Decoupling Representation and Classifier for Long-Tailed Recognition

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, Yannis Kalantidis
Long-tailed classification

Problem statement
- Training set: long-tailed distribution
  - Head v.s. Tail
- Testing set: balanced distribution
- Evaluation: three splits based on cardinality

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.
The problem behind long-tail classification performance

- Representation Quality
- Classifier Quality

Final Performance

- Normal Training

- Representation
- Classifier
- Performance
The problem behind long-tail classification performance is determined by the quality of representation and classifier.
The problem behind long-tail

Classification performance $\equiv$ Representation Quality $\oplus$ Classifier Quality

NOTE: Such observations are drawn empirically!
Notations

- Feature representation: $f(x; \theta) = z$
- Linear classifiers: $g_i(z) = W_i^T z + b$
- Final prediction: $\hat{y} = \text{argmax } g_i(z)$
What is the problem with the classifier?

After joint training with instance-balanced sampling, the norms of the weights $\|w_j\|$ are correlated with the size of the classes $n_j$. 

ImageNet_LT

ResNext50

Jointly learned classifier

Small weight scale; Small confidence score; Poor performance.

Dataset distribution

Weight norm visualization

Joint

cRT

$\tau$-Norm

LWS

data

Class Index

Many

Medium

Few
How to improve the classifier? -- Three ways

KEY: break the norm v.s. class size correlation.

I. Classifier Retraining (cRT)

- Freeze the representation.
- Retrain the linear classifier with class-balanced sampling.
How to improve the classifier? -- Three ways

KEY: break the norm v.s. #data correlation.

I. Classifier Retraining (cRT)
   - Freeze the representation.
   - Retrain the linear classifier with class-balanced sampling.

II. Tau-Normalization ($\tau$-Norm)
   - Adjust the classifier weight norms directly
     \[ \tilde{w}_i = \frac{w_i}{||w_i||^\tau} \]
   - Tau is “temperature” of the normalization.
How to improve the classifier? -- Three ways

KEY: break the norm v.s. #data correlation.

I. Classifier Retraining (cRT)
   - Freeze the representation.
   - Retrain the linear classifier with class-balanced sampling

II. Tau-Normalization (τ-Norm)
   - Adjust the classifier weight norms directly.
     \[ \tilde{w}_i = \frac{w_i}{\|w_i\|^\tau} \]
   - Tau is “temperature” of the normalization.

III. Learnable Weight Scaling (LWS)
   - Tune the scale of each weight vector
     \[ \tilde{w}_i = f_i \cdot w_i, \text{ where } f_i = \frac{1}{\|w_i\|^\tau} \]
Classifier Rebalancing

- Without classifier rebalancing (i.e. Joint training), progressively-balanced sampling works best
- When instance-balanced sampling is used and classifiers are re-balanced, medium-shot, and few-shot performance increases significantly, and achieve best results
How Does Classifier Rebalancing Work?

- Larger weights ==> Wider classification cone
- Un-normalized weights ==> Unbalanced decision boundaries
- Classifier rebalancing ==> More balanced decision boundaries

\[ \tilde{w}_i = \frac{w_i}{||w_i||_\tau} \]

\[ \tau \to 0 \]

\[ \tau \to 1 \]
Can we finetune both trunk and classifier?

The best performance is achieved when only classifier is retrained, and backbone model is fixed.

Table 1: Retraining/finetuning different parts of a ResNeXt-50 model on ImageNet-LT. B: backbone; C: classifier; LB: last block.

| Re-train      | Many | Medium | Few  | All  |
|---------------|------|--------|------|------|
| B+C           | 55.4 | 45.3   | 24.5 | 46.3 |
| B+C(0.1×lr)   | 61.9 | 45.6   | 22.8 | 48.8 |
| LB+C          | 61.4 | 45.8   | 24.5 | 48.9 |
| C             | 61.5 | **46.2** | **27.0** | **49.5** |
Experiments

Datasets

I. ImageNet_LT
   - Constructed from ImageNet 2012
   - 1000 categories, 115.8k images

II. iNaturalist 2018
   - Contains only species.
   - 8142 categories, 437.5k images

III. Places_LT
   - Constructed from Places365
   - 365 classes
Experiments

Datasets

I. ImageNet_LT

- Constructed from ImageNet 2012
- 1000 categories, 115.8k images

- From joint to LWS/cRT/tau-norm, with little sacrifice on many shot
- New SOTA can be achieved
- Improvement on Medium: ~10, few: 20+

| Classifier          | Many | Medium | Few  | All  |
|--------------------|------|--------|------|------|
| OLTR               | 43.2 | 35.1   | 18.5 | 35.6 |
| OLTR(rerun)        | 40.7 | 33.3   | 18.1 | 34.1 |
| Joint              | 65.9 | 37.5   | 7.7  | 44.4 |
| NCM                | 56.6 | 45.3   | 28.1 | 47.3 |
| cRT                | 61.8 | 46.2   | 27.4 | 49.6 |
| τ-normalized       | 59.1 | 46.9   | 30.7 | 49.4 |
| LWS                | 60.2 | **47.2** | 30.3 | **49.9** |
Experiments

Datasets

II. iNaturalist 2018

- Contains only species.
- 8142 categories, 437.5k images

- From joint to cRT/tau-norm, little sacrifice on head classes, Large gain on tail classes.
- Once representation is sufficiently trained, New SOTA can be easily obtained.

```
| Classifier     | Many   | Medium   | Few     | All     |
|----------------|--------|----------|---------|---------|
| CB-Focal       | -      | -        | -       | 61.1    |
| LDAM           | -      | -        | -       | 64.6    |
| LDAM+DRAW      | -      | -        | -       | 68.0    |
| Joint          | 72.2/75.7 | 63.0/66.9 | 57.2/61.7 | 61.7/65.8 |
| NCM            | 55.5/61.0 | 57.9/63.5 | 59.3/63.6 | 58.2/63.1 |
| cRT            | 69.0/73.2 | 66.0/68.8 | 63.2/66.1 | 65.2/68.2 |
| τ-normalized   | 65.6/71.1 | 65.3/68.9 | 65.9/69.3 | 65.6/69.3 |
```

* Notation: 90 epochs/200 epochs
Take home messages

- For solving long-tailed recognition problem, representation and classifiers should be considered separately.
- Our methods achieve performance gain by finding a better tradeoff (currently the best one) between head and tail classes.
- Future research might be focusing more on improving representation quality.

Code is available!
https://github.com/facebookresearch/classifier-balancing