불균형 데이터세트 학습에서 정확도 균일화를 위한 학습 방법에 관한 연구

박근표 [1], 박흠우 [2], 김종국 [3]
[1] 고려대학교 전자전자공학과 석사과정
[2] 고려대학교 전기전자공학과 박사과정
[3] 고려대학교 전기전자학부 교수

gppark@korea.ac.kr, xypiao97@korea.ac.kr, jongkook@korea.ac.kr

A Study of a Method for Maintaining Accuracy Uniformity When Using Long-tailed Dataset

Geun-pyo Park, XinYu Piao, Jong-Kook Kim
School of Electrical Engineering, Korea University

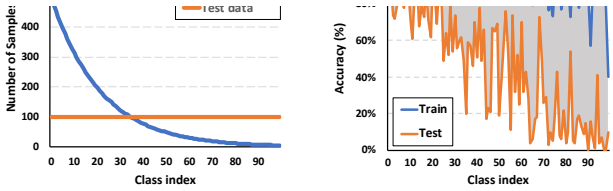
요 약

Long-tailed datasets have an imbalanced distribution because they consist of a different number of data samples for each class. However, there are problems of the performance degradation in tail-classes and class-accuracy imbalance for all classes. To address these problems, this paper suggests a learning method for training of long-tailed dataset. The proposed method uses and combines two methods; one is a resampling method to generate a uniform mini-batch to prevent the performance degradation in tail-classes, and the other is a reweighting method to address the accuracy imbalance problem. The purpose of our proposed method is to train the learning models to have uniform accuracy for each class in a long-tailed dataset.

1. Introduction

Modern real-world datasets have an imbalanced data distribution, where some classes have less data compared to other classes (is shown in figure 1a), known as long-tailed datasets. Unfortunately, recent deep learning models such as ResNet and ViT models, which are widely used learning models in visual recognition tasks, perform poorly on long-tailed datasets. Because the number of data samples in tail-classes in training data is not enough to train, the model shows significant performance degradation for tail-classes in test dataset. There is also a problem of significant accuracy differences between classes. Figure 1b shows that accuarcy in tail-classes shows a significant lower than head-classes. We observed that accuracy may be related with data distribution, and this observation is also introduced in [1]. In this paper, we define and formulate this observation as a *class-accuracy imbalance* problem.

There are many approaches to address these problems. Resampling method (e.g., [2], [3]) is a widely used method in recent year, and there are two main methods; 1) over-sampling the tail-classes that have a few samples, and 2) under-sampling the head-classes. The other method is the cost-sensitive re-weighting method (e.g., [4], [5], [6]), which assigns adaptive weights to different classes when calculating loss. Adaptive weights are determined by the number of samples for each class. However, the above approaches still show weaknesses in class-accuracy imbalance problem.



(a) Difference in distribution between train- and test data

(b) Results for the degree of accuracy imbalance by class index

Figure 1: Problems in the long-tailed dataset

To address this problem, this paper proposes a learning method that combines resampling and reweighting methods. Our proposed method works two additional compuations during training; 1) resampling phase generates a uniform mini-

batch to reduce the impact of the imbalanced data distribution from a given dataset, and 2) reweighting phase provides a different weight for each class to prevent overfitting problems when calculating loss. The proposed method allows the learning models to be trained as if using uniform distributed datasets and prevents the class-accuracy imbalance problem.

2. Proposed Approach

2.1. Resampling phase

For accuracy uniformity, our proposed method uses a different resampling method to reduce the impact of the imbalanced data distribution from long-tailed dataset to learning models during training. This resampling method generates a mini-batch that has a uniform data distribution for all classes. Resampling scheme ($\boldsymbol{F_R(s)}$) and mini-batch ($\boldsymbol{B}$) can be formulated as follows:

$$F_R(s) = \bigcup_{i=0}^{n-1} f_R(\mathbb{C}_i, s), \qquad s > 0 \tag{1}$$

$$B = F_R(s) \tag{2}$$

where $\boldsymbol{s}$ means the number of data samples to select from each class, $\boldsymbol{n}$ is the total number of classes. In Equation 1, $\boldsymbol{f_R(\mathbb{C}_i, s)}$ means a function that selects $\boldsymbol{s}$ samples from the set of i-th class, where $\mathbb{C}_i$ means the set of i-th class. According to equation 1, a mini-batch contains at least one data sample from all classes and has a uniform distribution. Thus, head-classes are selected as if using under-sampling and tail-classes are selected as if using over-sampling. Note that resampling in our proposed method selects the same number of data samples for all classes without increasing the total number of data samples, such as over-sampling method.

2.2. Reweighting phase

To address class-imbalance problem, our proposed method adopts re-weighting scheme introduced in [5] and modifies reweighting scheme as follows:

$$\mathcal{W}_i = \frac{N_{total}}{N_i \times n} \tag{3}$$

$$\mathcal{L} = \frac{1}{s \cdot n} \cdot \sum_{i=0}^{C-1} (-\log(P_i) \times \mathcal{W}_i) \tag{4}$$

where $\boldsymbol{\mathcal{W}_i}$ is the weight, $\boldsymbol{N_i}$ is the number of samples for i-th class, $\boldsymbol{N_{total}}$ is the total number of samples for a given dataset. In equation 4, $\boldsymbol{\mathcal{L}}$ means the value of loss for one mini-batch. According to the equation 3, the weight is larger than 1 in tail-classes and less than 1 in head-classes. Therefore, a large weight can prevent the overfitting problem due to fewer data samples when training tail-classes. Also, a large weight has the same effect as providing a large error margin to tail-classes, and this large error margin provides many opportunities for learning model to predict tail-classes. Without any additional computations in equation 3, however, the weight will be larger than 10 or less than 1. We found that learning model fails to converge to the global minima when using too large or small weights during training. Thus, we limited weights ($\mathcal{W}_i$) to $\mathcal{W}_i \in [1, \sqrt[4]{n}]$ in our experiments.

2.3. Experimental Setup

System	CPU	AMD Ryzen 9 5950X 16-Core
	GPU	Geforce RTX 2080 Ti (12GB)
	RAM	128GB
Model	ResNet-32 ([1], [4], [5])	
Dataset	CIFAR10-LT (Imbalance factor: 50, 100, and 500)	
Optimizer	SGD (lr: 0.1, weight decay: $5e^{-4}$, momentum: 0.9)	
LR scheduler	Cosine Annealing LR	
Loss Functions	Softmax	$\mathcal{L}_i = -\log(P_i)$ w/ RS
	CB-Softmax [5]	$\mathcal{L}_i = -\frac{1}{E_{n_y}} \cdot \log(P_i)$ w/ RS
	Ours	$\mathcal{L}_i = -\mathcal{W}_i \cdot \log(P_i)$ w/ BS

Table 1: Experimental setup. RS means Random Sampling method, and BS means Balanced Sampling method introduced in this paper.

We run our experiments on the system shown in table 1. CIFAR10-LT dataset is re-configured by using imbalance factor 10 and 100. Imbalance factor in table 1 is the degree of imbalance calculated by dividing the maximum number of samples in head-classes from the minimum number of samples in tail-classes. If imbalance factor is 50, it means that the number of samples between head-class and tail-class differs by 50 times. Note that the difference in the number of data samples between classes increases as the value of imbalance factor increases. The proposed method is implemented in PyTorch 1.10.1 and CUDA 11.3 version.

2.4. Experimental Results

Peformance Degradation Figure 2 shows top-1 accuracy for each method and imbalance factor. In this result, we can observe that our proposed method performs higher accuracy and can prevent the performance degradation than other methods in CIFAR10-LT datasets. Especially, in imbalance factor 500, our method shows 16.22% and 20.80% higher accuracy compared to Softmax and CB-Softmax methods, respectively.

Class-accuracy Imbalance Figure 3 shows accuracy per class index for each method in ResNet-32. Results show overall comparable performance and the result in CIFAR10-LT with imbalance factor 500 shows the best performance compared to other methods. In imbalance factor 500, other methods show significant performance degradation in tail-classes because they are trained a given dataset that consists of imbalanced data distribution. On the other hand, accuracy of tail-classes that have fewer samples is lower than head-

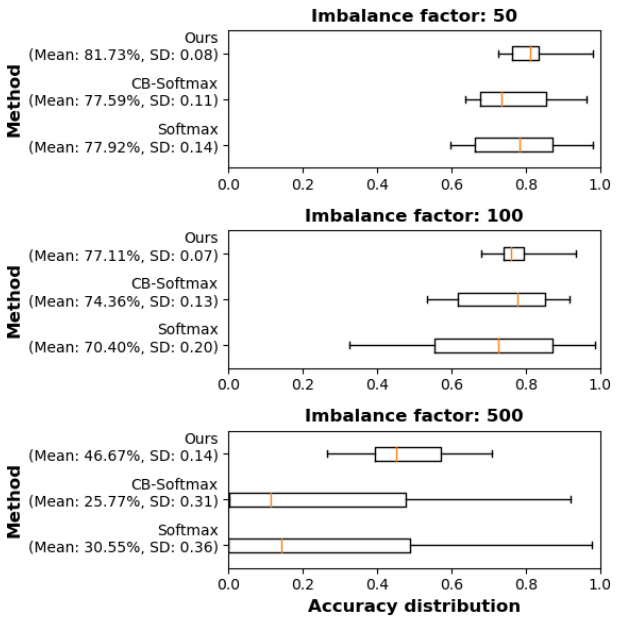
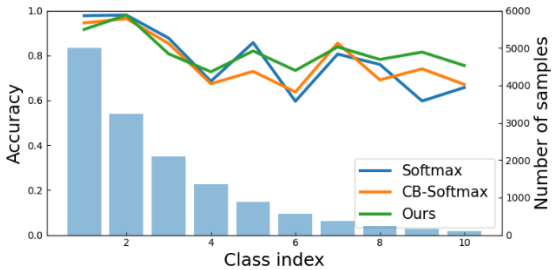
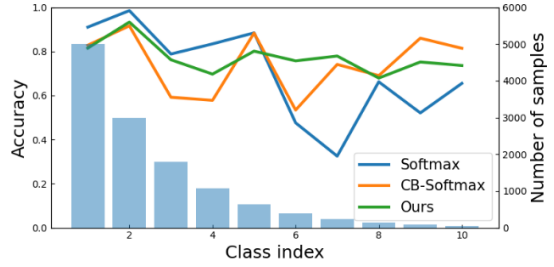


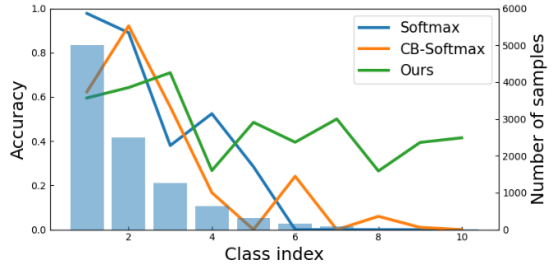
Figure 2: Results show top-1 accuracy (Mean), standard deviation (SD) showing the accuracy difference between classes, and distribution of accuracy (Box plot) for each method and imbalance factor.



(a) Imbalance factor: 50



(b) Imbalance factor: 100



(c) Imbalance factor: 500

Figure 3: Degree of class-accuracy imbalance for each method.

classes, but our proposed method shows higher accuracy in tail-classes than other methods.

In order to trust learning models are trained well about all classes, it is important to evalute the degree of class-accuracy imbalance in long-tailed datasets. Thus, we use two ways to evaluate the degree of class-accuracy imbalance in more detail; one is standard deviation that presents the degree of accuracy distribution as a single value, and the other is a box plot that visualizes accuracy distribution for all classes (are shown in figure 2). In this result, the proposed method shows 0.22 and 0.17 lower standard deviation compared to Softmax and CB-Softmax methods in imbalance factor 500, respectively. Our method also shows the smallest accuracy distribution than other methods for each imbalance factor.

Summary Above results show that our proposed method performs well and is useful to train learning models in CIFAR10-LT dataset than other methods. It is effectively to train tail-classes that have fewer samples and achieves the highest performance compared to other methods.

3. Conclusion

This paper proposes a learning method, which combines the resampling and reweighting methods, to address problmes of performance degradation and class-accuracy imbalance that are occurred in the long-tailed dataset. Results show that our proposed method performs well in CIFAR10-LT dataset and effectively solves the problems of class-accuracy imbalance and performance degradation occurred in long-tailed datasets. In the future, we plan to extend our research to other vision datasets, such as Flower102 or iNaturalist2018 datasets.

Reference

[1] Hong, Youngkyu, et al. "Disentangling label distribution for long-tailed visual recognition." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021.

[2] Estabrooks, Andrew, Taeho Jo, and Nathalie Japkowicz. "A multiple resampling method for learning from imbalanced data sets." Computational intelligence 20.1 (2004): 18-36.

[3] Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.

[4] Elkan, Charles. "The foundations of cost-sensitive learning." International joint conference on artificial intelligence. Vol. 17. No. 1. Lawrence Erlbaum Associates Ltd, 2001.

[5] Cui, Yin, et al. "Class-balanced loss based on effective number of samples." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.

[6] Ren, Jiawei, et al. "Balanced meta-softmax for long-tailed visual recognition." Advances in neural information processing systems 33 (2020): 4175-4186.