropy loss, SFL improves 0.8% in the average class accuracy. We also found that almost the average class accuracy has been improved compared to the softmax cross-entropy loss. SFL can effectively balance the samples loss and focus all attention on the hard-classified samples.
5.2. **The efficiency and stability for training**

![Figure 2](image1.png)

**Figure 2.** The relationship between loss value and epoch. SFL loss decays faster than the softmax cross-entropy loss.

![Figure 3](image2.png)

**Figure 3.** The relationship between loss value and accuracy. SFL is more stable than softmax cross-entropy loss during training.

To evaluate the efficiency and stability of the training for SFL, we set epoch=250, batch size=32, decay rate=0.7, and learning rate=0.001 to study the speed at which the loss value decays and the stability of the decay process. In Figure 2, we find that the SFL loss decays faster than the softmax cross-entropy loss. The more accurate the prediction is, the faster the loss of the sample decreases.

SFL is more stable than softmax cross-entropy loss in training as shown in Figure 3. For hard-classified samples, the SFL attenuation will be more stable and approach the exact classification result.

6. **Conclusion**

In this work, our primary challenge is that PointNet does not handle the problem of easy-classified samples overwhelming in the 3D object classification. And we propose the SFL for the model concentrates on hard-classified samples in training. We evaluated that our approach has better results compared to the softmax cross-entropy loss function. In addition, we analyze the efficiency and the stability of the new model in 3D point clouds classification.

**Acknowledgements**

This work was supported by the National Natural Science Foundation of China with Grand Nos. 61772185. Thanks to all of the relevant persons.

**References**

[1] Charles, R. Q., Su, H., Mo, K., & Guibas, L. J. 2016* Pointnet: deep learning on point sets for 3d classification and segmentation.* 77-85.

[2] S. Bu, Z. Liu, J. Han, J. Wu, R. Ji. 2014*Learning High-Level Feature by Deep Belief Networks for 3-D Model Retrieval and Recognition.* IEEE TMM, 16(8): 2154-2167.

[3] Jin, X., Dai, G., Fan, Z., Wong, E., & Yi, F. 2017* Deepsshape: deep-learned shape descriptor for 3d shape retrieval.* IEEE Transactions on Pattern Analysis & Machine Intelligence, 39(7), 1335-1345.

[4] Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. 2016* Multi-view Convolutional Neural Networks for 3D Shape Recognition.* IEEE International Conference on Computer Vision pp.945-953.

[5] Qi, C. R., Su, H., Niebner, M., Dai, A., Yan, M., & Guibas, L. J. 2016* Volumetric and multi-view cnns for object classification on 3d data.* 5648-5656.

[6] Maturana, D., & Scherer, S. 2015* VoxNet: A 3D Convolutional Neural Network for real-time object recognition.* Ieee/rsj International Conference on Intelligent Robots and Systems pp.922-928.

[7] Y. Li, S. Pirk, H. Su, C. R. Qi, and L. J. Guibas. 2016* Fpnn:Field Probing Neural Networks for 3D Data.* arXiv preprint arXiv:1605.06240.
[8] Christoffersen, P., & Jacobs, K. 2004 *The importance of The Loss Function in Option Valuation*. Journal of Financial Economics. 72(2), 291-318.

[9] Rosasco, L., De Vito, E., Caponnetto, A., Piana, M., Verri, A. 2004 *Are loss functions all the same Neural*. Comput. 16(5), 1063–1076.

[10] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, V., Vanhoucke, Rabinovich, A. 2015 *Going deeper with convolutions*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9

[11] Simonyan, K., & Zisserman, A. 2014 *Very deep convolutional networks for large-scale image recognition*. Computer Science.

[12] Lu, Z. Q. J. 2010 *The elements of statistical learning: data mining, inference, and prediction*. Mathematical Intelligencer, 99(466), 567-567.

[13] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017 *Focal loss for dense object detection*. arXiv preprint arXiv:1708.02002.