Logit Perturbation

Mengyang Li, Fengguang Su, Ou Wu*, Ji Zhang

Jiuantianxia Inc.
National Center for Applied Mathematics, Tianjin University
The University of Southern Queensland

November 23, 2023

Overview

1. Research Background
2. Related Work
3. Our Method LPL

Research Background

- New Network Architecture.
- New Training Loss.
- New Learning Strategy.
- New Training Data Perturbation Scheme [Feature, Label, Logit].

LDAM (Cao et al. 2019)

- LDAM is designed for long-tail classification:

$$\mathcal{L} = -\sum_{i=1}^{N} \log \frac{\exp(u_{i,y_i} - C(\pi_{y_i})^{-1/4})}{\exp(u_{i,y_i} - C(\pi_{y_i})^{-\frac{1}{4}}) + \sum_{c \neq y_i} \exp(u_{i,c})}. \tag{1}$$

- Logit perturbation:

$$\delta_i = \tilde{\delta}_{y_i} = \lambda[0, \cdots, -C(\pi_{y_i})^{-\frac{1}{4}}, \cdots, 0]^T. \tag{2}$$

- The losses for all categories are increased.

LA (Wang et al. 2019)

- LA achieves a competitive performance on long-tail classification:

$$\mathcal{L} = \sum_i l(\text{softmax}(u_i + \delta_i), y_i) = -\sum_i \log \frac{\exp(u_{i,y_i} + \lambda \log \pi_{y_i})}{\sum_c \exp(u_{i,c} + \lambda \log \pi_c)}. \tag{3}$$

- Logit perturbation:

$$\delta_i = \tilde{\delta} = \lambda[\log\pi_1, \cdots, \log\pi_c, \cdots, \log \pi_C]^T. \tag{4}$$

- The losses of the samples in the first category (head) are decreased.
- The losses of the samples in the last category (tail) are increased.

ISDA (Menon et al. 2021)

- ISDA assumes that each (virtual) new sample can be sampled from a distribution $\mathcal{N}(x_i, \Sigma_{y_i})$, when (virtual) #samples $\to +\infty$, the upper bound of the loss becomes:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(u_{i,y_i})}{\sum_{c=1}^{C}\exp(u_{i,c}+\frac{\lambda}{2}(\mathrm{w}_c-\mathrm{w}_{y_i})^T\Sigma_{y_i}(\mathrm{w}_c-\mathrm{w}_{y_i}))}. \tag{5}$$

- Logit perturbation:

$$\delta_i = \tilde{\delta}_{y_i} = \frac{\lambda}{2}\begin{bmatrix}(\mathrm{w}_1-\mathrm{w}_{y_i})^T\Sigma_{y_i}(\mathrm{w}_1-\mathrm{w}_{y_i})\\ \vdots \\ (\mathrm{w}_C-\mathrm{w}_{y_i})^T\Sigma_{y_i}(\mathrm{w}_C-\mathrm{w}_{y_i})\end{bmatrix}. \tag{6}$$

- The losses for all categories are increased.

Analysis

The losses of the three example methods analyzed can be written as follows:

$$\mathcal{L} = \sum_i l(\text{softmax}(u_i + \tilde{\delta}_{y_i}), y_i). \tag{7}$$

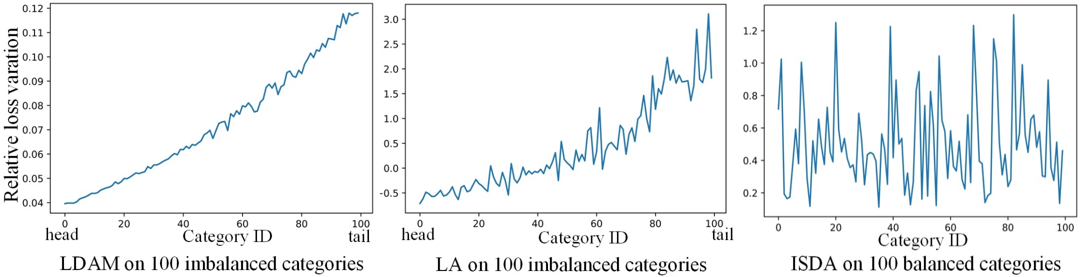

Figure: The relative loss variations ($\frac{l'-l}{l}$) of the three methods on different categories.

Motivation

Conjectures

- If one aims to positively augment the samples in a category, the loss of this category should be increased. The larger the loss increment, the greater the augmentation.
- If one aims to negatively augment the samples in a category, then the loss of this category should be reduced. The larger the loss decrement, the greater the negative augmentation.

Our Method LPL

$$\mathcal{L}=\sum_{c\in\mathcal{N}_a}\sum_{x_i\in S_c}\min_{\|\tilde{\delta}_c\|\le\epsilon_c} l(\mathrm{softmax}(u_i+\tilde{\delta}_c),c)+\sum_{c\in\mathcal{P}_a}\sum_{x_i\in S_c}\max_{\|\tilde{\delta}_c\|\le\epsilon_c} l(\mathrm{softmax}(u_i+\tilde{\delta}_c),c). \quad (8)$$

Overview of the LPL

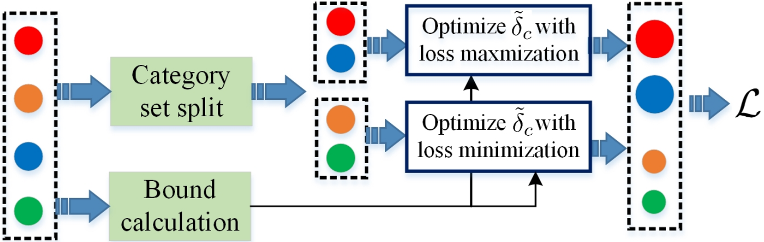

Figure: Four solid circles denote four categories. Two categories are positively augmented via loss maximization and the rest two are negatively augmented via minimization.

Category Set Split

Balanced classification

$$\mathcal{L} = \sum\nolimits_c \{\mathbb{S}(\tau - \bar{q}_c) \times \sum_{x_i \in S_c} \max_{\|\tilde{\delta}_c\| \leq \epsilon_c} [l(\text{softmax}(u_i + \tilde{\delta}_c), c)\mathbb{S}(\tau - \bar{q}_c)]\}. \tag{9}$$

Long-tail classification

$$\mathcal{L} = \sum\nolimits_c \{\mathbb{S}(c - \tau) \times \sum_{x_i \in S_c} \max_{\|\tilde{\delta}_c\| \leq \epsilon_c} [l(\text{softmax}(u_i + \tilde{\delta}_c), c)\mathbb{S}(c - \tau)]\}. \tag{10}$$

PGD-like Optimize

Eqs. (9) and (10) can be solved with an optimization approach similar to PGD. According to the derivative of the cross-entropy loss function with respect to logit vector, our PGD-like optimization method can be implemented simply.

- In the maximization of Eqs. (9) and (10), $\tilde{\delta}_{y_i}$ is updated by

$$\tilde{\delta}_{y_i} = \frac{\lambda}{N_{y_i}} \sum_{j:y_j=y_i} (\mathrm{softmax}(u_j) - \hat{y}_j). \tag{11}$$

- In the minimization of Eqs. (9) and (10), $\tilde{\delta}_{y_i}$ is updated by

$$\tilde{\delta}_{y_i} = -\frac{\lambda}{N_{y_i}} \sum_{j:y_j=y_i} (\mathrm{softmax}(u_j) - \hat{y}_j). \tag{12}$$

PGD-like Optimize

Algorithm 1 PGD-like Optimization

Input: The logit vectors (u_i) for the cth category in the current mini-batch, ϵ_c, and α.

1: Let $u_i^0 = u_i$ for the input vectors;
2: Calculate K_c according $K_c = \lfloor \frac{\epsilon_c}{\alpha} \rfloor$;
3: **for** $k = 0$ to $K_c - 1$ **do**
4: Calculate $\left.\frac{\partial l(\mathrm{softmax}(u_i^k + \tilde{\delta}_c), c)}{\partial \tilde{\delta}_c}\right|_{\mathbf{0}} = \mathrm{softmax}(u_i^k) - \hat{c}$;
5: Calculate $\tilde{\delta}_{y_i}^{k+1}$ according to Eq. (11) for maximization and Eq. (12) for minimization;
6: $u_i^{k+1} := u_i^k + \tilde{\delta}_{y_i}^{k+1}$.
7: **end for**

Output: $\delta_c = u_i^{K_c} - u_i$

Bound Calculation

Balanced classification

$$\epsilon_c = \epsilon + \Delta\epsilon \left| \tau - \bar{q}_c \right| . \tag{13}$$

Long-tail classification

$$\epsilon_c = \begin{cases} \epsilon + \Delta\epsilon \frac{\bar{q}_c}{\bar{q}_1} & c \leq \tau \\ \epsilon + \Delta\epsilon \frac{\bar{q}_C}{\bar{q}_c} & c > \tau \end{cases} . \tag{14}$$

Learning to Perturb Logit

Algorithm 2 Learning to Perturb Logits (LPL)

Input: S, τ, max iteration T, hyper-parameters for PGD-like optimization, and other conventional training hyper-parameters.

```
Randomly initialize Θ.
for t = 0 to T do
    Sample a mini-batch from S;
    Update τ if it is not fixed (e.g., mean(q̄_c) is used) and split the category set;
    Compute ε_c for each category using (13) and (14) if varied bounds are used;
    Infer δ_c for each category using a PGD-like optimization method for (9) in balanced
    classification or (10) in long-tail classification;
    Update the logits for each sample and compute the new cross entropy loss;
    Update Θ with SGD.
end for
```

Output: Θ

Experiments on Data Augmentation

Method	Wide-ResNet-28-10	ResNet-110
Basic	$3.82 \pm 0.15\%$	$6.76 \pm 0.34\%$
Large Margin	$3.69 \pm 0.10\%$	$6.46 \pm 0.20\%$
Disturb Label	$3.91 \pm 0.10\%$	$6.61 \pm 0.04\%$
Focal Loss	$3.62 \pm 0.07\%$	$6.68 \pm 0.22\%$
Center Loss	$3.76 \pm 0.05\%$	$6.38 \pm 0.20\%$
Lq Loss	$3.78 \pm 0.08\%$	$6.69 \pm 0.07\%$
CGAN	$3.84 \pm 0.07\%$	$6.56 \pm 0.14\%$
ACGAN	$3.81 \pm 0.11\%$	$6.32 \pm 0.12\%$
infoGAN	$3.81 \pm 0.05\%$	$6.59 \pm 0.12\%$
ISDA	$3.58 \pm 0.15\%$	$6.33 \pm 0.19\%$
ISDA+DropOut	$3.58 \pm 0.15\%$	$5.98 \pm 0.20\%$
LPL (mean+ fixed ϵ_c)	$3.39 \pm 0.04\%$	$5.83 \pm 0.21\%$
LPL (mean+ varied ϵ_c)	**$3.37 \pm 0.04\%$**	**$5.72 \pm 0.05\%$**

Table: Test Top-1 errors on CIFAR10.

Method	Wide-ResNet-28-10	ResNet-110
Basic	$18.53 \pm 0.07\%$	$28.67 \pm 0.44\%$
Large Margin	$18.48 \pm 0.05\%$	$28.00 \pm 0.09\%$
Disturb Label	$18.56 \pm 0.22\%$	$28.46 \pm 0.32\%$
Focal Loss	$18.22 \pm 0.08\%$	$28.28 \pm 0.32\%$
Center Loss	$18.50 \pm 0.25\%$	$27.85 \pm 0.10\%$
Lq Loss	$18.43 \pm 0.37\%$	$28.78 \pm 0.35\%$
CGAN	$18.79 \pm 0.08\%$	$28.25 \pm 0.36\%$
ACGAN	$18.54 \pm 0.05\%$	$28.48 \pm 0.44\%$
infoGAN	$18.44 \pm 0.10\%$	$27.64 \pm 0.14\%$
ISDA	$17.98 \pm 0.15\%$	$27.57 \pm 0.46\%$
ISDA+DropOut	$17.98 \pm 0.15\%$	$26.35 \pm 0.30\%$
LPL (mean+ fixed ϵ_c)	$18.19 \pm 0.07\%$	$26.09 \pm 0.16\%$
LPL (mean+ varied ϵ_c)	**$17.61 \pm 0.30\%$**	**$25.87 \pm 0.07\%$**

Table: Test Top-1 errors on CIFAR100.

Experiments on Data Augmentation

Method	#Params	CIFAR10	CIFAR100
ResNet-32+ISDA	0.5M	$7.09 \pm 0.12\%$	$30.27 \pm 0.34\%$
ResNet-32+LPL (mean + fixed ϵ_c)	0.5M	$7.01 \pm 0.16\%$	$29.59 \pm 0.27\%$
ResNet-32+LPL (mean + varied ϵ_c)	0.5M	**$6.66 \pm 0.09\%$**	**$28.53 \pm 0.16\%$**
SE-Resnet110+ISDA	1.7M	$5.96 \pm 0.21\%$	$26.63 \pm 0.21\%$
SE-Resnet110+LPL (mean + fixed ϵ_c)	1.7M	$5.87 \pm 0.17\%$	$26.12 \pm 0.24\%$
SE-Resnet110+LPL (mean + varied ϵ_c)	1.7M	**$5.39 \pm 0.10\%$**	**$25.70 \pm 0.07\%$**
Wide-ResNet-16-8+ISDA	11.0M	$4.04 \pm 0.29\%$	$19.91 \pm 0.21\%$
Wide-ResNet-16-8+LPL (mean + fixed ϵ_c)	11.0M	$3.97 \pm 0.09\%$	$19.87 \pm 0.02\%$
Wide-ResNet-16-8+LPL (mean + varied ϵ_c)	11.0M	**$3.93 \pm 0.10\%$**	**$19.83 \pm 0.09\%$**

Table: Number of parameters and test Top-1 errors of ISDA and LPL with different base networks.

Experiments on Long-tail Classification

Ratio	100:1	10:1
Class-balanced CE loss	61.23%	42.43%
Class-balanced fine-tuning	58.50%	42.43%
Meta-weight net	58.39%	41.09%
Focal Loss	61.59%	44.22%
Class-balanced focal loss	60.40%	42.01%
LDAM	59.40%	42.71%
LDAM-DRW	57.11%	41.22%
ISDA + Dropout	62.60%	44.49%
LA	56.11%	41.66%
LPL (varied τ + fixed ϵ_c)	58.03%	41.86%
LPL (varied τ + varied ϵ_c)	**55.75%**	**39.03%**

Table: Test Top-1 errors on CIFAR100-LT

Ratio	100:1	10:1
Class-balanced CE loss	27.32%	13.10%
Class-balanced fine-tuning	28.66%	16.83%
Meta-weight net	26.43%	12.45%
Focal Loss	29.62%	13.34%
Class-balanced focal loss	25.43%	12.52%
LDAM	26.45%	12.68%
LDAM-DRW	21.88%	11.63%
ISDA + Dropout	27.45%	12.98%
LA	22.33%	11.07%
LPL (varied τ + fixed ϵ_c)	23.97%	11.09%
LPL (varied τ + varied ϵ_c)	**22.05%**	**10.59%**

Table: Test Top-1 errors on CIFAR10-LT

Combination Method

In ISDA and LA, the perturbations are directly calculated rather than optimization, we propose a combination method with LA loss in imbalance image classification.

$$
\begin{aligned}
&\sum_{c\in\mathcal{N}_a}\sum_{x_i\in S_c}\min_{\|\tilde{\delta}_{y_i}\|\le\epsilon_c} l(\mathrm{softmax}(u_i+\lambda\log\pi_{y_i}+\tilde{\delta}_{y_i}),y_i)\\
&+\sum_{c\in\mathcal{P}_a}\sum_{x_i\in S_c}\max_{\|\tilde{\delta}_{y_i}\|\le\epsilon_c} l(\mathrm{softmax}(u_i+\lambda\log\pi_{y_i}+\tilde{\delta}_{y_i}),y_i).
\end{aligned}
\tag{15}
$$

Method	CIFAR10-LT100	CIFAR100-LT100
LA	22.33%	56.11%
LPL	22.05%	55.75%
LA+LPL	**21.46%**	**53.89%**

Table: Test Top-1 errors of three methods on two data sets.

Loss Variations of LPL

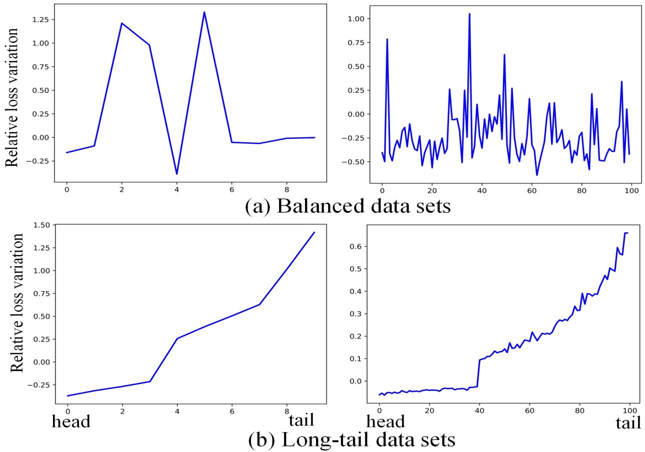

Figure: Relative loss variations of our LPL

Conclusions

- A conjecture for the relationship between (logit perturbation-incurred) loss increment/decrement and positive/negative data augmentation is proposed.
- LPL achieves the best performances in both situations under different basic networks.
- Existing methods with logit perturbation (e.g. LA) can also be improved by using our method.

The End