Rethinking Class-Balanced Methods for Long-Tailed Visual Recognition from a Domain Adaptation Perspective

- Muhammad Abdullah Jamal， Matthew Brown， Ming-Hsuan Yang， Liqiang Wang， Boqing Gong

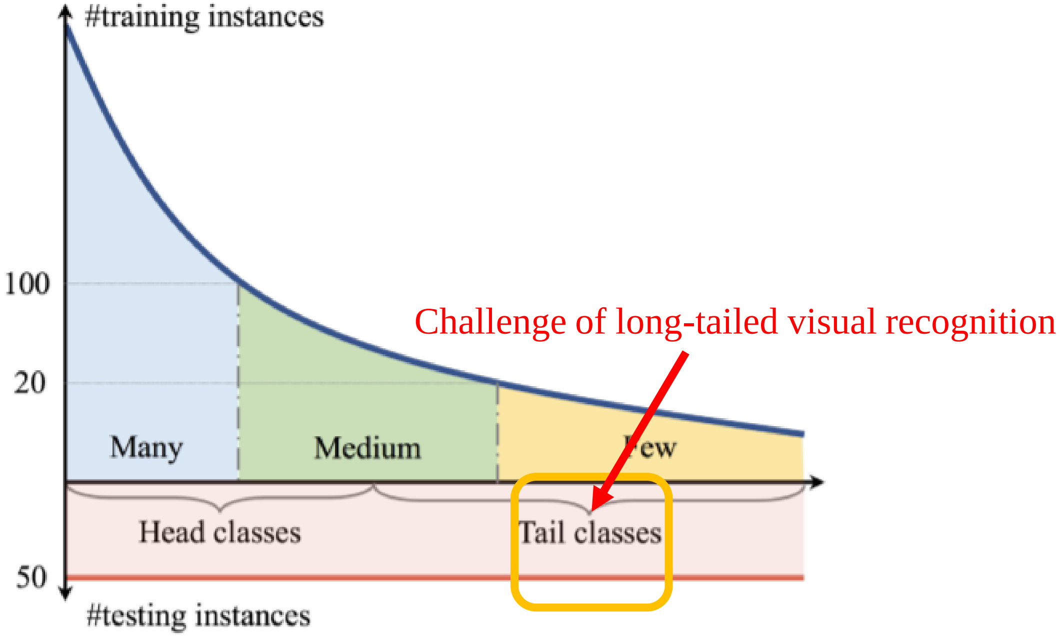
Presented by Taibiao Zhao

Published in Arxiv 2020

Contents

Introduction

Long tailed problem: (What? When?)
Uncommon objects in context,
positive patients in medic diagnosis;
emerges as the datasets grow in scale
prevalent in fine-grained recognition, detection.



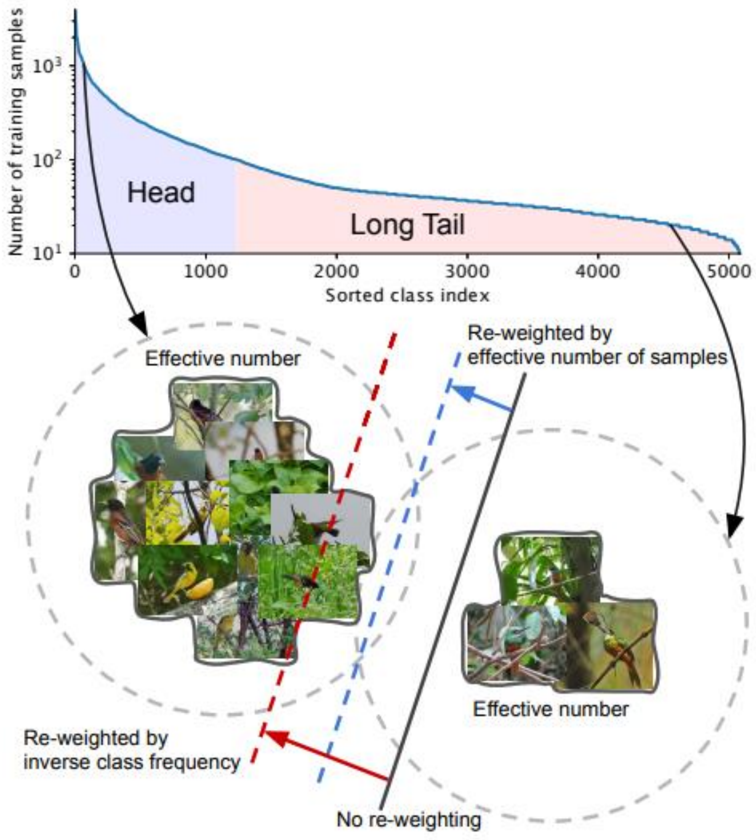
Present challenge for image classification tasks, especially for tail classes

Introduction

Problem statement:

- ❑ Training set: long-tailed distribution
  Head v.s. Tail

- ❑ Test set: balanced distribution



Existing methods

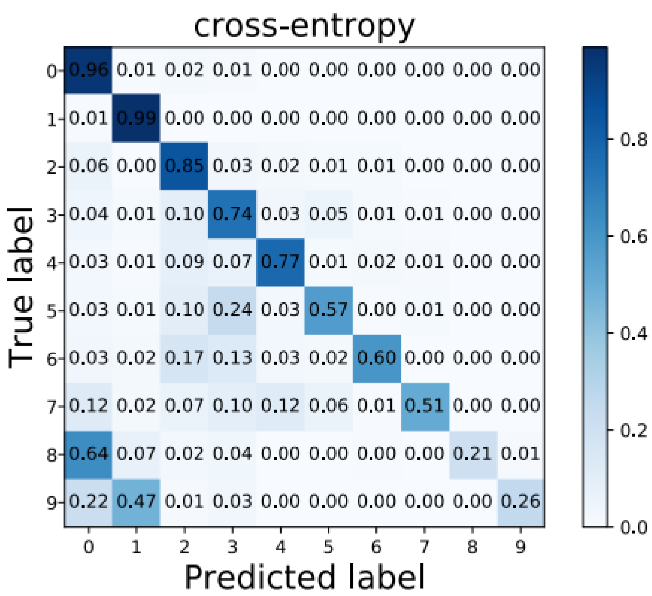
- ❑ Rebalancing the data
  Up/Down sampling tail/head classes

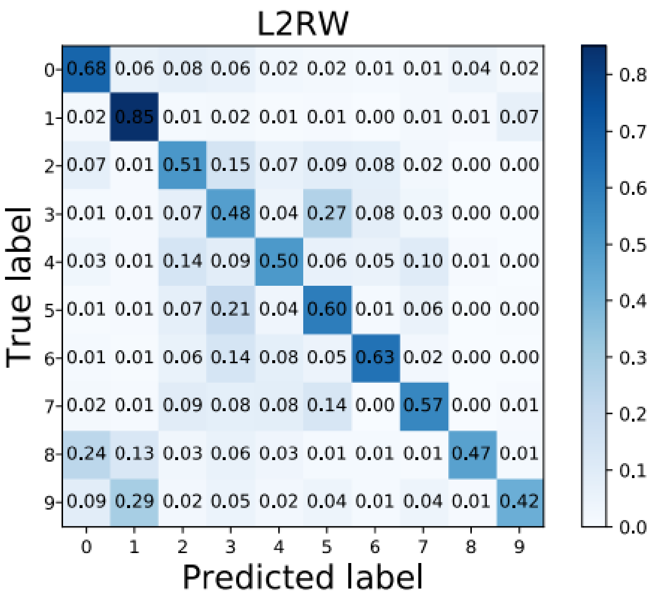
- ❑ Rebalancing the loss
  Assign larger/smaller weight to tail/head classes.
  e.g., CB-Focal[1], LDAM[2]

[1] Cui, Yin, et al. "Class-balanced loss based on effective number of samples." CVPR. 2019.
[2] Cao, Kaidi, et al. "Learning imbalanced datasets with label-distribution-aware margin loss." NIPS. 2019.

Introduction

Current methods overly fit the dominant classes and fail in the underrepresented tail classes as they implicitly assume that the test sets are drawn i.i.d. from the same underlying distribution as the long-tailed training set





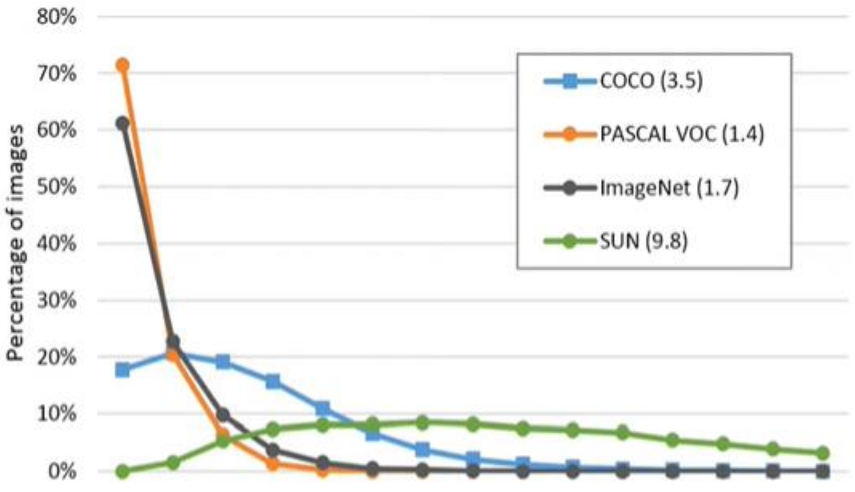
Accuracy on head classes ↑
Accuracy on tail classes ↓

Accuracy on head classes ↓
Accuracy on tail classes ↑

Introduction

Object: training a model to uncover the assumption that the training and test set share the same class-conditioned distribution.

Datasets: iNaturalist, LVIS, ImageNet, COCO, etc.



Theoretical analysis

New perspective: Domain adaptation

Source domain (with labeled data, training set)

$$Ds = \{(x_m, y_m)\} \sim \boxed{P_s(X,Y)}$$

Target domain(no labels for training, test set)

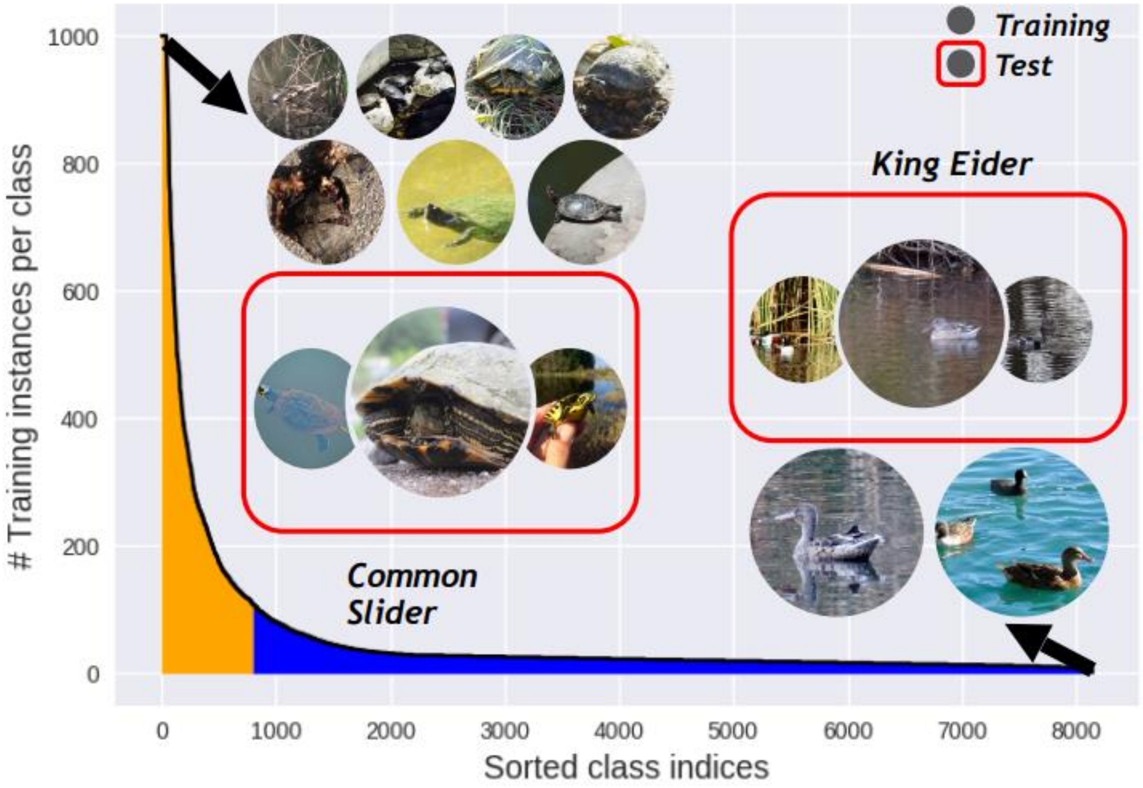
$$D_T = \{(x_n, ?)\} \sim \boxed{P_t(X,Y)}$$

Different distributions

Object: Learn model to work well on target domain.

Theoretical analysis

For current works:



Assumption: target shift.

$$P_s(X, Common\ Slider) = P_t(X, Common\ Slider)$$

$$P_s(X, King\ Eider) = P_t(X, King\ Edier)$$

Reasonable!

Wrong!!

But, in truth:

$$P_s(X, Common\ Slider) = P_t(X, Common\ Slider)$$

$$P_s(X, King\ Eider) \neq P_t(X, King\ Edier)$$

Theoretical analysis

Two-component approach: accounting for $P_s(X, King\ Eider) \neq P_t(X, King\ Edier)$

$$\begin{aligned}
\texttt{error} &= \mathbb{E}_{P_t(x,y)} L(f(x;\theta), y) \\
&= \mathbb{E}_{P_s(x,y)} L(f(x;\theta), y) P_t(x,y)/P_s(x,y) \\
&= \mathbb{E}_{P_s(x,y)} L(f(x;\theta), y) \frac{P_t(y) P_t(x|y)}{P_s(y) P_s(x|y)} \\
&:= \mathbb{E}_{P_s(x,y)} L(f(x;\theta), y) w_y (1 + \tilde{\epsilon}_{x,y})
\end{aligned}$$

Where $w_y = P_t(y)/\ P_s(y), \hat{\varepsilon}_{x,y} = \frac{P_t(x|y)}{P_s(x|y)} - 1$

For specifically, $w_y = (1-\beta)/(1-\beta^n)$, $\beta = \frac{n-1}{n}$, n is the effective number of samples

$$\begin{aligned}
\texttt{error} &= \mathbb{E}_{P_s(x,y)} L(f(x;\theta), y)(w_y + \epsilon_{x,y}) \\
&\approx \frac{1}{n} \sum_{i=1}^{n} \boxed{(w_{y_i} + \epsilon_i)} L(f(x_i;\theta), y_i),
\end{aligned}$$

Two components

Unbiased loss function!

Theoretical analysis

Learn the two components:

Estimating the class-wise weights w_y :
Supposing there are n_y training examples for the $y-th$ class

Then:

$$w_y \approx (1-\beta)/(1-\beta^{n_y}),$$

$$\beta = (n-1)/n$$

n is the number of training examples.

e.g.in CIFAR10-LT w_y = [0.2923, 0.3583, 0.4441, 0.5551, 0.6995, 0.8860, 1.1265, 1.4328, 1.8409, 2.3588]

Theoretical analysis

Meta-learning the condition weights $\{\varepsilon\}$

$$\min_{\boldsymbol{\epsilon}} \quad \frac{1}{|D|}\sum_{i\in D} L(f(x_i;\theta^*(\boldsymbol{\epsilon})), y_i) \text{ with}$$

$$\theta^*(\boldsymbol{\epsilon}) \leftarrow \arg\min_{\theta} \frac{1}{|T|}\sum_{i\in T}(w_{y_i}+\epsilon_i)L(f(x_i;\theta), y_i)$$

D is the balanced development dataset from training set.

Here is the solution for above problem. (Modified meta-learning framework)

$$\tilde{\theta}^{t+1}(\boldsymbol{\epsilon}^t) \leftarrow \theta^t - \eta \frac{\partial \sum_{i\in T}(w_{y_i}+\epsilon_i^t)L(f(x_i;\theta^t), y_i)}{\partial \theta}$$

$$\boldsymbol{\epsilon}^{t+1} \leftarrow \boldsymbol{\epsilon}^t - \tau \frac{\partial \sum_{i\in D} L(f(x_i;\tilde{\theta}^{t+1}(\boldsymbol{\epsilon}^t)), y_i)}{\partial \boldsymbol{\epsilon}}$$

$$\theta^{t+1} \leftarrow \theta^t - \eta \frac{\partial \sum_{i\in T}(w_{y_i}+\epsilon_i^{t+1})L(f(x_i;\theta^t), y_i)}{\partial \theta}$$

Theoretical analysis

Differences between this method and L2RW

	L2RW	Ours
Pre-training	×	√
Clip negative ε	√	×
Normalization ε	√	×
Free space of ε	reduced	large

Theoretical analysis

Overall algorithm for Long-tailed recognition

Algorithm 1 Meta-learning for long-tailed recognition

Require: Training set T, balanced development set D
Require: Class-wise weights $\{w_y\}$ estimated by using

$$w_y = (1-\beta)/(1-\beta^n)$$

Require: Learning rates η and τ, stopping steps t_1 and t_2
Require: Initial parameters θ of the recognition network

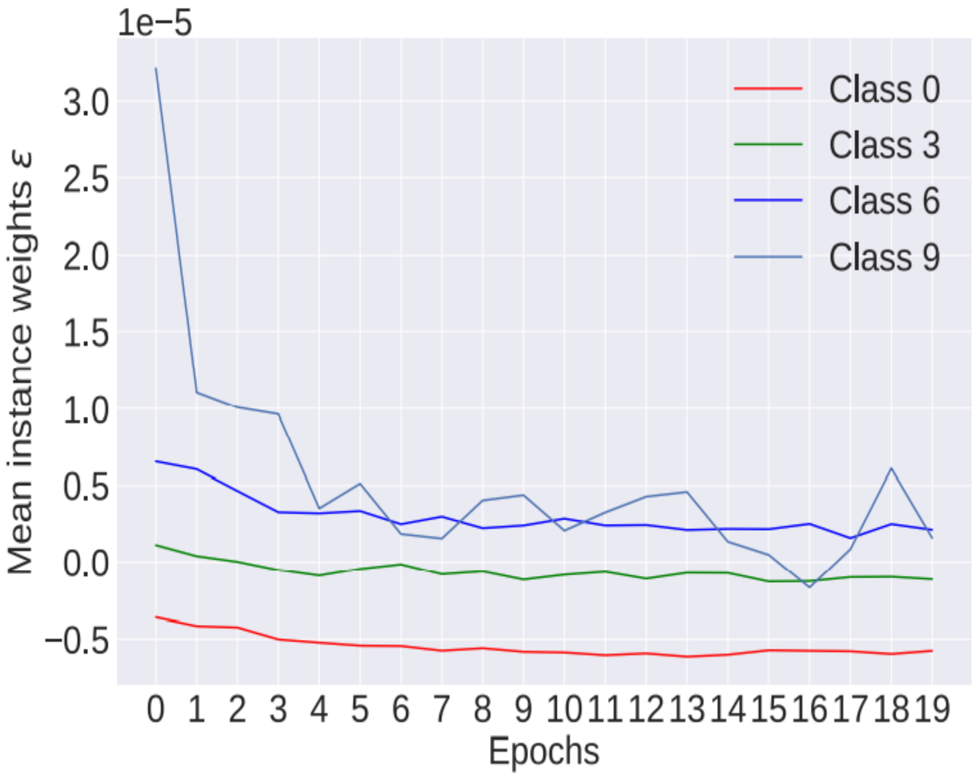
for $t = 1, 2, \cdots, t_1$ **do**
  Sample a mini-batch B from the training set T
  Compute loss $\mathcal{L}_B = \frac{1}{|B|}\sum_{i \in B} L(f(x_i;\theta), y_i)$
  Update $\theta \leftarrow \theta - \eta\nabla_\theta \mathcal{L}_B$
end for

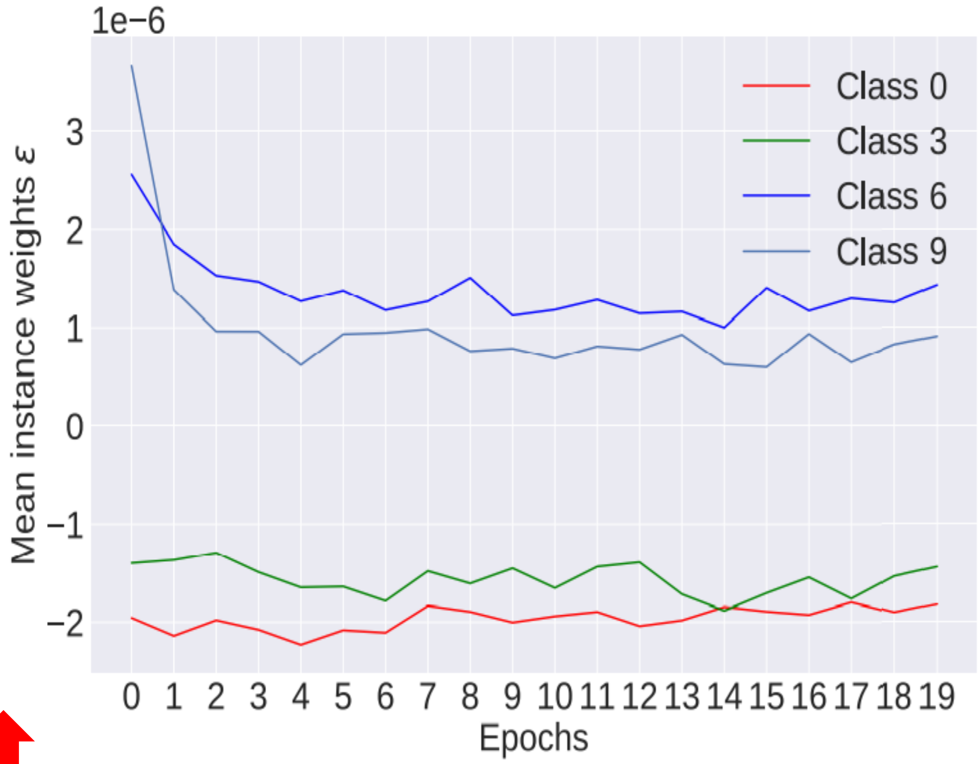
ε: small part of weigth

for $t = t_1 + 1, \cdots, t_1 + t_2$ **do**
  Sample a mini-batch B from the training set T
  Set $\epsilon_i \leftarrow 0, \forall i \in B$, and denote by $\boldsymbol{\epsilon} := \{\epsilon_i, i \in B\}$
  Compute $\mathcal{L}_B = \frac{1}{|B|}\sum_{i \in B}(w_{y_i} + \epsilon_i) L(f(x_i;\theta), y_i)$
  Update $\tilde{\theta}(\boldsymbol{\epsilon}) \leftarrow \theta - \eta\nabla_\theta \mathcal{L}_B$
  Sample B_d from the balanced development set D
  Compute $\mathcal{L}_{B_d} = \frac{1}{|B_d|}\sum_{i \in B_d} L(f(x_i;\tilde{\theta}(\boldsymbol{\epsilon})), y_i)$
  Update $\boldsymbol{\epsilon} \leftarrow \boldsymbol{\epsilon} - \tau\nabla_{\boldsymbol{\epsilon}} \mathcal{L}_{B_d}$
  Compute new loss with the updated $\boldsymbol{\epsilon}$
  $\tilde{\mathcal{L}}_B = \frac{1}{|B|}\sum_{i \in B}(w_{y_i} + \epsilon_i) L(f(x_i;\theta), y_i)$
  Update $\theta \leftarrow \theta - \eta\nabla_\theta \tilde{\mathcal{L}}_B$
end for

Theoretical analysis

What are the learned ε





Left: imbalanced factor: 100; right: IF:10

Experiments

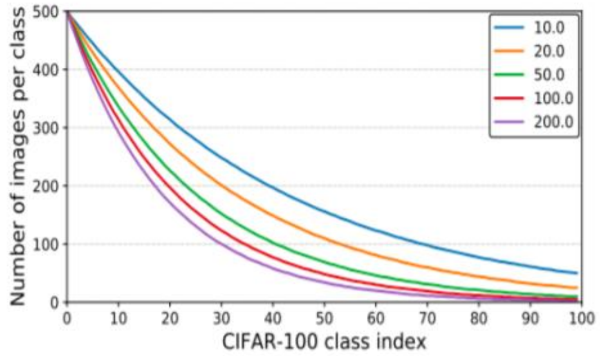
Datasets:

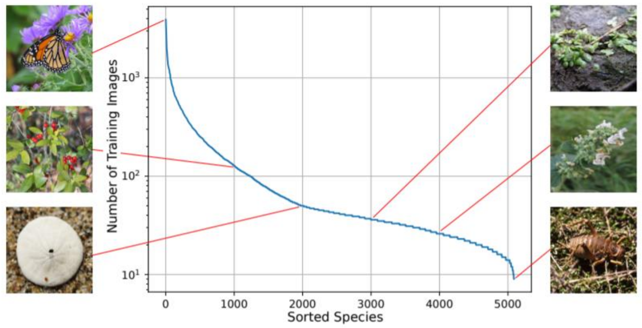
I. CIFAR-LT-10(100)
  Constructed from CIFAR10(100)
  10(100) categories, 50,000-11,203(50,000-9,502) images
  Ten images per class as development set D

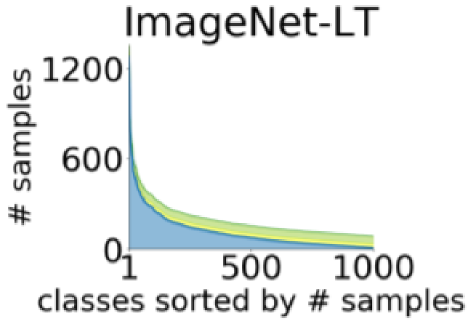
II. iNaturalist 2017(2018)
  Contains only species
  5,089(8,142) categories, 579,184(435,713) images
  Five(two) images per class as D

III. ImageNet-LT
  Constructed from ImageNet 2012
  1,000 categories, 115.8k images
  20 images per class as D

IV. Places-LT
  Constructed from Places 365
  365 classes, 62.5k images
  Ten images per class as D







Experiments

Loss Functions{$L(\ )$}:

I. Cross-Entropy

$$\vec{\mathcal{L}}_B = -\frac{1}{|B|}\sum_{i \in B} y_i * log f(x_i, \theta)$$

II. Class balanced loss

$$\vec{\mathcal{L}}_B = -\frac{1}{|B|}\sum_{i \in B} w_y * L(f(x_i, \theta))$$

III. Focal loss

$$\vec{\mathcal{L}}_B = -\frac{1}{|B|}\sum_{i \in B} y_i * \alpha_t \left(1 - f(x_i, \theta)\right)^{\gamma} log f(x_i, \theta)$$

IV. Lable-distribution-aware margin loss

$$\mathcal{L}_{\mathrm{LDAM}}((x, y); f) = -\log \frac{e^{z_y - \Delta_y}}{e^{z_y - \Delta_y} + \sum_{j \neq y} e^{z_j}}$$

$$\text{where } \Delta_j = \frac{C}{n_j^{1/4}} \text{ for } j \in \{1, \dots, k\}$$

Experiments

Implementation details

*Neural network: Resnet32
training: learning rate = 0.1((0.01 ** int(epoch >= 160)) * (0.01 ** int(epoch >= 180));
200 epochs.
batch size: 100
(train all models on a single GPU using the stochastic gradient descent with momentum)

Experiments

CIFAR-LT -10-classification error:

Imbalance factor	200	100	50	20	10	1
Cross-entropy training	34.32	29.64	25.19	17.77	13.61	7.53/7.11*
Class-balanced cross-entropy loss [7]	31.11	27.63	21.95	15.64	13.23	7.53/7.11*
Class-balanced fine-tuning [8]	33.76	28.66	22.56	16.78	16.83	7.08
Class-balanced fine-tuning [8]*	33.92	28.67	22.58	13.73	13.58	**6.77**
L2RW [43]	33.75	27.77	23.55	18.65	17.88	11.60
L2RW [43]*	33.49	25.84	21.07	16.90	14.81	10.75
Meta-weight net [47]	32.8	26.43	20.9	15.55	12.45	7.19
Ours with cross-entropy loss	**29.34**	**23.59**	**19.49**	**13.54**	**11.15**	7.21
Focal loss [34]	34.71	29.62	23.29	17.24	13.34	6.97
Class-balanced focal Loss [7]	31.85	25.43	20.78	16.22	12.52	6.97
Ours with focal Loss	**25.57**	**21.1**	**17.12**	**13.9**	**11.63**	7.19
LDAM loss [4] (results reported in paper)	-	26.65	-	-	13.04	11.37
LDAM-DRW [4] (results reported in paper)	-	22.97	-	-	**11.84**	-
Ours with LDAM loss	**22.77**	**20.0**	**17.77**	**15.63**	12.6	**10.29**

$$\frac{\text{The number of Samples from Largest class}}{\text{The number of samples from smallest class}}$$

the number of examples dropped from the $y-th$ class is $n_y\mu^y$

Experiments

CIFAR-LT-100-classification error

Imbalance factor	200	100	50	20	10	1
Cross-entropy training	65.16	61.68	56.15	48.86	44.29	29.50
Class-balanced cross-entropy loss [7]	64.30	61.44	55.45	48.47	42.88	29.50
Class-balanced fine-tuning [8]	61.34	58.5	53.78	47.70	42.43	29.37
Class-balanced fine-tuning [8]*	61.78	58.17	53.60	47.89	42.56	29.28
L2RW [43]	67.00	61.10	56.83	49.25	47.88	36.42
L2RW [43]*	66.62	59.77	55.56	48.36	46.27	35.89
Meta-weight net [47]	63.38	58.39	54.34	46.96	41.09	29.9
Ours with cross-entropy loss	**60.69**	**56.65**	**51.47**	**44.38**	**40.42**	**28.14**
Focal Loss [34]	64.38	61.59	55.68	48.05	44.22	**28.85**
Class-balanced focal Loss [7]	63.77	60.40	54.79	47.41	42.01	**28.85**
Ours with focal loss	**60.66**	**55.3**	**49.92**	**44.27**	**40.41**	29.15
LDAM Loss [4] (results reported in paper)	-	60.40	-	-	43.09	-
LDAM-DRW [4] (results reported in paper)	-	57.96	-	-	**41.29**	-
Ours with LDAM loss	**60.47**	**55.92**	**50.84**	**47.62**	42.0	-

Experiments

iNat2017 and 2018 classification error:

Dataset	iNat 2017		iNat 2018	
Method	Top-1	Top-3/5	Top-1	Top-3/5
CE	43.49	26.60/21.00	36.20	19.40/15.85
CB CE [7]	42.59	25.92/20.60	34.69	19.22/15.83
Ours, CE	**40.62**	**23.70/18.40**	**32.45**	**18.02/13.83**
CB focal [7]*	41.92	–/20.92	38.88	–/18.97
LDAM [4]*	–	–	35.42	–/16.48
LDAM-drw*	–	–	32.00	–/14.82
cRT [30]*	–	–	34.8	–
cRT+epochs*	–	–	32.4	–

CB: class balanced, CE: cross-entropy

Experiments

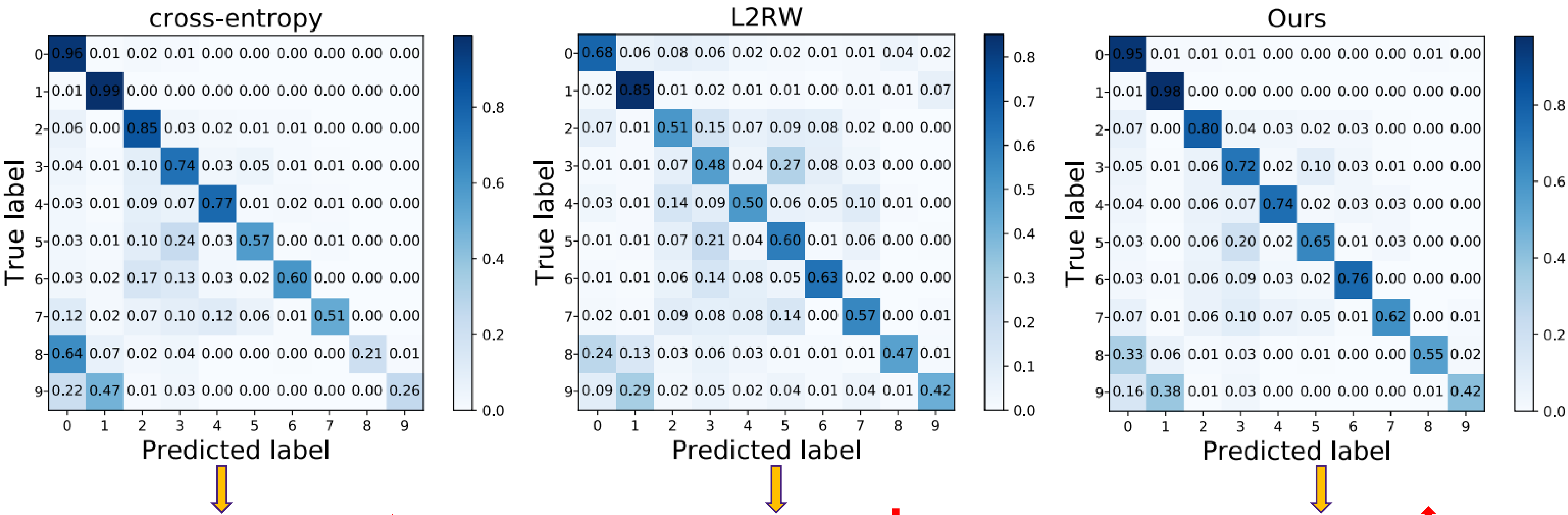
ImageNet-LT, Places-LT classification error:

Dataset	ImageNet-LT		Places-LT	
Method	Top-1	Top-3/5	Top-1	Top-3/5
CE	74.74	61.35/52.12	73.00	52.05/41.44
CB CE [7]	73.41	59.22/50.49	71.14	51.58/41.96
Ours, CE	**70.10**	**53.29/45.18**	**69.20**	**47.95/38.00**

Details: ImageNet: Resnet32; learn rate = 0.1**(epochs|35)
Places: Resnet152; learn rate = 0.01*0.1**(epochs|10)

Experiments

This method V.S. current method (CIFAR10, IF=200)



Accuracy on head classes

Accuracy on tail classes

Accuracy on head classes

Accuracy on tail classes

Accuracy on head classes

Accuracy on tail classes

Conclusion

There are two major contributions to the long-tailed visual recognition.

One is the novel domain adaptation perspective for analyzing the mismatch problem in long-tailed classification.

The second contribution is to relax this assumption to explicitly model the ratio between two class conditioned distributions.

Thanks!