Inverse Weight-Balancing for Deep Long-Tailed Learning

Wenqi Dang[1], Zhou Yang[1], Weisheng Dong[1]*, Xin Li[2], Guangming Shi[1,3]

[1]XiDian University, China
[2]West Virginia University, America
[3]Peng Cheng Laboratory, China
{wqdangxdu, yangzhouxdu}@gmail.com, wsdong@mail.xidian.edu.cn,
xin.li@mail.wvu.edu, gmshi@xidian.edu.cn

Abstract

The performance of deep learning models often degrades rapidly when faced with imbalanced data characterized by a long-tailed distribution. Researchers have found that the fully connected layer trained by cross-entropy loss has large weight-norms for classes with many samples, but not for classes with few samples. How to address the data imbalance problem with both the encoder and the classifier seems an under-researched problem. In this paper, we propose an inverse weight-balancing (IWB) approach to guide model training and alleviate the data imbalance problem in two stages. In the first stage, an encoder and classifier (the fully connected layer) are trained using conventional cross-entropy loss. In the second stage, with a fixed encoder, the classifier is fine-tuned through an adaptive distribution for IWB in the decision space. Unlike existing inverse image frequency that implements a multiplicative *margin adjustment* transformation in the classification layer, our approach can be interpreted as an adaptive *distribution alignment* strategy using not only the class-wise number distribution but also the sample-wise difficulty distribution in both encoder and classifier. Experiments show that our method can greatly improve performance on imbalanced datasets such as CIFAR100-LT with different imbalance factors, ImageNet-LT, and iNaturelists2018.

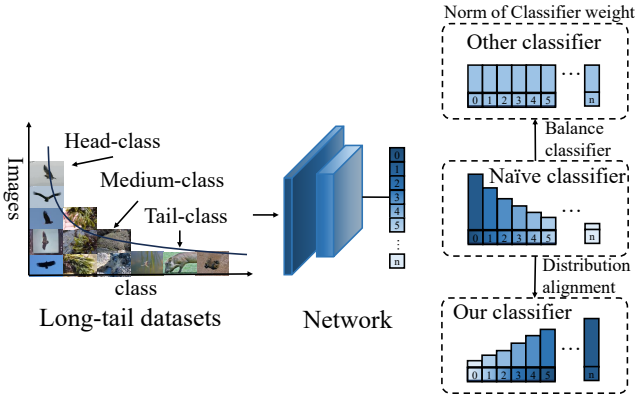


Figure 1: Inverse weight-balancing (IWB) for long-tailed learning (note the opposite trends in naive and our classifier weight distribution). Previous studies balance the weight-norm classifier to cope with long-tailed datasets, ignoring the imbalance problem in the encoder. We use IWB-based adaptive distribution alignment to reverse the weight-norms distribution and compensate for the imbalance in both the encoder and the classifier.

Introduction

A great deal of progress has been made in image recognition with the development of deep learning (Krizhevsky, Sutskever, and Hinton 2017; He et al. 2016), but all of these are based on relatively perfect datasets. In the presence of imbalanced datasets, deep convolutional neural networks will perform poorly. Imbalanced data in the real world is a common problem, particularly in the medical field. We may not be able to obtain enough relevant data for some rare diseases, resulting in a data imbalance between head-class and tail-class. Thus, solving the data imbalance problem, also known as deep long-tailed learning, has great practical significance (Zhang et al. 2021c).

In traditional classification methods, deep convolutional neural networks tend to focus on head-class. They are more accurate because they contribute more gradients than other classes. Tail-class accuracy is often poor because of the small number of samples and too little attention from the network. Oversampling tail-class and downsampling head-class is the simplest way to solve the long-tail problem. Oversampling (Joloudari et al. 2023; Han, Wang, and Mao 2005; Feng, Zhong, and Huang 2021) generates new samples from existing samples. Having few tail samples reduces the diversity of generated samples, so the overfitting problem is easily aggravated and an effective generation method cannot be determined. Furthermore, this method of generating additional data introduces additional costs in both the generation and training phases. When downsampling (Drummond, Holte et al. 2003; Estabrooks, Jo, and Japkowicz 2004) is performed, there is often an issue of information loss during model training.

In this paper, we propose a new method called inverse weight-balancing (IWB) to address the problem of imbalanced data. In previous studies (Kang et al. 2019; Alshammari et al. 2022), cross-entropy loss classifiers are found to be imbalanced, as the weight-norms for the head-class are

*Corresponding Author.

higher than those of the tail-class (see Figure 1). As shown in (Zhang et al. 2021c), it is often more difficult to classify samples in the tail-class than those in the head-class. To solve this unfairness issue, existing studies (Alshammari et al. 2022; Kang et al. 2019) increase the weight-norms related to the tail-class and decrease the weight norms-related to the head-class simultaneously. The hope is to balance the classifier's weight norms using the two-stage method (Alshammari et al. 2022; Kang et al. 2019), while overlooking the imbalance in the encoder. As shown in Figure 2, as the degree of imbalance in the dataset intensifies, the features of tail-class become more dispersed. It is desirable to compensate for the imbalance in the encoder by "overbalancing" the classifier. Specifically, we propose a learnable target inverse distribution to inverse the distribution of weight-norms, as shown in Figure 1.

To better explain the mechanism of IWB, we resort to the concept of decision space. The features after the encoder are distributed in the same latent space L with a fixed size. The classifier parameters divide this latent space into C subspaces (C is the number of classes). In the case of long-tailed distribution, a larger weight-norms implies a larger decision space, which means that the tail-class occupies less space. This is disadvantageous to the tail-class, leading to poor classification performance. Note that a large decision space is not needed for the head-class because the features of the head-class are often more concentrated than those of the tail-class, as shown in Figure 2. An important new insight brought by this work is that a more effective use of decision space should make the decision boundary closer to the head-class. Due to the imbalance of the dataset, the features of the tail-class are more scattered and require more space than the head-class. To achieve this objective, IWB increases the weight-norms associated with the tail-class and decreases the weight-norms associated with the head-class. By inversing weight-norms distribution of classifier, we compensate for the imbalance in the encoder. Extensive experimental results have shown that our method performs the best on the CIFAR100-LT dataset. It is simple and effective compared to other methods such as knowledge distillation (Iscen et al. 2021; Li, Wang, and Wu 2021; Li et al. 2022; Hinton, Vinyals, and Dean 2015), contrastive learning (Yang et al. 2022; Cui et al. 2021; Zhu et al. 2022), model ensemble (Wang et al. 2020; Cai, Wang, and Hwang 2021; Zhang et al. 2021b) because our method does not require additional data and model resources.

Our main contributions are summarized as follows:

(1) We propose a class-wise and sample-wise distribution to guide network training. Our method can greatly improve performance on imbalanced datasets.

(2) We introduce a new loss function that does not require additional data or model resources. Tail-class accuracy is greatly improved by this simple and effective method.

(3) A target distribution is constructed with two heuristic coefficients, and experiments indicate that both coefficients improve model performance.

(4) The two-stage learning strategy in this work can be combined with other representation learning methods for greater accuracy.

Related Work

Long-tailed identification: Long-tailed recognition is a common problem in computer vision and pattern recognition. Because of the large number of samples in head-class, they dominate the training of the network. The most direct solution is oversampling (Joloudari et al. 2023; Han, Wang, and Mao 2005; Feng, Zhong, and Huang 2021) and downsampling (Drummond, Holte et al. 2003; Estabrooks, Jo, and Japkowicz 2004). Oversampling is often accompanied by the problem of overfitting and has higher requirements on the method of oversampling. Undersampling is associated with the problem of information loss. In addition, some researchers use the method of contrastive learning, and others use the method of knowledge distillation.

Logits adjustment: By adjusting the logits (Alshammari et al. 2022; Zhao et al. 2022; Kang et al. 2019; Cao et al. 2019; Menon et al. 2020; Ren et al. 2020; Wang et al. 2021; Alexandridis et al. 2022) of the network, the tail-class score can be adaptively improved. They compensate for the logits of tail-class based on Bayesian theory. Specifically, they use $p(i)$ to compensate for logits, where $p(i)$ represents the probability of the i-th class appearing in the dataset. This method can effectively improve the accuracy of the tail-class, but it is often accompanied by a decrease in the accuracy of the head-class. Our method can maintain the accuracy of the head-class as much as possible while improving the accuracy of the tail-class. In this paper, we increase the weight norms in the classifier to the tail-class, which can also play a role in compensating for the logits of the tail-class. Our method indirectly adjusts logits by adjusting the parameters of the classifier, which are more in-depth than them. The experimental results also show that the effect of adjusting classifier parameters directly is better than that of compensating by Bayes' theorem. The latest inverse image frequency (IIF) (Alexandridis et al. 2022) is a multiplicative margin adjustment transformation of the logits in the classification layer of CNN.

Difficult sample mining: Difficult sample mining is an important tool to long-tailed recognition (Zhao et al. 2022; Gal and Ghahramani 2016; Peng, Islam, and Tu 2022; Agarwal, D'souza, and Hooker 2022). In the head-class, there may be samples that are difficult to classify, and in the tail-class, there may also be samples that are easy to classify (Zhao et al. 2022). Therefore, more detailed adjustments can be made to the model training from the perspective of samples. In previous studies, they used cosine similarity (Zhao et al. 2022; Wang and Yan 2022), gradient (Agarwal, D'souza, and Hooker 2022), uncertainty (Gal and Ghahramani 2016; Peng, Islam, and Tu 2022), margin gap (Lin and Bradic 2021) to measure the difficulty of sample learning. For simplicity, this paper uses cosine similarity to measure the learning difficulty of samples. Unlike previous work (Zhao et al. 2022), this paper not only considers the cosine similarity between the sample and its corresponding category center, but also considers the similarity between the sample and other category centers. We believe that information from other categories should not be discarded, which has been proven useful by experiments.

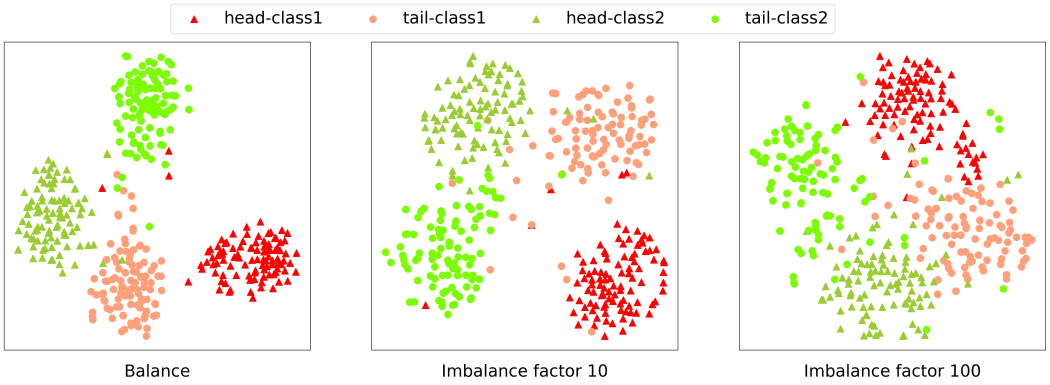


Figure 2: t-SNE (Laurens and Hinton 2008) feature visualization on CIFAR100-LT dataset equipped with ResNet32. Imbalance in the encoder was overlooked in other literature. It can be seen that as the degree of imbalance increases, the degree of imbalance in the encoder gradually intensifies, mainly reflected in the more dispersed tail-class features and almost unchanged aggregation of head-class features.

Proposed Method

Preliminary and Motivation

Deep long-tail learning faces a serious imbalance problem in the encoder and classifier. We aim to obtain a balanced network $f(x, \Theta)$ with a collection of parameters $\Theta = \theta_{ij}$, where i represents the i-th layer of the network and j represents the j-th filter. Let the real label of the input data be y, and the output label of the network be $\hat{y}$, $\hat{y} = f(x, \Theta)$. The measurement prediction error through the loss function $l(y, \hat{y})$, where l is a common classification loss such as the cross-entropy loss and the class-balance loss (Cui et al. 2019).

In this paper, for each sample x, we assume that the encoder output features $\hat{x}$ belong to the latent space L, and the corresponding category of each sample is y_i belonging to $\mathcal{C} = \{1, 2, ..., C\}$. $|W_i|$ is the i-th vector norm in the fully connected layer, where $i \in \mathcal{C}$. For category y_i, we define its decision boundary as,

$$W_i\hat{x} + b_i = W_j\hat{x} + b_j,$$

that is,

$$(W_i - W_j)\hat{x} + (b_i - b_j) = 0, \tag{1}$$

where $i, j \in \mathcal{C}$, $i \neq j$, and $\hat{x}$ represent the sample feature output by the encoder, b_i represents the bias corresponding to class y_i in the classifier. When the feature of the network output falls on the decision boundary. The network cannot distinguish whether the sample belongs to y_i or y_j. When the left side of Eq.1 is greater than 0, we believe that the sample belongs to the category y_i.

Assuming that y_i belongs to the head-class, y_j belongs to the tail-class, several studies (Menon et al. 2020; Kang et al. 2019) have found that $|W_i|$ is often greater than $|W_j|$, as shown in Figure 1(naive classifier). However, they did not explain why this would affect the accuracy of tail-class recognition. From Eq. (1), it can be seen that $|W_i| > |W_j|$ means the decision space of class y_i' s is larger than that of y_j's. Due to the small size of the tail-class decision space and the scattered tail-class features, more tail-class features will fall into the head-class decision space, leading to misclassification. Therefore, we decide to adjust the weight-norms distribution of the classifier. Unlike other methods (Kang et al. 2019; Alshammari et al. 2022), we consider the imbalance issue in the encoder, as shown in Figure 2. Specifically, we compensate for the imbalance in the encoder by inversing the weight-norms distribution of the classifier.

Weight-Norms Distribution Construction to Inversely Balance the Classifier

In this section, we show how to inverse the weight-norms distribution in the classifier. The weight-norms distribution of the classifier is made to be close to the class-wise and sample-wise distribution P. The distribution P is constructed as follows:

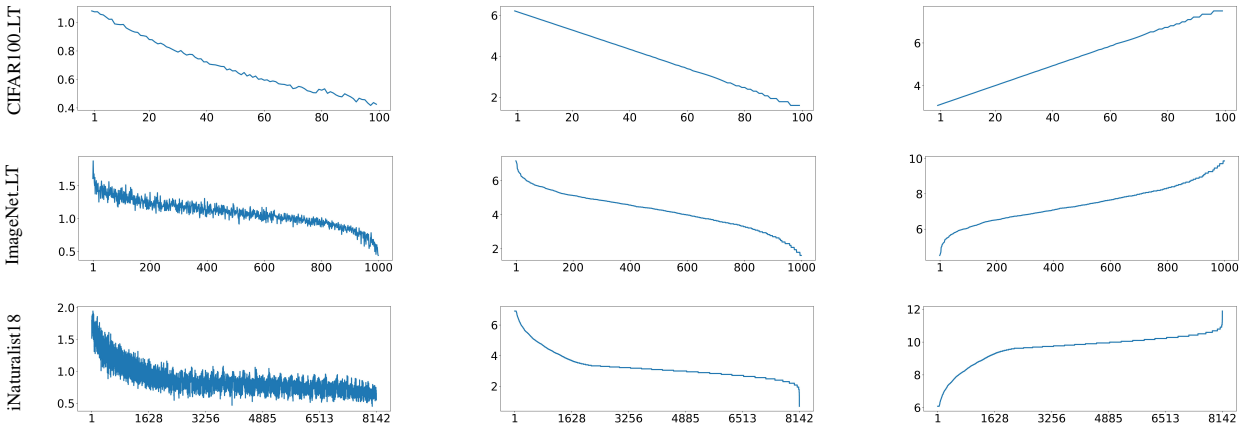
$$P = NC * HC. \tag{2}$$

where the two terms (number coefficient NC and hardness coefficient HC) will be defined next.

Number coefficient NC: Intuitively, NC is prior knowledge about the data distribution. Let $N = \sum_i N_i$ represent the total number of samples and N_i represent the number of samples in class i. In the long-tailed problem, the number of samples is the most important information, so how to use this information is particularly important. NC is related to the number of samples N_i of each class in the dataset. It should be noted that we find that the weight-norms distribution of the classifier is consistent with the logarithmic distribution of the number of samples, as shown in Figure 3. Therefore, we obtain the inverse weight-norms distribution through $-log_2(N_i + 1)$. To ensure that $NC > 0$, we added a constant term $log_2 N$. Finally, NC can be calculated as follows:

$$NC = log_2(\frac{N}{N_i + 1}). \tag{3}$$

Similarly to other logits adjustment methods (Zhao et al. 2022; Alexandridis et al. 2022), NC is negatively correlated with the number of samples. The difference is that our NC is inverted, as shown in Figure 3.



(a) Naive (b) Per-class Images (c) NC

Figure 3: Each row (from top to bottom) represents the respective dataset CIFAR100-LT, ImageNet-LT and iNaturalist18, each column (from left to right) represents the weight norms distribution of classifier, the logarithmic distribution of the number of images and the distribution of NC. It can be seen that the weight-norms distribution of the naive classifier is highly consistent with the distribution of the number of samples in each class. The distribution of NC is opposite to the other two distributions.

Hardness coefficient HC: In the head-class, there may be samples that are difficult to classify, and in the tail-class, there may also be samples that are easy to classify (Zhao et al. 2022). However, NC only has relations with the number of samples in each category, which does not contain information about the difficulty of classifying a sample. To properly account for the difficulty with classification, we introduce the other parameter HC - a parameter to measure the learning difficulty of the sample. It is related to not only the category of the sample but also the sample itself. The cosine similarity (Zhao et al. 2022; Wang and Yan 2022) between the features of the sample and the center of the category is used to measure the difficulty of a sample:

$$g(x_i) = \frac{W_j \hat{x_i}}{|W_j||\hat{x_i}|}, \tag{4}$$

where W_j is the parameter vector corresponding to j-th class in the fully connected layer and $\hat{x_i}$ is the feature of sample x_i. For each sample, $g(x_i)$ is a scalar. Our approach is different from this. We believe that information from other categories should not be discarded during the calculation of HC. In this way, we can obtain the relative difficulty between different categories. HC with category information can better guide the classifier to learn, that is,

$$g(x_i) = \frac{W \hat{x_i}}{|W||\hat{x_i}|}, \tag{5}$$

where W is the parameter matrix in the fully connected layer.

For each sample, $g(x_i)$ is a vector of length C, of the same form as logits. Experimental results have shown that this method is better than using the scalar cosine similarity directly. Similarly, HC should be negatively correlated with cosine similarity, and the higher the similarity, the lower the value of HC. To avoid negative HC values, we map the interval of $[-1, 1]$ to $[0, 1]$ by

$$HC = \frac{1 - g(x_i)}{2}. \tag{6}$$

HC can mine difficult samples in the head-class and simple samples in the tail-class. The general trend is still that HC in the tail-class is larger, which is consistent with our intuition. However, this does not mean that it can be replaced by NC because it is a finer and more adaptive parameter than NC. At the same time, to smooth out the distribution P, we introduce the hyperparameter $\lambda \in (0, 1)$, and rewrite Eq. (2) into

$$P = \lambda * NC * HC. \tag{7}$$

Experiments show that the hyperparameter λ is necessary to moderate the parameter distribution of the classifier and reduce the absolute difference between $|W_{head}|$ and $|W_{tail}|$, as shown in Figure 5.

KL divergence constraint: To make the distribution $|W|$ close to the distribution P, we used the KL-divergence constraint training procedure as shown below.

$$\begin{aligned} D_{KL}(|W|, P) = softmax(|W|) \\ [(log(softmax(|W|)) - log(softmax(P))]. \end{aligned} \tag{8}$$

It is important to note that the KL-divergence measures a probability distribution, so softmax operations for $|W|$ and P are required in the above equation. Furthermore, we found that using MSE-loss to constrain $|W|$ did not work well for the following two reasons: 1) MSE-loss constraints of the network are too strict (Kang et al. 2019). It is not conducive to network learning because we do not require $|W|$ and P to be equal in value. Instead, it is better to let the network learn the numerical size by itself. 2) MSE-loss is often used in regression problems, which is not suitable to have two distributions close together. As shown in Table 4, KL-divergence is better than MSE-loss.

Training Pipeline

In end-to-end training (Cui et al. 2019; Zhang et al. 2017; Zhao et al. 2022; Menon et al. 2020; Cao et al. 2019), some methods are used to improve the accuracy of the tail-class, but the representation of the head-class feature is damaged. To achieve an improved trade-off between the head-class and tail-class, we adopted a two-stage training approach (Zhong et al. 2021; Alexandridis et al. 2022; Alshammari et al. 2022; Kang et al. 2019) to separate the feature learning stage from the accuracy improvement stage of the tail-class. The loss function $l(y, \widetilde{y})$ is used to predict the error between the real value and the predicted value, and l is the classification loss such as the cross-entropy (CE) loss, class-balanced (CB) loss (Cui et al. 2019). We used the CE-loss training network in the first stage, as shown below.

$$\Theta^* = \arg\min_{\Theta} F(\Theta; D) = \sum_{i=1}^{N} l(f(x_i; \Theta), y_i), \quad (9)$$

where Θ indicates the network parameter and D denotes the entire dataset.

There is a serious category imbalance in the classifier and encoder obtained in the first stage, as shown in Figure 1 and Figure 2. Therefore, in the second stage, the encoder is fixed, CB-loss (Cui et al. 2019) and KL-divergence are used to fine-tune the classifier. The purpose of using the KL-divergence is to get $|W|$ close to the inverse distribution P. The final loss functions are as follows:

$$\begin{aligned} \Theta^* &= \arg\min_{\Theta} F(\Theta; D) \\ &= \sum_{i=1}^{N} (l(f(x_i; \Theta), y_i) + D_{KL}(|W|, P)). \end{aligned} \quad (10)$$

Relationship to Existing Works

The idea of weight balancing for long-tailed recognition is not new. For example, the τ-normalized classifier (Kang et al. 2019) attempts to eliminate the imbalance of the classifier by adjusting the weight-norms of the classifier through a so-called τ-normalization procedure. More recently, weight balancing strategies, consisting of L2-normalization, weight decay, and MaxNorm, were developed in (Alshammari et al. 2022). The difference between our approach and those existing works is that we consider the imbalance phenomenon in the encoded feature. We compensate for the imbalance in the encoder by "overbalancing" the classifier. Conventional wisdom such as τ-normalized (Kang et al. 2019) and WD+MaxNorm(Alshammari et al. 2022) believes that the weight-norms distribution of the classifier should be balanced. As shown in the curve highlighted in yellow and green in Figure 4, the distribution of the weight-norms obtained by their method is almost balanced. The change in weight-norms obtained by WD+MaxNorm(Alshammari et al. 2022) is small, which means that the decision space is nearly uniform. This is equivalent to setting λ small in our method.

Both MaxNorm (Alshammari et al. 2022) and WD+MaxNorm(Alshammari et al. 2022) did not consider the imbalance in the encoder. We challenge their view by advocating higher priority for the tail class. Specifically, we believe that the imbalance in the encoder is consistent with the classifier due to end-to-end training, so we compensate for the imbalance in the pre-trained encoder by *inverting* the weight-norms distribution of the classifier. We hope to combine an imbalanced encoder with an "overbalanced" classifier to create a more balanced model. As shown in Fig. 4, the result of compounding the two distributions highlighted in red and blue leads to a balanced distribution.

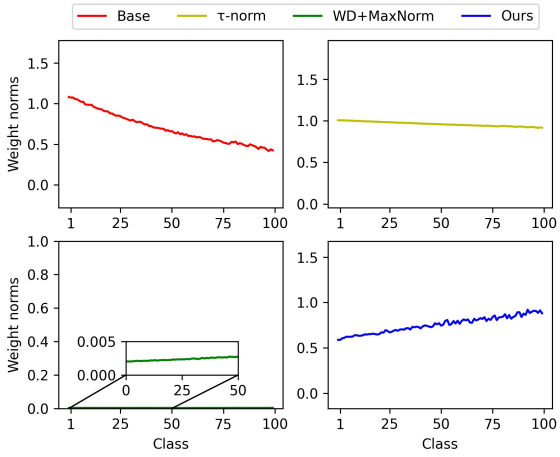


Figure 4: Comparison of different weight norms distribution. The x-axis represents the number of class and the y-axis represents the value of weight-norms.

It is also interesting to contrast our approach with the distribution alignment method (Zhang et al. 2021a). The key idea behind distribution alignment is to introduce a calibration function that facilitates the adjustment of classification scores for each data point. Similarly to ours, distribution alignment (Zhang et al. 2021a) also involves a two-stage learning to balance the class prior by generalized re-weighting. Unlike ours, the two-stage imbalance learning in (Zhang et al. 2021a) adopts a balanced distribution as a reference for calibration. Note that our method is not a derivative of the re-weighting approach such as (Cui et al. 2019) but a combination of imbalanced encoder and overbalanced classifier during the second-stage training. The outstanding performance of this work is due to the inverse distribution which simultaneously considers both the imbalance and the difficulty of the different categories and samples (i.e., re-weighting + re-margining).

Experimental Results

We conducted a number of experiments to demonstrate the effectiveness of our method. The model was first evaluated on a variety of popular long-tailed datasets. In addition, some key parameters of the ablation experiment are proving to be necessary. Finally, we compare the proposed method with other competing methods.

Experimental Setup

Datasets: We have carried out a series of experiments in CIFAR100-LT (Krizhevsky, Hinton et al. 2009), ImageNet-LT (Liu et al. 2018), iNaturalist2018 (Van Horn et al. 2018). CIFAR100-LT was obtained from CIFAR100 by exponential decay downsampling. It contains 10.8k training images in 100 categories. ImageNet-LT contains 115.8K training images, a total of 1000 categories. The sample number for each category ranges from 5 to 1280, and the imbalance factor is 256. It is a subset of the ImageNet dataset. iNaturalist 2018 is a large-scale real dataset that contains 437.5k training images with a total of 8142 categories. The category with the largest sample size contains 2101 images, while the category with the smallest sample size has only one image.

Implementation details: CIFAR100-LT dataset requires only one GeForce RTX 2080 card, and the other two datasets require 8 GeForce RTX 2080 cards due to batch size and image size. According to previous studies (Cui et al. 2019; Jamal et al. 2020; Kang et al. 2019; Liu et al. 2019; Yang and Xu 2020; Alshammari et al. 2022), the baseline networks used ResNet32 (He et al. 2016)(for CIFAR100-LT), ResNeXt50 (Xie et al. 2017)(for ImageNet-LT), ResNet50 (He et al. 2016) (for iNaturalist 2018). First, in the first stage, for CIFAR100LT, the batch size is 64, the learning rate is 0.01, and the weight decay is 5e-3. For ImageNet-LT, the batch size is 128, the learning rate is 0.01, and the weight decay is 5e-4. For iNaturalist 2018, the batch size is 512, the learning rate is 0.02, and the weight decay is 1e-4. Each of the three datasets trains 200 epochs, and the learning rate uses the cosine decay to 0. Next is the second stage. For CIFAR100-LT, the batch size is 64, the learning rate is 0.005 and the hyperparameter λ=0.15. For ImageNet-LT, the batch size is 512, the learning rate is 0.01, and the hyperparameter λ=0.05. For iNaturalist 2018, the batch size is 512, the learning rate is 0.0002, and the hyperparameter λ=0.01. Each of the three datasets trains only 10 epochs in the second stage, and the learning rate uses the cosine decay to 0.

Evaluation protocol: All training is done on the long-tailed dataset, and the test or valid set is balanced. Consistent with other long-tail work (Alshammari et al. 2022; Ren et al. 2020), we divide the categories according to the number of images N_i, the head class: $N_i > 100$, the medium class $100 \geq N_i \geq 20$, and the tail class $N_i < 20$.

Comparison with Competing Methods

Our approach is essentially a logits adjustment approach, and some of the main comparison methods (Alshammari et al. 2022; Zhao et al. 2022; Kang et al. 2019; Cao et al. 2019; Menon et al. 2020; Ren et al. 2020; Wang et al. 2021; Alexandridis et al. 2022) in this paper are based on this idea. As well as other comparison methods such as knowledge distillation, contrast learning, model ensemble. Our method achieves the best results among logits adjustment methods. In the following, we present the experimental results for each dataset.

Experimental results on CIFAR100-LT dataset. As shown in Table 1, the top-1 accuracy of our method is 1.25% higher than the current best result WD+MaxNorm (Alshammari et al. 2022), and 5.7% higher than RIDE (Wang et al. 2020). Although RIDE is a model ensemble method, which requires much larger computational resources than ours, we can still achieve higher accuracy with a single model. To make a fair comparison with WD+MaxNorm, we also used CB-loss in stage 2. Their method decreased sharply without CB-loss, but our success does not depend on CB-loss. The effectiveness of our method can be seen from Table 1. As the imbalance factor increases, the gain gradually increases (1.25%(0.11%)higher than WD+MaxNorm+CB when imabalance factor is 100(50)). This is mainly because the imbalance problem in the encoder also increases, as described in Fig 2. This proves the effectiveness of the IWB from another perspective.

	Imbalance factor	100	50
naive	CE	38.38	43.85
rebalance loss	focal	38.41	44.32
	CB	39.60	45.32
logits adj.	LDAM-DRW	42.04	46.62
	LogitAjust	43.89	47.03
	τ-norm	47.73	52.53
	IIF	48.8	-
	BALMS	49.20	-
	MARC	52.96	-
	WD+MaxNorm	48.60	53.20
	WD+MaxNorm+CB	53.55	57.71
	IWB(ours)	53.3	56.3
	IWB+CB(ours)	**54.8**	**57.82**
knowledge dis.	KD	40.36	45.49
	SSD	46.00	50.50
contrastive lea.	Paco	52.00	56.00
	BCL	51.93	56.59
model ensemble	ACE	49.6	51.90
	RIDE(4 experts)	49.1	-

Table 1: Top-1 accuracy on the testset of CIFAR100-LT. Imbalance factor is 50 and 100.

	ImageNet-LT				iNaturalist2018			
Method	Many	Med.	Few	All	Many	Med.	Few	All
CE	65.9	37.5	7.7	44.4	72.2	63.0	57.2	61.7
LogitAjust	-	-	-	51.1	-	-	-	69.9
τ-norm	59.1	46.9	30.7	49.4	65.6	65.3	65.5	65.6
cRT	61.8	46.2	27.3	49.6	69.0	66.0	63.2	65.2
BALMS	50.3	39.5	25.3	41.8	-	-	-	-
MARC	60.4	50.3	36.6	52.3	-	-	-	70.4
ALA	64.1	49.9	34.7	53.3	71.3	**70.8**	70.4	70.7
WD+MaxN.	62.5	50.4	**41.5**	53.9	71.2	70.4	69.7	70.2
IWB (ours)	**64.2**	**52.2**	40.2	**55.2**	**72.3**	70.6	**72.5**	**71.5**

Table 2: Top-1 accuracy on the testset of ImageNet-LT and iNaturalist2018.

Experimental results on ImageNet-LT and iNaturalist2018 dataset. Our method achieves the best results among the published methods of logits adjustment. As shown in Table 2, we outperformed (Alshammari et al. 2022)

by 1.3%, much more than other logits adjustment methods. Unlike other logits adjustment methods, these methods sacrifice the head-class accuracy to improve the tail-class accuracy. Our method can ensure the preservation of head-class accuracy; meanwhile, improve the accuracy of the tail-class and medium-class, as shown in Table 2. Similar to the results on ImageNet-LT, we have achieved best results than all existing methods of logits adjustment on iNaturalist2018 dataset. As shown in Table 2, the top-1 accuracy of our method is 1.3% higher than (Alshammari et al. 2022). Again, we have a better trade-off between head-class and tail-class.

Ablation Study

Effect of λ on experimental results. The quality of target distribution P directly affects the distribution in the latent space L. However, if $NC * HC$ is used as the distribution P directly, the result is not satisfactory. This is mainly because a direct use of $NC * HC$ will cause a large difference between $|W_{head}|$ and $|W_{tail}|$. Accordingly, we multiply it by λ ($\lambda \in (0, 1)$), to smooth the distribution P and reduce the gap between $|W_{head}|$ and $|W_{tail}|$. This is conceptually similar to the effect of annealing in knowledge distillation. Parameter smoothing is also adopted by other works (Kang et al. 2019) without substantial justification.

Figure 5 shows the effects of different λ on CIFAR100-LT dataset. As shown in Figure 5, the model performs poorly when λ is not used ($\lambda = 1$). As λ decreases, the effect of the model gradually increases, but λ can not be too small. First, if λ is too small, the total accuracy of the model will be reduced. For example, when λ=0.05, the total accuracy of the model will be 54.4%; when λ is 0.15, the total accuracy of the model will be 54.8%. Second, as shown in Figure 5, when λ decreases, the model's effect on the tail-class will gradually decline, while its effect on the head-class will increase. In contrast, as λ increases from 0.05 to 0.2, the accuracy of the tail-class has improved from 33. 5% to 36. 1%. Based on our previous analysis, this is because the absolute gap between $|W_{tail}|$ and $|W_{head}|$ increases, the compensation for imbalance in encoder is gradually increasing. Furthermore, during the experiment, we found that the larger the dataset, the smaller the optimal value of λ. In our opinion, this is due to the fact that the more categories there are, the more detailed the division of latent space, and the model is more sensitive to changes in λ.

The influence of NC and HC. Studies on NC and HC were used to demonstrate their effectiveness. We remove NC and HC successively to measure the performance of the model. To eliminate the influence of λ on the experimental results, we adjust λ to the optimal value for each experiment. As shown in Table 3, we observe that when NC acts alone, the accuracy is 7.2% higher than that of naive and 0.2% lower than that of the combined action. When NC is combined with HC, the accuracy can reach the highest of 54.8%. Furthermore, the combined action can greatly improve the precision of the tail-class as shown in the penultimate line. In the last row of Table 3, we show the result of using scalar cosine similarity to measure the difficulty of the sample. Although the total accuracy is not affected,

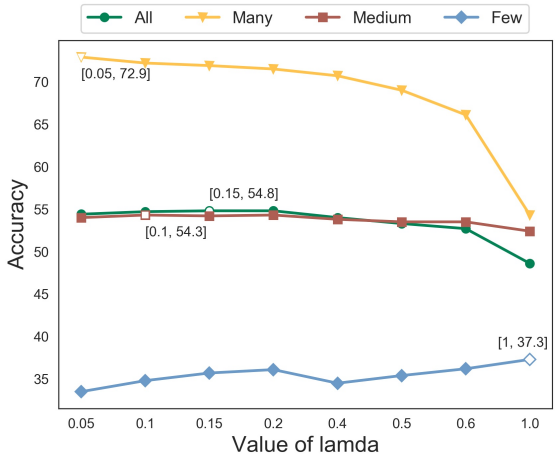


Figure 5: Influence of λ on CIFAR100-LT, imbalance factor is 100. The x-axis represents the value of λ and the y-axis represents top-1 accuracy. Notice that the x-axis is not uniform.

NC	HC	Many	Medium	Few	All
×	×	**77.5**	46.5	13.3	47.4
✓	×	72.3	54.3	34.5	54.6
✓	✓(vector)	71.9	54.2	**35.7**	**54.8**
✓	✓(scalar)	72.4	**55.5**	33.3	**54.8**

Table 3: Ablation studies of NC and HC on CIFAR100-LT. The imbalance factor is 100.

the recognition accuracy of the tail-class decreases sharply, 2.4% lower than the best result.

Method	Many	Medium	Few	All
MSE	**72.7**	**54.3**	33.4	54.4
KL	71.9	54.2	**35.7**	**54.8**

Table 4: Ablation studies of MSE Loss and KL Loss on CIFAR100-LT. The imbalance factor is 100.

Conclusion

This paper uses a two-stage approach to solve the long-tailed issue. In the second stage, we introduced a sample-wise and class-wise distribution to make the weight-norms distribution of the classifier reversal, which improves the size of the tail decision space in the latent space and compensates for imbalance in the encoder. In case of ensuring the accuracy of head-class, the recognition accuracy of tail-class is greatly improved. Our method is essentially a logits adjustment method with inverse weight-balancing. According to our experimental results, our method makes the best scores in logits adjustment methods. Last but not least, our approach is simple to implement and requires no additional data or model resources.

Acknowledgments

This work was supported in part by the Natural Science Foundation of China under Grant 61991451 and Grant 61836008.

References

Agarwal, C.; D'souza, D.; and Hooker, S. 2022. Estimating example difficulty using variance of gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10368–10378.

Alexandridis, K. P.; Luo, S.; Nguyen, A.; Deng, J.; and Zafeiriou, S. 2022. Inverse Image Frequency for Long-tailed Image Recognition. *arXiv preprint arXiv:2209.04861*.

Alshammari, S.; Wang, Y.-X.; Ramanan, D.; and Kong, S. 2022. Long-tailed recognition via weight balancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6897–6907.

Cai, J.; Wang, Y.; and Hwang, J.-N. 2021. Ace: Ally complementary experts for solving long-tailed recognition in one-shot. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 112–121.

Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; and Ma, T. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32.

Cui, J.; Zhong, Z.; Liu, S.; Yu, B.; and Jia, J. 2021. Parametric contrastive learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, 715–724.

Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9268–9277.

Drummond, C.; Holte, R. C.; et al. 2003. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11, 1–8.

Estabrooks, A.; Jo, T.; and Japkowicz, N. 2004. A multiple resampling method for learning from imbalanced data sets. *Computational intelligence*, 20(1): 18–36.

Feng, C.; Zhong, Y.; and Huang, W. 2021. Exploring classification equilibrium in long-tailed object detection. In *Proceedings of the IEEE/CVF International conference on computer vision*, 3417–3426.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059. PMLR.

Han, H.; Wang, W.-Y.; and Mao, B.-H. 2005. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I 1*, 878–887. Springer.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Iscen, A.; Araujo, A.; Gong, B.; and Schmid, C. 2021. Class-balanced distillation for long-tailed visual recognition. *arXiv preprint arXiv:2104.05279*.

Jamal, M. A.; Brown, M.; Yang, M.-H.; Wang, L.; and Gong, B. 2020. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7610–7619.

Joloudari, J. H.; Marefat, A.; Nematollahi, M. A.; Oyelere, S. S.; and Hussain, S. 2023. Effective Class-Imbalance Learning Based on SMOTE and Convolutional Neural Networks. *Applied Sciences*, 13(6): 4006.

Kang, B.; Xie, S.; Rohrbach, M.; Yan, Z.; Gordo, A.; Feng, J.; and Kalantidis, Y. 2019. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90.

Laurens, V. D. M.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(2605): 2579–2605.

Li, J.; Tan, Z.; Wan, J.; Lei, Z.; and Guo, G. 2022. Nested collaborative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6949–6958.

Li, T.; Wang, L.; and Wu, G. 2021. Self supervision to distillation for long-tailed visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, 630–639.

Lin, J. Z.; and Bradic, J. 2021. Learning to combat noisy labels via classification margins. *arXiv preprint arXiv:2102.00751*.

Liu, S.; Garrepalli, R.; Dietterich, T.; Fern, A.; and Hendrycks, D. 2018. Open category detection with PAC guarantees. In *International Conference on Machine Learning*, 3169–3178. PMLR.

Liu, Z.; Miao, Z.; Zhan, X.; Wang, J.; Gong, B.; and Yu, S. X. 2019. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2537–2546.

Menon, A. K.; Jayasumana, S.; Rawat, A. S.; Jain, H.; Veit, A.; and Kumar, S. 2020. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*.

Peng, B.; Islam, M.; and Tu, M. 2022. Angular Gap: Reducing the Uncertainty of Image Difficulty through Model Calibration. In *Proceedings of the 30th ACM International Conference on Multimedia*, 979–987.

Ren, J.; Yu, C.; Ma, X.; Zhao, H.; Yi, S.; et al. 2020. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33: 4175–4186.

Van Horn, G.; Mac Aodha, O.; Song, Y.; Cui, Y.; Sun, C.; Shepard, A.; Adam, H.; Perona, P.; and Belongie, S. 2018. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8769–8778.

Wang, H.; and Yan, J. 2022. Leveraging Angular Information Between Feature and Classifier for Long-tailed Learning: A Prediction Reformulation Approach. *arXiv preprint arXiv:2212.01565*.

Wang, X.; Lian, L.; Miao, Z.; Liu, Z.; and Yu, S. X. 2020. Long-tailed recognition by routing diverse distribution-aware experts. *arXiv preprint arXiv:2010.01809*.

Wang, Y.; Zhang, B.; Hou, W.; Wu, Z.; Wang, J.; and Shinozaki, T. 2021. Margin calibration for long-tailed visual recognition. *arXiv preprint arXiv:2112.07225*.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1492–1500.

Yang, Y.; and Xu, Z. 2020. Rethinking the value of labels for improving class-imbalanced learning. *Advances in neural information processing systems*, 33: 19290–19301.

Yang, Z.; Pan, J.; Yang, Y.; Shi, X.; Zhou, H.-Y.; Zhang, Z.; and Bian, C. 2022. ProCo: Prototype-Aware Contrastive Learning for Long-Tailed Medical Image Classification. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VIII*, 173–182. Springer.

Zhang, S.; Li, Z.; Yan, S.; He, X.; and Sun, J. 2021a. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2361–2370.

Zhang, X.; Fang, Z.; Wen, Y.; Li, Z.; and Qiao, Y. 2017. Range loss for deep face recognition with long-tailed training data. In *Proceedings of the IEEE International Conference on Computer Vision*, 5409–5418.

Zhang, Y.; Hooi, B.; Hong, L.; and Feng, J. 2021b. Test-agnostic long-tailed recognition by test-time aggregating diverse experts with self-supervision. *arXiv e-prints*, arXiv–2107.

Zhang, Y.; Kang, B.; Hooi, B.; Yan, S.; and Feng, J. 2021c. Deep long-tailed learning: A survey. *arXiv preprint arXiv:2110.04596*.

Zhao, Y.; Chen, W.; Tan, X.; Huang, K.; and Zhu, J. 2022. Adaptive logit adjustment loss for long-tailed visual recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 3472–3480.

Zhong, Z.; Cui, J.; Liu, S.; and Jia, J. 2021. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 16489–16498.

Zhu, J.; Wang, Z.; Chen, J.; Chen, Y.-P. P.; and Jiang, Y.-G. 2022. Balanced contrastive learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6908–6917.