Strong-weak Dual-branch Network with Hard-aware Loss for Long-tailed Classification

Qingheng Zhang[1], Haibo Ye[1,2]

{zhangqh@nuaa.edu.cn, yhb@nuaa.edu.cn}

[1.] Nanjing University of Aeronautics and Astronautics, Nanjing, China;

[2.] State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing, China

Abstract. Natural data usually exhibit a long-tailed distribution, with the minority classes occupying the majority of the data, while the majority classes have few samples. Although deep learning has made remarkable progress in visual recognition on large-scale balanced datasets, it remains challenging to model long-tailed distributions. Recent multi-branch methods have shown great potential to address long-tailed problems. We find that these methods work due to the difference between branches, so we also propose a new structure called Strong-weak Dual-branch Network (SDN) to enlarge the difference between branches. In particular, our SDN is equipped with a new Difference to Classification (D2C) learning strategy, designed to amplify the differences between the branches first, and then pay attention to classification. In addition, we propose a new Hard-aware Loss (HL) for the sake of handling hard examples. Our SDNHL method achieves SOTA on four long-tailed datasets: CIFAR-10-LT, CIFAR-100-LT, ImageNet-LT and iNaturalist 2018.

Keywords: Long-tailed distribution; Dual-branch Network; Hard-aware Loss

1 Introduction

Real-world data often exhibit long-tailed distributions. Existing methods usually favor the majority classes, resulting in poor generalization performance for rare classes. Early works alleviate the deterioration of long-tailed training data by re-balancing data distribution [2], [7], [8]. These rebalancing methods often distort the original data distribution and thus overfit the tail classes. Recently, two-stage methods [2], [6], [13] have achieved significant improvements. Deferred re-balancing methods first train the network with long-tailed distribution, and then use re-balancing strategies to adjust the network in the second stage. Ensemble approaches [11], [13] reorganize datasets into groups and each group is assigned a model for training. We find that these multi-branch methods inherently increase the difference between branches, so we propose a simple Strong-weak Dual-branch Network (SDN) to increase the difference between branches. In addition, we find that some classes have many images, but the accuracy rate is not high, while some classes have a small number of pictures, but have a high accuracy rate. So we propose the hard-aware (HL) loss function to adjust the weights of difficult and easy classes.

In this paper, we propose Strong-weak Dual-branch Network with Hard-aware Loss (SDNHL) for long-tailed classification. Specifically, we feed the strongly augmented and weakly

ITEI 2022, December 23-25, Harbin, People's Republic of China

DOI 10.4108/eai.23-12-2022.2329170

augmented samples into two separate branches, while ensuring the difference between the branches through KL divergence. In addition, the weights of each class are dynamically adjusted through a hard-aware loss function.

2 Overall Framework

The overall framework of our Strong-weak Dual-branch Network with Hard-aware Loss (SDNHL) is illustrated in **[1] Figure**. Specifically, we design two branches to learn strong feature representation and weak feature representation respectively. The difference between the two branches is that we process the input data, one branch uses strong augmented data and the other uses weak augmented data. Furthermore, we increase the difference between the two branches by maximizing KL divergence. And we use a new learning strategy to adjust the focus of learning from difference to classification. In addition, we design a new Hard-aware Loss (HL) function for difficult samples and use it in the final stage of training.

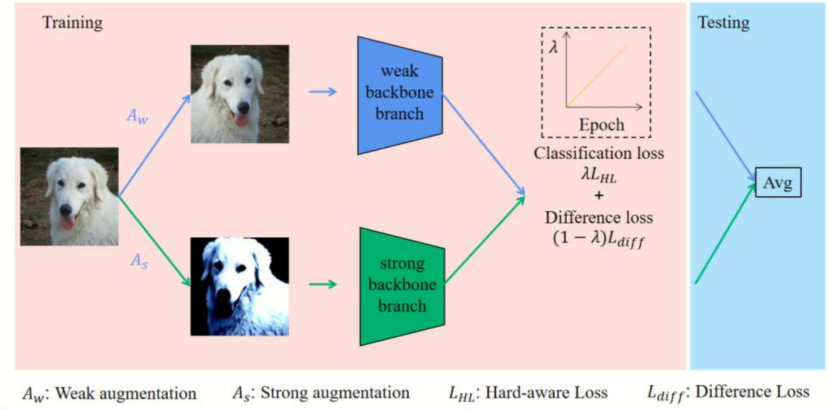

Fig. 1. The framework of SDNHL method.

2.1 Strong-weak Dual-branch Network

As shown in **[1] Figure**, the two branches are called "weak branch" and "strong branch". For the weak branch, we follow [4] to apply augmentation strategies, while for the strong branch, we randomly add grayscale, blur and color distortion. Let x denote a training sample and $y \in 1,2,\ldots,C$, where C is the number of classes. We apply strong and weak augmentation strategies for "Strong branch" and "weak branch", and send the two obtained samples (x_s, y) and (x_w, y) to their corresponding branches, respectively. Then, we send the obtained feature vectors $f_s \in \mathbb{R}^D$ and $f_w \in \mathbb{R}^D$ into the classifiers $W_s \in \mathbb{R}^{D\times C}$ and $W_w \in \mathbb{R}^{D\times C}$ respectively:

$$z_s = W_s{}^T f_s, z_w = W_w^T f_w \ .$$

$z_s, z_w \in \mathbb{R}^{D\times C}$ denotes the predicted output of the strong branch and weak branch respectively. We calculate the probability of the class by the softmax function as

$$P_s^i = \frac{e^{z_s^i}}{\sum_{j=1}^{C} e^{z_s^j}}, P_w^i = \frac{e^{z_w^i}}{\sum_{j=1}^{C} e^{z_w^j}}.$$

Furthermore, we add a regularization term to ensure the difference between the two branches. We maximize the KL divergence of the classification probabilities of the two branches over a total of C categories as

$$L_{diff}\ (P||P_w) = D(P_s\ ||P_w) = \sum\nolimits_{i=1}^{C} P_s^i \log \frac{P_s^i}{P_w^i}.$$

L_{diff} denotes the difference loss. We utilize the proposed hard-aware (HL) loss in section 2.3 for classification loss as L_{cls} and the final loss is

$$L_{SDN} = \lambda(L_{cls}(y, P_s) + L_{cls}(y, P_w)) + (1 - \lambda)L_{diff}(P_s||P_w).$$

λ and $1 - \lambda$ are the weight of classification loss and difference loss respectively. A detailed description of the parameter λ can be found in Section 2.2.

2.2 D2C Learning Strategy for SDN

We propose a new learning strategy that shifts the learning focus from Difference to Classification (D2C). Specifically, we want to obtain as different branches as possible through difference loss early in training, and gradually shift the focus of training to classification as the training progresses. We define the λ as

$$\lambda = \left(\frac{E_{curr}}{E_{total}}\right)^{\gamma}.$$

E_{total} denotes the number of total epochs, and E_{curr} is the current epoch. We can see that λ is automatically generated based on the training epoch and will gradually increase with the training epoch. γ controls how quickly the learning strategy shifts from difference to classification. More experiments on γ can be found in ablation studies.

2.3 Hard-aware Loss

In this section, we describe hard-aware loss in detail. Our motivation stems from the observation that in a long-tailed data recognition task, a class with a small number of samples is not necessarily a hard-to-learn class, and similarly, a class with many samples is not necessarily an easy-to-learn class. However, the common re-weighting loss function just adjusts the weights according to the number of classes. To solve the above problem, we propose a new Hard-aware Loss (HL).

We introduce the hard-aware loss starting with a common re-weighting loss:

$$L_{WCE} = -\frac{1}{M}\sum\nolimits_{c=1}^{C}\sum\nolimits_{m=1}^{M} w^c \times y_m^c \times \log(p_m^c).$$

M denotes the number of training examples, C represents the number of classes, w^c denotes the weight of class c, y_m^c is the target label for class c of training example m, and p_m^c is estimated probability for the class c of training example m. p_m^c is calculated by $Softmax(z)$.

Formally, we introduce a weight term $\widetilde{w}$ into the re-weighting loss function to obtain our proposed hard-aware loss:

$$L_{HL} = -\frac{1}{M}\sum\nolimits_{c=1}^{C}\sum\nolimits_{m=1}^{M}\widetilde{w}^{c} \times w^{c} \times y_{m}^{c} \times log(p_{m}^{c}) .$$

$\widetilde{w}^{c}$ is the hard-aware weight of class c. we set $\widetilde{w}^{c}$ with the following regulations:

$$\widetilde{w}_{e+1}^{c} = m \times \widetilde{w}_{e}^{c} + \frac{1}{acc_{e}^{c}} .$$

$\widetilde{w}_{e}^{c}$ denotes the weight for class c at the e-th epoch, $m \in (0,1)$ is the momentum factor to adjust weights smoothly, and acc_{e}^{c} is the accuracy of the class c at the e-th epoch.

3 Experiments

3.1 Datasets and Implementation Details

We follow [2], [13] to generate long-tailed version of CIFAR datasets. Following [12], we use official ImageNet-LT training and validation images. For iNaturalist 2018, we use the official training and validation set in [9].

We train the network for 200 epochs for all experiments with warm-up schedule. The classification loss used for the first 160 epochs is LDAM-DRW loss [2], and for the last 40 epochs is replaced by hard-aware loss.

Following[2], [7], [10], [11], [12], [13], we train the ResNet-32 as our backbone for CIFAR-10-LT and CIFAR-100-LT, train ResNet-10 and ResNet-50 for ImageNet-LT, and train ResNet-50 for iNaturalist 2018. For weak branch training samples, we use augmentation in [2]. For strong branch training samples, we randomly add grayscale, blur and color distortion with a probability of 0.2, 0.5, and 0.8 respectively.

3.2 Performance Comparison

Results on CIFAR-LT. The top-1 accuracy on CIFAR-LT with ResNet-32 is reported in Table 1. The imbalanced ratios are 200, 100, 50 and 20. Our proposed method SDNHL performs the best across all the datasets. We also report the accuracy of many-shot (>100 images), medium-shot (20~100 images) and few-shot (< 20 images) on CIFAR-100-LT-100 in Table 2. Our SDNHL outperforms the state-of-the-art methods by more than 2%.

Results on ImageNet-LT and iNaturalist 2018. We further validate the effectiveness of our method on large-scale datasets in Table 3. Ours outperforms the RIDE by 2.2% (ResNet-10) and 1.9% (ResNet-50) on ImageNet-LT, and 2.5% (ResNet-50) on iNaturalist 2018, respectively. In Table 4, we report the accuracy of many-shot, medium-shot and few-shot.

Table 1. Top-1 accuracy on CIFAR-10-LT and CIFAR-100-LT.

Datasets	CIFAR-10-LT				CIFAR-100-LT			
Imbalance ratio	200	100	50	20	200	100	50	20
Class-Balanced	68.89	74.57	79.27	84.36	36.23	39.60	45.32	52.59
LDAM-DRW	-	77.03	-	-	-	42.04	-	-

Equalization	-	-	-	-	43.38	-	-	-
BBN	-	79.82	82.18	-	-	42.56	47.02	-
RIDE	-	-	-	-	-	49.10	-	-
Remix-DRW	-	79.76	-	-	-	46.77	-	-
CAM	-	80.03	83.59	-	-	47.83	51.69	-
Domain Adap.	77.23	80.00	82.88	86.46	39.53	44.70	50.08	55.73
LADE	-	-	-	-	-	45.40	50.50	-
Ours	78.11	81.10	85.08	87.71	46.77	50.55	53.56	58.57

Table 2. The accuracies of many-shot, medium shot and few-shot on CIFAR-100-LT. ∗ denotes the results from RIDE [10]. Other results are copied from ACE [1].

Method	Many	Medium	Few	All
Focal loss	64.3	37.4	7.1	37.4
CB loss	65.0	37.6	10.3	38.7
Remix	69.6	40.7	8.8	40.9
OLTR∗	61.8	41.4	17.6	41.2
Mixup	**70.7**	40.4	8.8	41.2
LDAM-DRW	61.5	41.7	20.2	42.0
τ-norm ∗	65.7	43.6	17.3	43.2
CRT∗	64.0	44.8	18.1	43.3
RIDE∗	69.3	49.3	26.0	49.1
Ours	66.2	**52.1**	**30.5**	**50.5**

Table 3. Top-1 accuracy on ImageNet-LT and iNaturalist 2018. ∗ denotes the results from BBN.

Datasets	ImageNet-LT		iNaturalist 2018
Backbone	ResNet-10	ResNet-50	ResNet-50
LDAW-DRW*	-	-	66.12
BBN*	-	-	69.62
CAM	43.13	-	70.87
Remix-DRW	-	-	70.49
DisAlign	-	52.90	70.60
RIDE	45.30	54.40	71.40
Ours	**47.53**	**56.31**	**73.98**

Table 4. The accuracies of many-shot, medium-shot and few-shot on ImageNet-LT and iNaturalist 2018. ∗ denotes the results from MiSLAS [5].

Datasets	ImageNet				iNaturalist 2018			
	Many	Medium	Few	All	Many	Medium	Few	All
cRT*	62.5	47.4	29.5	50.3	73.2	68.8	66.1	68.2
LWS*	61.8	48.6	33.5	51.2	71.0	69.8	68.8	69.5
MiSLAS*	61.7	51.3	35.8	52.7	73.2	72.4	70.4	71.6
RIDE	65.8	51.0	34.6	54.4	70.2	71.3	71.7	71.4
Ours	**66.1**	**53.7**	**37.5**	**56.3**	**73.3**	**74.5**	**73.4**	**73.9**

Table 5. Ablation studies of the proposed SDNHL on CIFAR-LT, ImageNet-LT and iNaturalist 2018.

Datasets	CIFAR-10-LT				CIFAR-100-LT				ImageNet-LT		iNat18
ResNet	32	32	32	32	32	32	32	32	10	50	50
Imbalance ratio	200	100	50	20	200	100	50	20	256	256	500
Baseline	73.21	77.20	81.73	84.41	38.01	41.16	46.30	52.57	42.61	51.94	68.83
with SDN	77.82	80.84	84.14	86.53	45.44	49.62	52.67	57.82	46.97	56.01	73.48
with SDNHL	**78.11**	**81.10**	**85.08**	**87.71**	**46.77**	**50.55**	**53.56**	**58.57**	**47.53**	**56.31**	**73.98**

4 Ablation Study

4.1 Effectiveness of each component

The effectiveness of SDN and HL is shown in Table 5. The baseline is the plain model with LDAM-DRW loss. "with SDN" denotes the accuracies of Strong-weak Dual-branch Network with D2C learning strategy. "with SDNHL" denotes the model with SDN and Hard-aware Loss.

4.2 Different D2C learning strategies

As shown in Table 6, we conduct several experiments about γ, which controls how quickly the learning strategy shifts from Difference to Classification (D2C). Linear form achieves the best result and we set gamma to 1 in all experiments.

Table 6. Ablation studies of different Difference to Classification (D2C) learning strategies on CIFAR-100-LT with the imbalance ratio of 50.

γ	D2C form	Accuracy
0.0	Constant	52.67
0.3	Concave	53.49
1.0	Linear	**53.56**
2.0	Convex	52.38

5 Conclusions

In this paper, we have introduced a Strong-weak Dual-branch Network (SDN) with the special Difference to Classification (D2C) learning strategy for long-tailed problems. Furthermore, we proposed a new Hard-aware Loss (HL) for samples that are hard to learn. Extensive experiments verify that our SDNHL method is effective.

References

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

[2] Cao K, Wei C, Gaidon A, et al. Learning imbalanced datasets with label-distribution-aware margin loss[J]. Advances in neural information processing systems, 2019, 32.

[3] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of artificial intelligence research, 2002, 16: 321-357. !

[4] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[5] Hong Y, Han S, Choi K, et al. Disentangling label distribution for long-tailed visual recognition[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 6626-6636.

[6] Kang B, Xie S, Rohrbach M, et al. Decoupling representation and classifier for long-tailed recognition[J]. arXiv preprint arXiv:1910.09217, 2019.

[7] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.

[8] Tan J, Wang C, Li B, et al. Equalization loss for long-tailed object recognition[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 11662-11671.

[9] Van Horn G, Mac Aodha O, Song Y, et al. The inaturalist species classification and detection dataset[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8769-8778.

[10] Wang X, Lian L, Miao Z, et al. Long-tailed recognition by routing diverse distribution-aware experts[J]. arXiv preprint arXiv:2010.01809, 2020.

[11] Xiang L, Ding G, Han J. Learning from multiple experts: Self-paced knowledge distillation for long-tailed classification[C]//European Conference on Computer Vision. Springer, Cham, 2020: 247-263.

[12] Zhang Y, Wei X S, Zhou B, et al. Bag of tricks for long-tailed visual recognition with deep convolutional neural networks[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(4): 3447-3455.

[13] Zhou B, Cui Q, Wei X S, et al. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 9719-9728.