南京航空航天大學
Nanjing University of Aeronautics and Astronautics

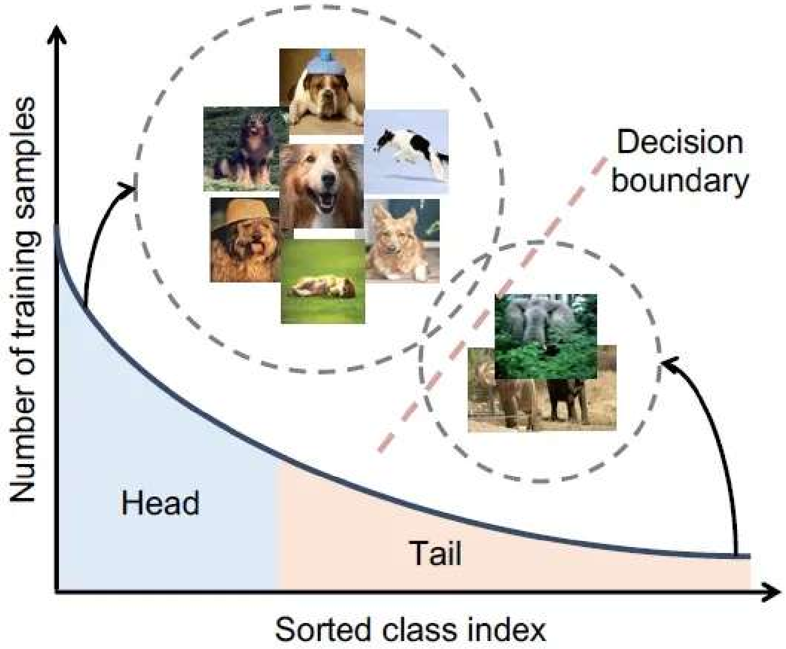
Long-tailed Visual Recognition via Gaussian Clouded Logit Adjustment

Mengke Li[1] Yiu-ming Cheung[1*] Yang Lu[2]
[1]Department of Computer Science, Hong Kong Baptist University, Hong Kong
[2]Department of Computer Science and Technology, School of Informatics, Xiamen University, China
{csmkli, ymc}@comp.hkbu.edu.hk, luyang@xmu.edu.cn

CVPR 2022

南京航空航天大學
Nanjing University of Aeronautics and Astronautics

Long-tailed Distribution



- the distorted embedding space
- the biased classifier

- Resampling
- Reweighting
- Logit-adjustment

$$\ell(y, f(x)) = -\log \frac{e^{f_y(x)+\tau\cdot\log\pi_y}}{\sum_{y'\in[L]} e^{f_{y'}(x)+\tau\cdot\log\pi_{y'}}}$$

$\pi \in \Delta_y$ are estimates of the class priors $\mathbb{P}(y)$.

南京航空航天大學
Nanjing University of Aeronautics and Astronautics

Long-tailed Visual Recognition via Gaussian Clouded Logit Adjustment

Mengke Li[1] Yiu-ming Cheung[1*] Yang Lu[2]

[1]Department of Computer Science, Hong Kong Baptist University, Hong Kong

[2]Department of Computer Science and Technology, School of Informatics, Xiamen University, China

{csmkli, ymc}@comp.hkbu.edu.hk, luyang@xmu.edu.cn

CVPR 2022

GCL

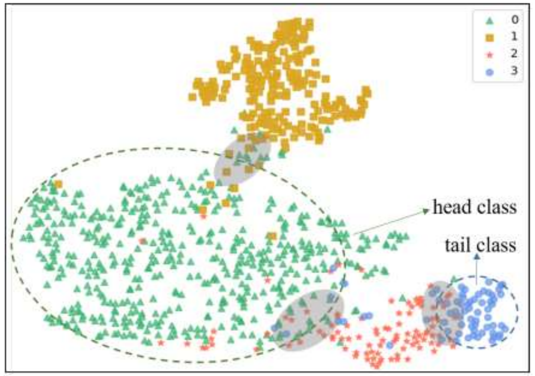


Figure 1. t-SNE visualization of the distorted embedding space. (Color for the best view.) The embeddings are calculated with ResNet-32 on a subset with four classes of CIFAR-10-LT. We randomly select four classes with the training numbers 500, 200, 100, and 50, respectively. The gray areas show the obscure regions between different classes.

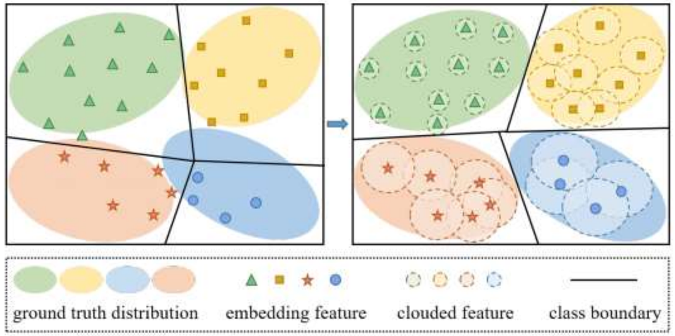


Figure 2. An overview of GCL. (Color for the best view.) The tail class logit is assigned to a larger sample cloud size than the head class, which corresponds to a large relative cloud size of the feature in the direction of the tail class anchor. In this way, the distortion of the embedding space can be calibrated well.

Suppose $\{x, y\} \in \mathcal{T}$ represents a sample $\{x, y\}$ from the training set $\mathcal{T}$ with the total N samples in C classes, and $y \in \{1, \ldots, C\}$ is the ground truth label. The softmax loss function for the input image x can be written as:

$$\mathcal{L}(x) = -\log p_y, \text{ with } p_y = \frac{e^{z_y}}{\sum_{j=1}^{C} e^{z_j}}, \quad (1)$$

$$\frac{\partial \mathcal{L}}{\partial z_j} = \begin{cases} p_j - 1, & j = y \\ p_j, & j \neq y. \end{cases} \quad (2) \quad \Rightarrow \quad \frac{\partial \mathcal{L}}{\partial z_1} = -\frac{1}{1 + e^{z_1 - z_2}}. \quad (3) \quad \Rightarrow$$

approaches zero with the increase of the logit difference

$$z_j^{cld} = s \cdot (\tilde{z}_j + \frac{\delta}{s_1}\varepsilon_j) \qquad \delta_j = \log n_{max} - \log n_j.$$

GCL——Proof of formula

Suppose the features of different class samples satisfy Gaussian distribution. We can obtain a disturbed feature $\mathbf{f}^{cld}$ of the input by Gaussian sampling, which is represented as:

$$\mathbf{f}^{cld} \triangleq \mathbf{f} + \delta\mathbf{E} \qquad \mathbf{E} \sim \mathcal{N}(\mathbf{u}, \mathbf{\Sigma}) \quad \delta > 0 \tag{4}$$

$$\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_C\} \in \mathbb{R}^{D\times C} \tag{5}$$

$$\begin{aligned} z_j^{cld} &= \mathbf{w}_j^T \mathbf{f}^{cld} + \mathbf{b}_j \\ &= \mathbf{w}_j^T \mathbf{f} + \mathbf{b}_j + \mathbf{w}_j^T(\delta\mathbf{E}) \\ &= z_j + \delta(\mathbf{w}_j^T \mathbf{E}). \end{aligned} \tag{6}$$

clouded term

$$\begin{aligned} z_j^{cld} &= \frac{s_1 \mathbf{w}_j^T \cdot s_2 \mathbf{f}^{cld}}{\|\mathbf{w}_j^T\| \|\mathbf{f}^{cld}\|} \\ &= s \cdot \left(\frac{\mathbf{w}_j^T \mathbf{f}}{\|\mathbf{w}_j^T\| \|\mathbf{f} + \delta\mathbf{E}\|} + \delta \frac{\mathbf{w}_j^T \mathbf{E}}{\|\mathbf{w}_j^T\| \|\mathbf{f} + \delta\mathbf{E}\|}\right) \end{aligned} \tag{7}$$

I_j =

where $s = s_1 \cdot s_2$ is a constant.

$\approx \|\mathbf{f}\|$

$= s_1$

$$z_j^{cld} \approx s \cdot \left(\frac{\mathbf{w}_j^T \mathbf{f}}{\|\mathbf{w}_j^T\| \|\mathbf{f}\|} + \frac{\delta}{s_1} I_j \mathbf{E}\right), \tag{8}$$

$$\begin{aligned} z_j^{cld} &= s \cdot (\tilde{z}_j + \tfrac{\delta}{s_1}\varepsilon_j) \\ &\Leftrightarrow s \cdot (\tilde{z}_j + \delta_j \varepsilon) \end{aligned} \tag{9}$$

$$\delta_j = \log n_{max} - \log n_j$$

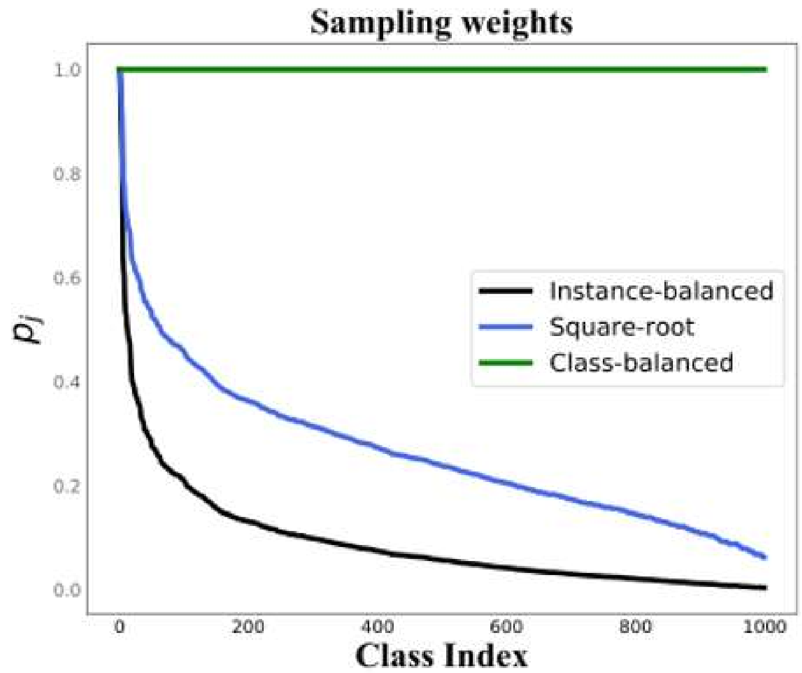
The Gaussian clouded logit difference $\Delta_{y\text{-}j}$ between the target and non-target classes is:

$$\begin{aligned} \Delta_{y\text{-}j} &= z_y^{cld} - z_j^{cld} \\ &= z_y - z_j + \varepsilon(\delta_y - \delta_j) \end{aligned}$$

$$z_j^{cld} = s \cdot (\tilde{z}_j - \delta_j \|\varepsilon\|)$$

$$\mathcal{L}_{GCL} = -\frac{1}{N} \sum_i \log \frac{e^{z_{y_i}^{cld}}}{\sum_j e^{z_j^{cld}}} \tag{10}$$

CBEN——Resampling



$$p_j = \frac{n_j^q}{\sum_{i=1}^{C} n_i^q} \begin{cases} q = 0 \\ q = 1/2 \\ q = 1 \end{cases}$$

The sampling probability ρ_j of a sample from class j is calculated by:

$$\rho_j = \frac{1-\beta_j}{1-\beta_j^{n_j}} \quad (11) \qquad \beta_j = b \times \frac{\delta_j - \delta_{min}}{\delta_{max} - \delta_{min}} + a \quad (12)$$

$$\delta_j = \log n_{max} - \log n_j$$

$$\rho_j \leftarrow \frac{\rho_j}{\sum_i \rho_i} \quad (13)$$

Algorithm Flow

Algorithm 1: Gaussian clouded logit

Input: Training dataset $\mathcal{T}$;
Output: Predicted labels;
Initialize the model parameters ω of the CNN network $\phi((x, y); \omega)$ randomly ;
for $iter = 1$ *to* I_0 **do**
  Sample a batch samples $\mathcal{B}$ from the original long-tailed data $\mathcal{T}$ with batch size b;
  Obtain the logit cloud size:
  $\delta_j \leftarrow \log n_{max} - \log n_j$;
  Calculate the loss by Eq. (12):
  $\mathcal{L}((x, y); \omega) = \frac{1}{b} \sum_{(x,y)\in\mathcal{B}} L_{GCL}(x, y)$;
  Update model parameters:
  $\omega = \omega - \alpha \nabla_{\omega} \mathcal{L}((x, y); \omega)$.
end
for $iter = I_0 + 1$ *to* $I_0 + I_1$ **do**
  Calculate sampling rate:
  $\beta_j \leftarrow b \times \frac{\delta_j - \delta_{max}}{\delta_{max} - \delta_{min}} + a$; $\rho_j \leftarrow \frac{1-\beta_j}{1-\beta_j^{n_j}}$;
  $\rho_j \leftarrow \frac{\rho_j}{\sum_i \rho_i}$;
  Sample a batch samples $\mathcal{B}'$ with the sampling probability ρ_j and the batch size b;
  Calculate the loss by Eq. (12):
  $\mathcal{L}((x, y); \omega) = \frac{1}{b} \sum_{(x,y)\in\mathcal{B}'} L_{GCL}(x, y)$;
  Update classifier parameters ω_{cls} (representation parameters are frozen):
  $\omega_{cls} = \omega_{cls} - \alpha \nabla_{\omega_{cls}} \mathcal{L}((x, y); \omega_{cls})$.
end

$$\delta_i \triangleq \delta_i / \delta_{max}$$

$$\mathcal{L}_{GCL} = -\frac{1}{N} \sum_i \log \frac{e^{\tilde{z}^{cld}_{y_i}}}{\sum_j e^{\tilde{z}^{cld}_j}}$$

$\beta_j \in [0.999, 0.9999]$, i.e. $a = 0.999$ and $b = 0.0009$.

Experiment——GCL

Table 1. Comparison results on CIFAR-10/100-LT in terms of top-1 accuracy (%), where the best and the second-best results are shown in **underline bold** and **bold**, respectively. *indicates that the results are quoted from the corresponding references. The other results are obtained by re-implementing with the official codes.

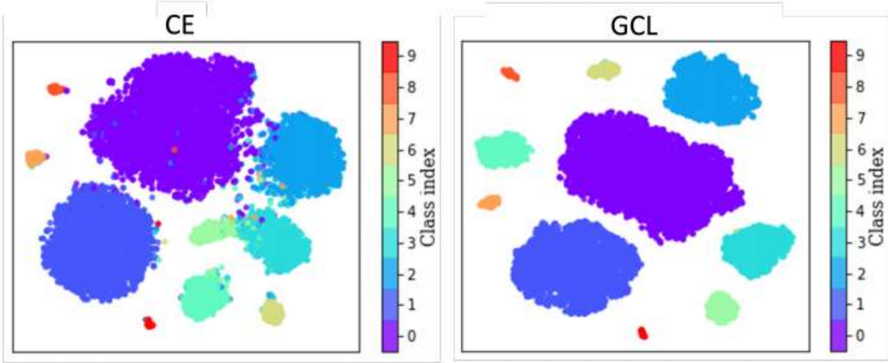
Dataset	CIFAR-10-LT			CIFAR-100-LT		
Backbone Net	ResNet-32					
Imbalance ratio	200	100	50	200	100	50
CE loss	65.68	70.70	74.81	34.84	38.43	43.9
CE loss + mixup [33] (2018)	65.84	72.96	79.48	35.84	40.01	45.16
LDAM-DRW [2] (2019)	73.52	77.03	81.03	38.91	42.04	47.62
De-confound-TDE * [24] (2020)	-	80.60	83.60	-	44.15	50.31
CE loss + mixup + cRT [10] (2020)	73.06	79.15	84.21	41.73	45.12	50.86
BBN [40] (2020)	73.47	79.82	81.18	37.21	42.56	47.02
Contrastive learning * [30] (2021)	-	81.40	**85.36**	-	46.72	51.87
MisLAS [39] (2021)	**77.31**	**82.06**	85.16	**42.33**	**47.50**	**52.62**
GCL	**<u>79.03</u>**	**<u>82.68</u>**	**<u>85.46</u>**	**<u>44.88</u>**	**<u>48.71</u>**	**<u>53.55</u>**

Experiment——GCL

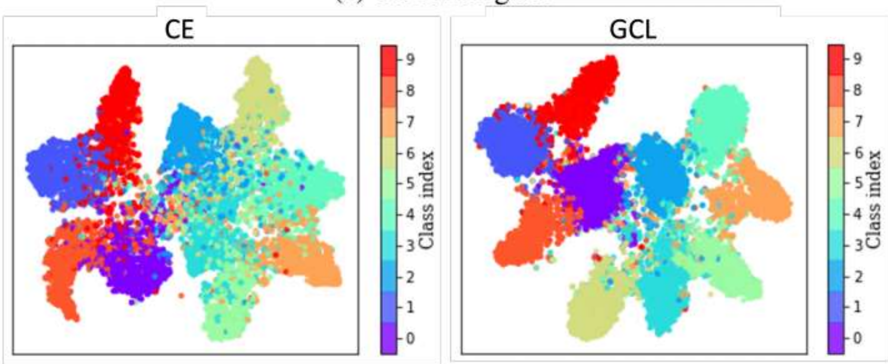
Table 2. Comparison results on ImageNet-LT, iNaturalist 2018 and Places-LT in terms of top-1 accuracy (%), where the best and the second-best results are shown in **underline bold** and **bold**, respectively. *indicates that the results are quoted from the corresponding references. The other results are obtained by re-implementing with the official codes.

Dataset	ImageNet-LT	iNaturalist 2018	Places-LT
Backbone Net	ResNet-50	ResNet-50	ResNet-152
CE loss	44.51	63.80	27.13
CE loss + mixup [33] (2018)	45.66	65.77	29.51
LDAM-DRW [2] * (2019)	48.80	68.00	-
OLTR * [16] (2019)	-	-	35.9
Decoupling [10] (2020)	47.70	69.49	37.62
CE loss + mixup + cRT [10] (2020)	51.68	70.16	38.51
Logit adjustment * [17](2021)	51.11	66.36	-
DisAlign * [35] (2021)	**52.91**	70.06	39.30
MisLAS [39] (2021)	52.11	**71.57**	**40.15**
GCL	<u>**54.88**</u>	<u>**72.01**</u>	<u>**40.64**</u>

Experiment——CE vs GCL



(a) On training set



(b) On test set

Figure 3. Visualization of the embedding via t-SNE from CIFAR-10-LT with $\gamma = 100$, where backbone network is ResNet-32. (Color for the best view.)

Experiment——GCL

Table 3. Ablation experiment of different cloud size adjustment strategies (AS) on CIFAR-10-LT with $\gamma = 100$.

AS	Expression	Acc.(%)
cos.	$\cos(n_j/n_{max} \cdot \pi/2)$	79.21
pow. diff. (e:1/3)	$n_{max}^{1/3} - n_j^{1/3}$	80.80
pow. diff. (e:1/4)	$n_{max}^{1/4} - n_j^{1/4}$	82.31
log. diff.	$\log n_{max} - \log n_j$	**82.68**

Table 4. Ablation experiment of different re-sampling strategy on CIFAR-10-LT with $\gamma = 100$.

Sam.	RT	Acc.(%)
IB	cRT	80.41
CB	cRT	82.43
EN	cRT	82.47
CBEN	cRT	**82.68**

Table 5. Ablation experiment of different re-training strategies on CIFAR-10-LT with $\gamma = 100$.

Sam.	RT	Acc.(%)
-	w/o RT	80.52
CBEN	LWS	82.25
CBEN	τ-nor.	82.16
CBEN	cRT	**82.68**

$$\tilde{z}_j^{cld} = s \cdot (\tilde{z}_j - \delta_j \|\varepsilon\|)$$

Conclusion

作者发现SoftMax饱和对头部类样本和尾部类样本的有效性有着不同程度的影响。故可以利用它来自动调整不同类别的训练样本有效性——提出了GCL：将尾类logits设置为较大的云大小，鼓励更多的有效的尾类样本参与训练，有利于获得均匀分布的嵌入空间。

通过GCL，不同类的有效性是不同的。在此基础上，提出了一种简单有效的CBEN采样策略，结合cRT进行分类器平衡，进一步提高了模型的性能。

Thanks