BBN: Bilateral-Branch Network with Cumulative Learning for Long-Tailed Visual Recognition

Boyan Zhou[1] Quan Cui[1,2] Xiu-Shen Wei[1*] Zhao-Min Chen[1,3]
[1]Megvii Technology [2]Waseda University [3]Nanjing University

Abstract

*Our work focuses on tackling the challenging but natural visual recognition task of long-tailed data distribution (*i.e.*, a few classes occupy most of the data, while most classes have rarely few samples). In the literature, class re-balancing strategies (*e.g.*, re-weighting and re-sampling) are the prominent and effective methods proposed to alleviate the extreme imbalance for dealing with long-tailed problems. In this paper, we firstly discover that these re-balancing methods achieving satisfactory recognition accuracy owe to that they could significantly promote the classifier learning of deep networks. However, at the same time, they will unexpectedly damage the representative ability of the learned deep features to some extent. Therefore, we propose a unified Bilateral-Branch Network (BBN) to take care of both representation learning and classifier learning simultaneously, where each branch does perform its own duty separately. In particular, our BBN model is further equipped with a novel cumulative learning strategy, which is designed to first learn the universal patterns and then pay attention to the tail data gradually. Extensive experiments on four benchmark datasets, including the large-scale iNaturalist ones, justify that the proposed BBN can significantly outperform state-of-the-art methods. Furthermore, validation experiments can demonstrate both our preliminary discovery and effectiveness of tailored designs in BBN for long-tailed problems. Our method won the first place in the iNaturalist 2019 large scale species classification competition, and our code is open-source and available at* https://github.com/Megvii-Nanjing/BBN.

1. Introduction

With the advent of research on deep Convolutional Neural Networks (CNNs), the performance of image classification has witnessed incredible progress. The success is undoubtedly inseparable to available and high-quality large-scale datasets, *e.g.*, ImageNet ILSVRC 2012 [24], MS COCO [18] and Places Database [37], *etc*. In contrast with these visual recognition datasets exhibiting roughly uniform distributions of class labels, real-world datasets always have skewed distributions with *a long tail* [15, 26], *i.e.*, a few classes (a.k.a. *head class*) occupy most of the data, while most classes (a.k.a. *tail class*) have rarely few samples, cf. Figure 1. Moreover, more and more long-tailed datasets reflecting the realistic challenges are constructed and released by the computer vision community in very recent years, *e.g.*, iNaturalist [6], LVIS [10] and RPC [29]. When dealing with

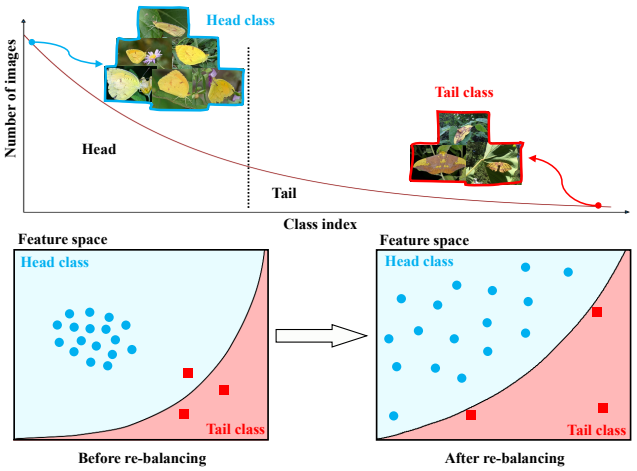

Figure 1. Real-world large-scale datasets often display the phenomenon of long-tailed distributions. The extreme imbalance causes tremendous challenges on the classification accuracy, especially for the tail classes. Class re-balancing strategies can yield better classification accuracy for long-tailed problems. In this paper, we reveal that the mechanism of these strategies is to significantly promote classifier learning but will unexpectedly damage the representative ability of the learned deep features to some extent. As conceptually demonstrated, after re-balancing, the decision boundary (*i.e.*, black solid arc) tends to accurately classify the tail data (*i.e.*, red squares). However, the intra-class distribution of each class becomes more separable. Quantitative results are presented in Figure 2, and more analyses can be found in the supplementary materials.

*X.-S. Wei is the corresponding author (weixs.gm@gmail.com). Q. Cui and Z.-M. Chen's contribution was made when they were interns in Megvii Research Nanjing, Megvii Technology, China. This research was supported by National Key R&D Program of China (No. 2017YFA0700800).

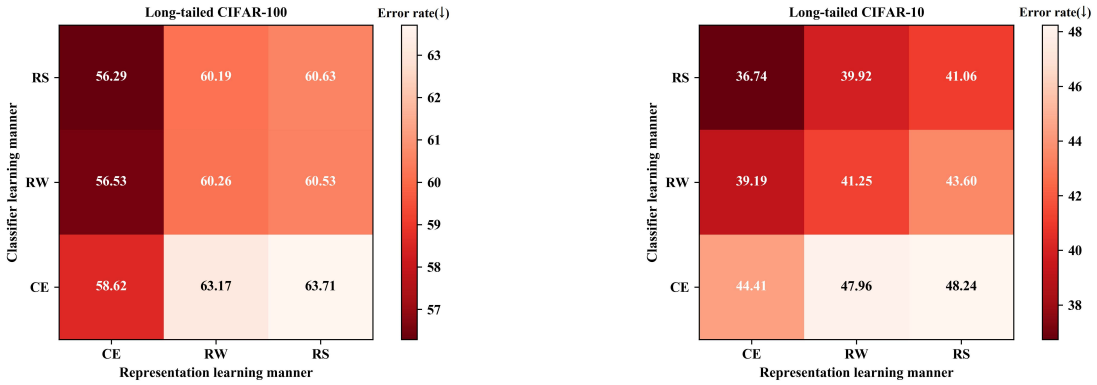

Figure 2. Top-1 error rates of different manners for representation learning and classifier learning on two long-tailed datasets CIFAR-100-IR50 and CIFAR-10-IR50 [3]. "CE" (Cross-Entropy), "RW" (Re-Weighting) and "RS" (Re-Sampling) are the conducted learning manners. As observed, when fixing the representation (comparing error rates of three blocks in the vertical direction), the error rates of classifiers trained with RW/RS are reasonably lower than CE. While, when fixing the classifier (comparing error rates in the horizontal direction), the representations trained with CE surprisingly get lower error rates than those with RW/RS. Experimental details can be found in Section 3.

such visual data, deep learning methods are not feasible to achieve outstanding recognition accuracy due to both the data-hungry limitation of deep models and also the extreme class imbalance trouble of long-tailed data distributions.

In the literature, the prominent and effective methods for handling long-tailed problems are class re-balancing strategies, which are proposed to alleviate the extreme imbalance of the training data. Generally, class re-balancing methods are roughly categorized into two groups, *i.e.*, re-sampling [25, 1, 14, 1, 11, 2, 7, 21, 4] and cost-sensitive re-weighting [13, 28, 5, 23]. These methods can adjust the network training, by re-sampling the examples or re-weighting the losses of examples within mini-batches, which is in expectation closer to the test distributions. Thus, class re-balancing is effective to directly influence the classifier weights' updating of deep networks, *i.e.*, promoting the classifier learning. That is the reason why re-balancing could achieve satisfactory recognition accuracy on long-tailed data.

However, although re-balancing methods have good eventual predictions, we argue that these methods still have adverse effects, *i.e.*, they will also unexpectedly damage the representative ability of the learned deep features (*i.e.*, the representation learning) to some extent. In concretely, re-sampling has the risks of over-fitting the tail data (by over-sampling) and also the risk of under-fitting the whole data distribution (by under-sampling), when data imbalance is extreme. For re-weighting, it will distort the original distributions by directly changing or even inverting the data presenting frequency.

As a preliminary of our work, by conducting validation experiments, we justify our aforementioned argumentations. Specifically, to figure out how re-balancing strategies work, we divide the training process of deep networks into two stages, *i.e.*, to separately conduct the representation learning and the classifier learning. At the former stage for representation learning, we employ plain training (conventional cross-entropy), re-weighting and re-sampling as three learning manners to obtain their corresponding learned representations. Then, at the latter stage for classifier learning, we first fix the parameters of representation learning (*i.e.*, backbone layers) converged at the former stage and then retrain the classifiers of these networks (*i.e.*, fully-connected layers) *from scratch*, also with the three aforementioned learning manners. In Figure 2, the prediction error rates on two benchmark long-tailed datasets [3], *i.e.*, CIFAR-100-IR50 and CIFAR-10-IR50, are reported. Obviously, when fixing the representation learning manner, re-balancing methods reasonably achieve lower error rates, indicating they can promote classifier learning. On the other side, by fixing the classifier learning manner, plain training on original imbalanced data can bring better results according to its better features. Also, the worse results of re-balancing methods prove that they will hurt feature learning.

Therefore, in this paper, for exhaustively improving the recognition performance of long-tailed problems, we propose a unified Bilateral-Branch Network (BBN) model to take care of *both representation learning and classifier learning* simultaneously. As shown in Figure 3, our BBN model consists of two branches, termed as the "conventional learning branch" and the "re-balancing branch". In general, each branch of BBN separately performs its own duty for representation learning and classifier learning, respectively. As the name suggests, the conventional learning branch equipped with the typical uniform sampler w.r.t. the original data distribution is responsible for learning universal patterns for recognition. While, the re-balancing branch coupled with a reversed sampler is designed to model the tail data. After that, the predicted outputs of these bilateral branches are aggregated in the cumulative learning part by an adaptive trade-off parameter α. α is automatically generated by the

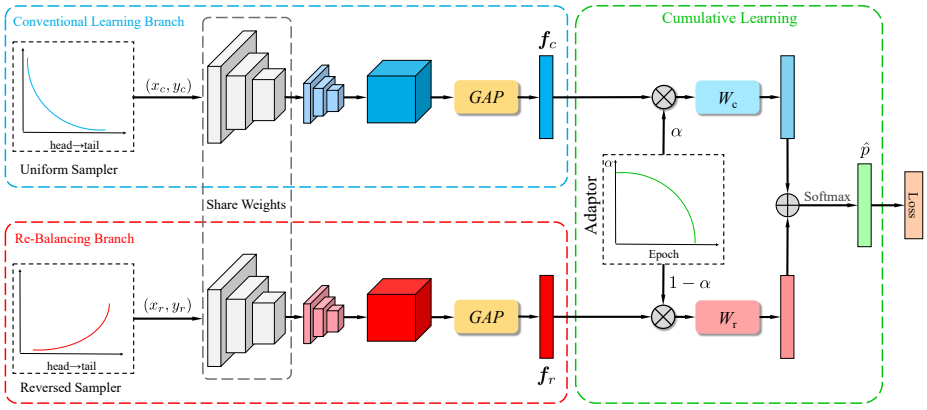

Figure 3. Framework of our Bilateral-Branch Network (BBN). It consists of three key components: 1) The *conventional learning branch* takes input data from a uniform sampler, which is responsible for learning universal patterns of original distributions. While, 2) the *re-balancing branch* takes inputs from a reversed sampler and is designed for modeling the tail data. The output feature vectors $\boldsymbol{f}_c$ and $\boldsymbol{f}_r$ of two branches are aggregated by 3) our *cumulative learning strategy* for computing training losses. "*GAP*" is short for global average pooling.

"Adaptor" according to the number of training epochs, which adjusts the whole BBN model to firstly learn the universal features from the original distribution and then pay attention to the tail data gradually. More importantly, α could further control the parameter updating of each branch, which, for example, avoids damaging the learned universal features when emphasizing the tail data at the later periods of training.

In experiments, empirical results on four benchmark long-tailed datasets show that our model obviously outperforms existing state-of-the-art methods. Moreover, extensive validation experiments and ablation studies can prove the aforementioned preliminary discovery and also validate the effectiveness of our tailored designs for long-tailed problems.

The main contributions of this paper are as follows:

- We explore the mechanism of the prominent class re-balancing methods for long-tailed problems, and further discover that these methods can significantly promote classifier learning and meanwhile will affect the representation learning w.r.t. the original data distribution.
- We propose a unified Bilateral-Branch Network (BBN) model to take care of both representation learning and classifier learning for exhaustively boosting long-tailed recognition. Also, a novel cumulative learning strategy is developed for adjusting the bilateral learnings and coupled with our BBN model's training.
- We evaluate our model on four benchmark long-tailed visual recognition datasets, and our proposed model consistently achieves superior performance over previous competing approaches.

2. Related work

Class re-balancing strategies: *Re-sampling* methods as one of the most important class re-balancing strategies could be divided into two types: 1) Over-sampling by simply repeating data for minority classes [25, 1, 2] and 2) under-sampling by abandoning data for dominant classes [14, 1, 11]. But sometimes, with re-sampling, duplicated tailed samples might lead to over-fitting upon minority classes [4, 5], while discarding precious data will certainly impair the generalization ability of deep networks.

Re-weighting methods are another series of prominent class re-balancing strategies, which usually allocate large weights for training samples of tail classes in loss functions [13, 28]. However, re-weighting is not capable of handling the large-scale, real-world scenarios of long-tailed data and tends to cause optimization difficulty [20]. Consequently, Cui *et al.* [5] proposed to adopt the effective number of samples [5] instead of proportional frequency. Thereafter, Cao *et al.* [3] explored the margins of the training examples and designed a label-distribution-aware loss to encourage larger margins for minority classes.

In addition, recently, some two-stage fine-tuning strategies [3, 6, 22] were developed to modify re-balancing for effectively handling long-tailed problems. Specifically, they separated the training process into two single stages. In the first stage, they trained networks as usual on the original imbalanced data and only utilized re-balancing at the second stage to fine-tune the network with a small learning rate.

Beyond that, other methods of different learning paradigms were also proposed to deal with long-tailed problems, *e.g.*, metric learning [34, 13], meta-learning [19] and knowledge transfer learning [28, 36], which, however, are not within the scope of this paper.

Mixup: Mixup [33] was a general data augmentation algorithm, *i.e.*, convexly combining random pairs of training images and their associated labels, to generate *additional* samples when training deep networks. Also, manifold mixup [27] conducted mixup operations on random

pairs of samples in the manifold feature space for augmentation. The mixed ratios in mixup were sampled from the β-distribution to increase the randomness of augmentation. Although mixup is clearly far from our unified end-to-end trainable model, in experiments, we still compared with a series of mixup algorithms to validate our effectiveness.

3. How class re-balancing strategies work?

In this section, we attempt to figure out the working mechanism of these class re-balancing methods. More concretely, we divide a deep classification model into two essential parts: 1) the feature extractor (*i.e.*, frontal base/backbone networks) and 2) the classifier (*i.e.*, last fully-connected layers). Accordingly, the learning process of a deep classification network could be separated into representation learning and classifier learning. Since class re-balancing strategies could boost the classification accuracy by altering the training data distribution closer to test and paying more attention to the tail classes, we propose a conjecture that the way these strategies work is to promote classifier learning significantly but might damage the universal representative ability of the learned deep features due to distorting original distributions.

In order to justify our conjecture, we design a two-stage experimental fashion to separately learn representations and classifiers of deep models. Concretely, in the first stage, we train a classification network with plain training (*i.e.*, cross-entropy) or re-balancing methods (*i.e.*, re-weighting/re-sampling) as learning manners. Then, we obtain different kinds of feature extractors corresponding to these learning manners. When it comes to the second stage, we fix the parameters of the feature extractors learned in the former stage, and retrain classifiers *from scratch* with the aforementioned learning manners again. In principle, we design these experiments to fairly compare the quality of representations and classifiers learned by different manners by following the control variates method.

The CIFAR [16] datasets are a collection of images that are commonly used to assess computer vision approaches. Previous work [5, 3] created long-tailed versions of CIFAR datasets with different imbalance ratios, *i.e.*, the number of the most frequent class divided by the least frequent class, to evaluate the performance. In this section, following [3], we also use long-tailed CIFAR-10/CIFAR-100 as the test beds.

As shown in Figure 2, we conduct several contrast experiments to validate our conjecture on CIFAR-100-IR50 (long-tailed CIFAR-100 with imbalance ratio 50). As aforementioned, we separate the whole network into two parts: the feature extractor and classifier. Then, we apply three manners for the feature learning and the classifier learning respectively according to our two-stage training fashion. Thus, we can obtain nine groups of results based on different permutations: (1) Cross-Entropy (CE): We train the networks as usual on the original imbalanced data with the conventional cross-entropy loss. (2) Re-Sampling (RS): We first sample a class uniformly and then collect an example from that class by sampling with replacement. By repeating this process, a balanced mini-batch data is obtained. (3) Re-Weighting (RW): We re-weight all the samples by the inverse of the sample size of their classes. The error rate is evaluated on the validation set. As shown in Figure 2, we have the observations from two perspectives:

- **Classifiers:** When we apply the same representation learning manner (comparing error rates of three blocks in the vertical direction), it can be reasonably found that RW/RS always achieve lower classification error rates than CE, which owes to their re-balancing operations adjusting the classifier weights' updating to match test distributions.
- **Representations:** When applying the same classifier learning manner (comparing error rates of three blocks in the horizontal direction), it is a bit of surprise to see that error rates of CE blocks are consistently lower than error rates of RW/RS blocks. The findings indicate that training with CE achieves better classification results since it obtains better features. The worse results of RW/RS reveal that they lead to inferior discriminative ability of the learned deep features.

Furthermore, as shown in Figure 2 (left), by employing CE on the representation learning and employing RS on the classifier learning, we can achieve the lowest error rate on the validation set of CIFAR-100-IR50. Additionally, to evaluate the generalization ability for representations produced by three manners, we utilize pre-trained models trained on CIFAR-*100*-IR50 as the feature extractor to obtain the representations of CIFAR-*10*-IR50, and then perform the classifier learning experiments as the same as aforementioned. As shown in Figure 2 (right), on CIFAR-10-IR50, it can have the identical observations, even in the situation that the feature extractor is trained on another long-tailed dataset.

4. Methodology

4.1. Overall framework

As shown in Figure 3, our BBN consists of three main components. Concretely, we design two branches for representation learning and classifier learning, termed "*conventional learning branch*" and "*re-balancing branch*", respectively. Both branches use the same residual network structure [12] and share all the weights except for the last residual block. Let $\mathbf{x}_i$ denote a training sample and $y_i \in \{1, 2, \ldots, C\}$ is its corresponding label, where C is the number of classes. For the bilateral branches, we apply uniform and reversed samplers to each of them separately and obtain two samples $(\mathbf{x}_c, y_c)$ and $(\mathbf{x}_r, y_r)$ as the input data, where $(\mathbf{x}_c, y_c)$ is for the conventional learning branch and $(\mathbf{x}_r, y_r)$ is for the re-balancing branch. Then, two samples

are fed into their own corresponding branch to acquire the feature vectors $\boldsymbol{f}_c \in \mathbb{R}^D$ and $\boldsymbol{f}_r \in \mathbb{R}^D$ by global average pooling.

Furthermore, we also design a specific cumulative learning strategy for shifting the learning "attention" between two branches in the training phase. In concretely, by controlling the weights for $\boldsymbol{f}_c$ and $\boldsymbol{f}_r$ with an adaptive trade-off parameter α, the weighted feature vectors $\alpha \boldsymbol{f}_c$ and $(1-\alpha)\boldsymbol{f}_r$ will be sent into the classifiers $\boldsymbol{W}_c \in \mathbb{R}^{D\times C}$ and $\boldsymbol{W}_r \in \mathbb{R}^{D\times C}$ respectively and the outputs will be integrated together by element-wise addition. The output logits are formulated as

$$\mathbf{z} = \alpha \boldsymbol{W}_c^\top \boldsymbol{f}_c + (1-\alpha)\boldsymbol{W}_r^\top \boldsymbol{f}_r, \tag{1}$$

where $\mathbf{z} \in \mathbb{R}^C$ is the predicted output, *i.e.*, $[z_1, z_2, \ldots, z_C]^\top$. For each class $i \in \{1, 2, \ldots, C\}$, the softmax function calculates the probability of the class by

$$\hat{p}_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}. \tag{2}$$

Then, we denote $E(\cdot,\cdot)$ as the cross-entropy loss function and the output probability distribution as $\hat{\boldsymbol{p}} = [\hat{p}_1, \hat{p}_2, ..., \hat{p}_C]^\top$. Thus, the weighted cross-entropy classification loss of our BBN model is illustrated as

$$\mathcal{L} = \alpha E(\hat{\boldsymbol{p}}, y_c) + (1-\alpha) E(\hat{\boldsymbol{p}}, y_r), \tag{3}$$

and the whole network is end-to-end trainable.

4.2. Proposed bilateral-branch structure

In this section, we elaborate the details of our unified bilateral-branch structure shown in Figure 3. As aforementioned, the proposed conventional learning branch and re-balancing branch do perform their own duty (*i.e.*, representation learning and classifier learning, respectively). There are two unique designs for these branches.

Data samplers. The input data for the conventional learning branch comes from a uniform sampler, where each sample in the training dataset is sampled only once with equal probability in a training epoch. The uniform sampler retains the characteristics of original distributions, and therefore benefits the representation learning. While, the re-balancing branch aims to alleviate the extreme imbalance and particularly improve the classification accuracy on tail classes [26], whose input data comes from a reversed sampler. For the reversed sampler, the sampling possibility of each class is proportional to the reciprocal of its sample size, *i.e.*, the more samples in a class, the smaller sampling possibility that class has. In formulations, let denote that the number of samples for class i is N_i and the maximum sample number of all the classes is N_{max}. There are three sub-procedures to construct the reversed sampler: 1) Calculate the sampling possibility P_i for class i according to the number of samples as

$$P_i = \frac{w_i}{\sum_{j=1}^{C} w_j}, \tag{4}$$

where $w_i = \frac{N_{max}}{N_i}$; 2) Randomly sample a class according to P_i; 3) Uniformly pick up a sample from class i with replacement. By repeating this reversed sampling process, training data of a mini-batch is obtained.

Weights sharing. In BBN, both branches economically share the same residual network structure, as illustrated in Figure 3. We use ResNets [12] as our backbone network, *e.g.*, ResNet-32 and ResNet-50. In details, two branch networks, except for the last residual block, share the same weights. There are two benefits for sharing weights: On the one hand, the well-learned representation by the conventional learning branch can benefit the learning of the re-balancing branch. On the other hand, sharing weights will largely reduce computational complexity in the inference phase.

4.3. Proposed cumulative learning strategy

Cumulative learning strategy is proposed to shift the learning focus between the bilateral branches by controlling both the weights for features produced by two branches and the classification loss $\mathcal{L}$. It is designed to first learn the universal patterns and then pay attention to the tail data gradually. In the training phase, the feature $\boldsymbol{f}_c$ of the conventional learning branch will be multiplied by α and the feature $\boldsymbol{f}_r$ of the re-balancing branch will be multiplied by $1-\alpha$, where α is automatically generated according to the training epoch. Concretely, the number of total training epochs is denoted as T_{max} and the current epoch is T. α is calculated by

$$\alpha = 1 - \left(\frac{T}{T_{max}}\right)^2, \tag{5}$$

which α will gradually decrease as the training epochs increasing.

In intuition, we design the adapting strategy for α based on the motivation that discriminative feature representations are the foundation for learning robust classifiers. Although representation learning and classifier learning deserve equal attentions, the learning focus of our BBN should gradually change from feature representations to classifiers, which can exhaustively improve long-tailed recognition accuracy. With α decreasing, the main emphasis of BBN turns from the conventional learning branch to the re-balancing branch. Different from two-stage fine-tuning strategies [3, 6, 22], our α ensures that both branches for different goals can be constantly updated in the whole training process, which could avoid the affects on one goal when it performs training for the other goal.

In experiments, we also provide the qualitative results of this intuition by comparing different kinds of adaptors, cf. Section 5.5.2.

Table 1. Top-1 error rates of ResNet-32 on long-tailed CIFAR-10 and CIFAR-100. (Best results are marked in bold.)

Dataset	Long-tailed CIFAR-10			Long-tailed CIFAR-100		
Imbalance ratio	100	50	10	100	50	10
CE	29.64	25.19	13.61	61.68	56.15	44.29
Focal [17]	29.62	23.28	13.34	61.59	55.68	44.22
Mixup [33]	26.94	22.18	12.90	60.46	55.01	41.98
Manifold Mixup [27]	27.04	22.05	12.97	61.75	56.91	43.45
Manifold Mixup (two samplers)	26.90	20.79	13.17	63.19	57.95	43.54
CE-DRW [3]	23.66	20.03	12.44	58.49	54.71	41.88
CE-DRS [3]	24.39	20.19	12.62	58.39	54.52	41.89
CB-Focal [5]	25.43	20.73	12.90	60.40	54.83	42.01
LDAM-DRW [3]	22.97	18.97	11.84	57.96	53.38	41.29
Our BBN	**20.18**	**17.82**	**11.68**	**57.44**	**52.98**	**40.88**

4.4. Inference phase

During inference, the test samples are fed into both branches and two features $\boldsymbol{f}_c'$ and $\boldsymbol{f}_r'$ are obtained. Because both branches are equally important, we simply fix α to 0.5 in the test phase. Then, the equally weighted features are fed to their corresponding classifiers (*i.e.*, $\boldsymbol{W}_c$ and $\boldsymbol{W}_r$) to obtain two prediction logits. Finally, both logits are aggregated by element-wise addition to return the classification results.

5. Experiments

5.1. Datasets and empirical settings

Long-tailed CIFAR-10 and CIFAR-100. Both CIFAR-10 and CIFAR-100 contain 60,000 images, 50,000 for training and 10,000 for validation with category number of 10 and 100, respectively. For fair comparisons, we use the long-tailed versions of CIFAR datasets as the same as those used in [3] with controllable degrees of data imbalance. We use an imbalance factor β to describe the severity of the long tail problem with the number of training samples for the most frequent class and the least frequent class, *e.g.*, $\beta = \frac{N_{max}}{N_{min}}$. Imbalance factors we use in experiments are 10, 50 and 100.

iNaturalist 2017 and iNaturalist 2018. The iNaturalist species classification datasets are large-scale real-world datasets that suffer from extremely imbalanced label distributions. The 2017 version of iNaturalist contains 579,184 images with 5,089 categories and the 2018 version is composed of 437,513 images from 8,142 categories. Note that, besides the extreme imbalance, the iNaturalist datasets also face the fine-grained problem [32, 35, 30, 31]. In this paper, the official splits of training and validation images are utilized for fair comparisons.

5.2. Implementation details

Implementation details on CIFAR. For long-tailed CIFAR-10 and CIFAR-100 datasets, we follow the data augmentation strategies proposed in [12]: randomly crop a 32×32 patch from the original image or its horizontal flip with 4 pixels padded on each side. We train the ResNet-32 [12] as our backbone network for all experiments by standard mini-batch stochastic gradient descent (SGD) with momentum of 0.9, weight decay of 2×10^{-4}. We train all the models on a single NVIDIA 1080Ti GPU for 200 epochs with batch size of 128. The initial learning rate is set to 0.1 and the first five epochs is trained with the linear warm-up learning rate schedule [8]. The learning rate is decayed at the 120^{th} and 160^{th} epoch by 0.01 for our BBN.

Implementation details on iNaturalist. For fair comparisons, we utilize ResNet-50 [12] as our backbone network in all experiments on iNaturalist 2017 and iNaturalist 2018. We follow the same training strategy in [8] with batch size of 128 on four GPUs of NVIDIA 1080Ti. We firstly resize the image by setting the shorter side to 256 pixels and then take a 224×224 crop from it or its horizontal flip. During training, we decay the learning rate at the 60^{th} and 80^{th} epoch by 0.1 for our BBN, respectively.

5.3. Comparison methods

In experiments, we compare our BBN model with three groups of methods:

- **Baseline methods.** We employ plaining training with cross-entropy loss and focal loss [17] as our baselines. Note that, we also conduct experiments with a series of mixup algorithms [33, 27] for comparisons.
- **Two-stage fine-tuning strategies.** To prove the effectiveness of our cumulative learning strategy, we also compare with the two-stage fine-tuning strategies proposed in previous state-of-the-art [3]. We train networks with cross-entropy (CE) on imbalanced data in the first stage, and then conduct class re-balancing training in the second stage. "CE-DRW" and "CE-DRS" refer to the two-stage baselines using re-weighting and re-sampling at the second stage.
- **State-of-the-art methods.** For state-of-the-art methods, we compare with the recently proposed LDAM [3] and CB-Focal [5] which achieve good classification accuracy on these four aforementioned long-tailed datasets.

Table 2. Top-1 error rates of ResNet-50 on large-scale long-tailed datasets iNaturalist 2018 and iNaturalist 2017. Our method outperforms the previous state-of-the-arts by a large margin, especially with 2× scheduler. "*" indicates original results in that paper.

Dataset	iNaturalist 2018	iNaturalist 2017
CE	42.84	45.38
CE-DRW [3]	36.27	40.48
CE-DRS [3]	36.44	40.12
CB-Focal [5]	38.88	41.92
LDAM-DRW* [3]	32.00	–
LDAM-DRW [3]	35.42	39.49
LDAM-DRW [3] (2×)	33.88	38.19
Our BBN	33.71	36.61
Our BBN (2×)	**30.38**	**34.25**

5.4. Main results

5.4.1 Experimental results on long-tailed CIFAR

We conduct extensive experiments on long-tailed CIFAR datasets with three different imbalanced ratios: 10, 50 and 100. Table 1 reports the error rates of various methods. We demonstrate that our BBN consistently achieves the best results across all the datasets, when comparing other comparison methods, including the two-stage fine-tuning strategies (*i.e.*, CE-DRW/CE-DRS), the series of mixup algorithms (*i.e.*, mixup, manifold mixup and manifold mixup with two samplers as the same as ours), and also previous state-of-the-arts (*i.e.*, CB-Focal [5] and LDAM-DRW [3]).

Especially for long-tailed CIFAR-10 with imbalanced ratio 100 (an extreme imbalance case), we get 20.18% error rate which is 2.79% lower than that of LDAM-DRW [3]. Additionally, it can be found from that table, the two-stage fine-tuning strategies (*i.e.*, CE-DRW/CE-DRS) are effective, since they could obtain comparable or even better results comparing with state-of-the-art methods.

5.4.2 Experimental results on iNaturalist

Table 2 shows the results on two large-scale long-tailed datasets, *i.e.*, iNaturalist 2018 and iNaturalist 2017. As shown in that table, the two-stage fine-tuning strategies (*i.e.*, CE-DRW/CE-DRS) also perform well, which have consistent observations with those on long-tailed CIFAR. Compared with other methods, on iNaturalist, our BBN still outperform competing approaches and baselines. Besides, since iNaturalist is large-scale, we also conduct network training with the 2× scheduler. Meanwhile, for fair comparisons, we further evaluate the previous state-of-the-art LDAM-DRW [3] with the 2× training scheduler. It is obviously to see that, with 2× scheduler, our BBN achieves significantly better results than BBN without 2× scheduler. Additionally, compared with LDAM-DRW (2×), we achieve $+3.50\%$ and $+3.94\%$ improvements on iNaturalist 2018 and iNaturalist 2017, respectively. In addition, even though we do not use the 2× scheduler, our BBN can still get the best results. For a detail, we conducted the experiments based on LDAM [3] with the source codes provided by the authors, but failed to reproduce the results reported in that paper.

Table 3. Ablation studies for different samplers for the re-balancing branch of BBN on long-tailed CIFAR-10-IR50.

Sampler	Error rate
Uniform sampler	21.31
Balanced sampler	21.06
Reversed sampler (Ours)	**17.82**

Table 4. Ablation studies of different adaptor strategies of BBN on long-tailed CIFAR-10-IR50.

Adaptor	α	Error rate
Equal weight	0.5	21.56
β-distribution	$Beta(0.2, 0.2)$	21.75
Parabolic increment	$\left(\frac{T}{T_{max}}\right)^2$	22.70
Linear decay	$1 - \frac{T}{T_{max}}$	18.55
Cosine decay	$\cos(\frac{T}{T_{max}} \cdot \frac{\pi}{2})$	18.04
Parabolic decay (Ours)	$1 - \left(\frac{T}{T_{max}}\right)^2$	**17.82**

5.5. Ablation studies

5.5.1 Different samplers for the re-balancing branch

For better understanding our proposed BBN model, we conduct experiments on different samplers utilized in the re-balancing branch. We present the error rates of models trained with different samplers in Table 3. For clarity, the uniform sampler maintains the original long-tailed distribution. The balanced sampler assigns the same sampling possibility to all classes, and construct a mini-batch training data obeying a balanced label distribution. As shown in that table, the reversed sampler (our proposal) achieves considerably better performance than the uniform and balanced samplers, which indicates that the re-balancing branch of BBN should pay more attention to the tail classes by enjoying the reversed sampler.

5.5.2 Different cumulative learning strategies

To facilitate the understanding of our proposed cumulative learning strategy, we explore several different strategies to generate the adaptive trade-off parameter α on CIFAR-10-IR50. Specifically, we test with both progress-relevant/irrelevant strategies, cf. Table 4. For clarity, progress-relevant strategies adjust α with the number of training epochs, *e.g.*, linear decay, cosine decay, *etc*. Progress-irrelevant strategies include the equal weight or generate from a discrete distribution (*e.g.*, the β-distribution).

Table 5. Feature quality evaluation for different learning manners.

Representation learning manner	Error rate
CE	**58.62**
RW	63.17
RS	63.71
BBN-CB	58.89
BBN-RB	61.09

As shown in Table 4, the decay strategies (*i.e.*, linear decay, cosine decay and our parabolic decay) for generating α can yield better results than the other strategies (*i.e.*, equal weight, β-distribution and parabolic increment). These observations prove our motivation that the conventional learning branch should be learned firstly and then the re-balancing branch. Among these strategies, the best way for generating α is the proposed parabolic decay approach. In addition, the parabolic increment, where re-balancing are attended before conventional learning, performs the worst, which validates our proposal from another perspective. More detailed discussions can be found in the supplementary materials.

5.6. Validation experiments of our proposals

5.6.1 Evaluations of feature quality

It is proven in Section 3 that learning with vanilla CE on original data distribution can obtain good feature representations. In this subsection, we further explore the representation quality of our proposed BBN by following the empirical settings in Section 3. Concretely, given a BBN model trained on CIFAR-100-IR50, firstly, we fix the parameters of representation learning of two branches. Then, we separately retrain the corresponding classifiers from scratch of two branches also on CIFAR-100-IR50. Finally, classification error rates are tested on these two branches independently.

As shown in Table 5, the feature representations obtained by the conventional learning branch of BBN ("BBN-CB") achieves comparable performance with CE, which indicates that our proposed BBN greatly preserves the representation capacity learned from the original long-tailed dataset. Note that, the re-balancing branch of BBN ("BBN-RB") also gets better performance than RW/RS and it possibly owes to the parameters sharing design of our model.

5.6.2 Visualization of classifier weights

Let denote $\boldsymbol{W} \in \mathbb{R}^{D\times C}$ as a set of classifiers $\{\boldsymbol{w}_1, \boldsymbol{w}_2, ..., \boldsymbol{w}_C\}$ for all the C classes, where $\boldsymbol{w}_i \in \mathbb{R}^D$ indicates the weight vector for class i. Previous work [9] has shown that the value of ℓ_2-norm $\{\|\boldsymbol{w}_i\|_2\}_{i=1}^{C}$ for different classes can demonstrate the preference of a classifier, *i.e.*, the classifier $\boldsymbol{w}_i$ with the largest ℓ_2-norm tends to judge one example as belonging to its class i. Following [9], we visualize the ℓ_2-norm of these classifiers.

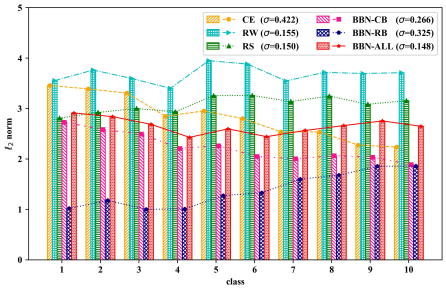

Figure 4. ℓ_2-norm of classifier weights for different learning manners. Specifically, "BBN-ALL" indicates the ℓ_2-norm of the combination of $\boldsymbol{W}_c$ and $\boldsymbol{W}_r$ in our model. σ in the legend is the standard deviation of ℓ_2-norm for ten classes.

As shown in Figure 4, we visualize the ℓ_2-norm of ten classes trained on CIFAR-10-IR50. For our BBN, we visualize the classifier weights $\boldsymbol{W}_c$ of the conventional learning branch ("BBN-CB") and the classifier weights $\boldsymbol{W}_r$ of the re-balancing branch ("BBN-RB"), as well as their combined classifier weights ("BBN-ALL"). Additionally, the visualization results on classifiers trained with these learning manners in Section 3, *i.e.*, CE, RW and RS, are also provided.

Obviously, the ℓ_2-norm of ten classes' classifiers for our proposed model (*i.e.*, "BBN-ALL") are basically equal, and their standard deviation $\sigma = 0.148$ is the smallest one. For the classifiers trained by other learning manners, the distribution of the ℓ_2-norm of CE is consistent with the long-tailed distribution. The ℓ_2-norm distribution of RW/RS looks a bit flat, but their standard deviations are larger than ours. It gives an explanation why our BBN can outperform these methods. Additionally, by separately analyzing our model, its conventional learning branch ("BBN-CB") has a similar ℓ_2-norm distribution with CE's, which justifies its duty is focusing on universal feature learning. The ℓ_2-norm distribution of the re-balancing branch ("BBN-RB") has a reversed distribution w.r.t. original long-tailed distributions, which reveals it is able to model the tail.

6. Conclusions

In this paper, for studying long-tailed problems, we explored how class re-balancing strategies influenced representation learning and classifier learning of deep networks, and revealed that they can promote classifier learning significantly but also damage representation learning to some extent. Motivated by this, we proposed a Bilateral-Branch Network (BBN) with a specific cumulative learning strategy to take care of both representation learning and classifier learning for exhaustively improving the recognition performance of long-tailed tasks. By conducting extensive experiments, we proved that our BBN could achieve the best results on long-tailed benchmarks, including the large-scale iNaturalist. In the future, we attempt to tackle the long-tailed detection problems with our BBN model.

References

[1] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018. 2, 3

[2] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *ICML*, pages 872–881, 2019. 2, 3

[3] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *NeurIPS*, pages 1–18, 2019. 2, 3, 4, 5, 6, 7

[4] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002. 2, 3

[5] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, pages 9268–9277, 2019. 2, 3, 4, 6, 7

[6] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *CVPR*, pages 4109–4118, 2018. 1, 3, 5

[7] Chris Drummond and Robert C Holte. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. *Workshop on Learning From Imbalanced Datasets II*, 11:1–8, 2003. 2

[8] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour. *arXiv preprint arXiv:1706.02677*, pages 1–12, 2017. 6

[9] Yandong Guo and Lei Zhang. One-shot face recognition by promoting underrepresented classes. *arXiv preprint arXiv:1707.05574*, pages 1–12, 2017. 8

[10] Agrim Gupta, Piotr Dollár, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, pages 5356–5364, 2019. 1

[11] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009. 2, 3

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4, 5, 6

[13] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *CVPR*, pages 5375–5384, 2016. 2, 3

[14] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002. 2, 3

[15] Maurice George Kendall, Alan Stuart, John Keith Ord, Steven F Arnold, Anthony O'Hagan, and Jonathan Forster. *Kendall's advanced theory of statistics*, volume 1. 1987. 1

[16] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 4

[17] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 6

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014. 1

[19] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, pages 1–10, 2019. 3

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, pages 3111–3119, 2013. 3

[21] Ajinkya More. Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint arXiv:1608.06048*, pages 1–7, 2016. 2

[22] Wanli Ouyang, Xiaogang Wang, Cong Zhang, and Xiaokang Yang. Factors in finetuning deep model for object detection with long-tail distribution. In *CVPR*, pages 864–873, 2016. 3, 5

[23] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, pages 1–13, 2018. 2

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, and Michael Bernstein. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1

[25] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *ECCV*, pages 467–482, 2016. 2, 3

[26] Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, pages 1–22, 2017. 1, 5

[27] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, pages 6438–6447, 2019. 3, 6

[28] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *NeurIPS*, pages 7029–7039, 2017. 2, 3

[29] Xiu-Shen Wei, Quan Cui, Lei Yang, Peng Wang, and Lingqiao Liu. RPC: A large-scale retail product checkout dataset. *arXiv preprint arXiv:1901.07249*, pages 1–24, 2019. 1

[30] Xiu-Shen Wei, Jian-Hao Luo, Jianxin Wu, and Zhi-Hua Zhou. Selective convolutional descriptor aggregation for fine-grained image retrieval. *IEEE Transactions on Image Processing*, 26(6):2868–2881, 2017. 6

[31] Xiu-Shen Wei, Peng Wang, Lingqiao Liu, Chunhua Shen, and Jianxin Wu. Piecewise classifier mappings: Learning fine-grained learners for novel categories with few examples. *IEEE Transactions on Image Processing*, 28(12):6116–6125, 2019. 6

[32] Xiu-Shen Wei, Jianxin Wu, and Quan Cui. Deep learning for fine-grained image analysis: A survey. *arXiv preprint arXiv:1907.03069*, pages 1–7, 2019. 6

[33] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, pages 1–13, 2018. 3, 6

[34] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. Range loss for deep face recognition with long-tailed training data. In *ICCV*, pages 5409–5418, 2017. 3

[35] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, 14(2):119–135, 2017. 6

[36] Yaoyao Zhong, Weihong Deng, Mei Wang, Jiani Hu, Jianteng Peng, Xunqiang Tao, and Yaohai Huang. Unequal-training for deep face recognition with long-tailed noisy data. In *CVPR*, pages 7812–7821, 2019. 3

[37] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for ccene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 1