

COCO + LVIS Joint Recognition Challenge

Workshop at ECCV 2020

Balanced Activation for Long-tailed Visual Recognition

Jiawei Ren*, Cunjun Yu*, Zhongang Cai*, and Haiyu Zhao

Innova

Abstract. Deep classifiers have achieved great success in visual recognition. However, real-world data is long-tailed by nature, leading to the mismatch between training and testing distributions. In this report, we introduce Balanced Activation (Balanced Softmax and Balanced Sigmoid), an elegant unbiased, and simple extension of Sigmoid and Softmax activation function, to accommodate the label distribution shift between training and testing in object detection. We derive the generalization bound for multiclass Softmax regression and show our loss minimizes the bound. In our experiments, we demonstrate that Balanced Activation generally provides $\sim 3\%$ gain in terms of mAP on LVIS-1.0 and outperforms the current state-of-the-art methods without introducing any extra parameters.

1 Introduction

Most real-world data comes with a long-tailed nature: a few high-frequency classes (or head classes) contribute to most of the observations, while a large number of low-frequency classes (or tail classes) are under-represented in data. In practice, the number of samples per class generally decreases from head to tail classes exponentially. Under the power law, the tails can be undesirably heavy. A model that minimizes empirical risk on long-tailed training datasets often underperforms on a class-balanced test dataset. As datasets are scaling up nowadays, long-tailed nature poses critical difficulties in many vision tasks.

An intuitive solution to long-tailed task is to re-balance the data distribution. Most state-of-the-art (SOTA) methods use the class-balanced sampling or loss re-weighting to “simulate” a balanced training set [1, 9]. However, they may under-represent the head class or have gradient issues during optimization.

In this report, we first show that the Softmax function is intrinsically biased in the long-tailed scenario. We derive Balanced Activation (Balanced Softmax and Balanced Sigmoid) from the probabilistic perspective that explicitly models the test-time label distribution shift. Theoretically, we found that optimizing for the Balanced Softmax cross-entropy loss is equivalent to minimizing the generalization error bound. We take the off-the-shelf Mask-RCNN as our baseline model and obtain $\sim 3\%$ mAP improvements on LVIS-1.0 via Balanced Activation without adding any extra computational overhead.

* equal contribution

2 Background

Label Distribution Shift We begin by revisiting the multi-class Softmax regression, where we are generally interested in estimating the conditional probability $p(y|x)$, which can be modeled as a multinomial distribution ϕ :

$$\phi = \phi_1^{\mathbf{1}\{y=1\}} \phi_2^{\mathbf{1}\{y=2\}} \dots \phi_k^{\mathbf{1}\{y=k\}}; \quad \phi_j = \frac{e^{\eta_j}}{\sum_{i=1}^k e^{\eta_i}}; \quad \sum_{j=1}^k \phi_j = 1 \quad (1)$$

From the Bayesian inference’s perspective, ϕ_j can also be interpreted as:

$$\phi_j = p(y = j|x) = \frac{p(x|y = j)p(y = j)}{p(x)} \quad (2)$$

where $p(y = j)$ is in particular interest under the class-imbalanced setting. Assuming that all instances in the training dataset and the test dataset are generated from the same process $p(x|y = j)$, there could still be a discrepancy between training and testing given different label distribution $p(y = j)$ and evidence $p(x)$. With a slight abuse of the notation, we re-define ϕ to be the conditional distribution on the balanced test set and define $\hat{\phi}$ to be the conditional probability on the imbalanced training set. As a result, standard Softmax provides a biased estimation for ϕ .

3 Balanced Activation

To eliminate the discrepancy between the posterior distributions of training and testing, we introduce the Balanced Activation (Balanced Softmax and Balanced Sigmoid). We reach the same conclusion from both label distribution shift (Section 2) and generalization error bound perspectives (Section 3.2) that gives rise to Balanced Softmax. The same strategy is used to deduct Balanced Sigmoid. **We refer readers to Section 3.4 for a summarized version of our findings.**

3.1 From Label Distribution Shift Perspective

We use the same model outputs η to parameterize two conditional probabilities: ϕ for testing and $\hat{\phi}$ for training.

Theorem 1. *Assume ϕ to be the desired conditional probability of the balanced data set, with the form $\phi_j = p(y = j|x) = \frac{p(x|y=j)}{p(x)} \frac{1}{k}$, and $\hat{\phi}$ to be the desired conditional probability of the imbalanced training set, with the form $\hat{\phi}_j = \hat{p}(y = j|x) = \frac{p(x|y=j)}{\hat{p}(x)} \frac{n_j}{\sum_{i=1}^k n_i}$. If ϕ is expressed by the standard Softmax function of model output η , then $\hat{\phi}$ can be expressed as*

$$\hat{\phi}_j = \frac{n_j e^{\eta_j}}{\sum_{i=1}^k n_i e^{\eta_i}}. \quad (3)$$

We use the exponential family parameterization to prove Theorem 1.

Proof. Observe that the conditional probability of a categorical distribution can be parameterized as an exponential family. It gives us a standard Softmax function as an *inverse parameter mapping*

$$\phi_j = \frac{e^{\eta_j}}{\sum_{i=1}^k e^{\eta_i}} \quad (4)$$

and also a *canonical link function*:

$$\eta_j = \log\left(\frac{\phi_j}{\phi_k}\right) \quad (5)$$

We begin by adding a term $-\log(\phi_j/\hat{\phi}_j)$ to both sides of Eqn. 5,

$$\eta_j - \log\frac{\phi_j}{\hat{\phi}_j} = \log\left(\frac{\phi_j}{\phi_k}\right) - \log\left(\frac{\phi_j}{\hat{\phi}_j}\right) = \log\left(\frac{\hat{\phi}_j}{\phi_k}\right) \quad (6)$$

Subsequently,

$$\phi_k e^{\eta_j - \log\frac{\phi_j}{\hat{\phi}_j}} = \hat{\phi}_j \quad (7)$$

$$\phi_k \sum_{i=1}^k e^{\eta_i - \log\frac{\phi_i}{\hat{\phi}_i}} = \sum_{i=1}^k \hat{\phi}_i = 1 \quad (8)$$

$$\phi_k = 1 / \sum_{i=1}^k e^{\eta_i - \log\frac{\phi_i}{\hat{\phi}_i}} \quad (9)$$

Substitute Eqn. 9 back to Eqn. 7, we have

$$\hat{\phi}_j = \phi_k e^{\eta_j - \log\frac{\phi_j}{\hat{\phi}_j}} = \frac{e^{\eta_j - \log\frac{\phi_j}{\hat{\phi}_j}}}{\sum_{i=1}^k e^{\eta_i - \log\frac{\phi_i}{\hat{\phi}_i}}} \quad (10)$$

Recall that

$$\phi_j = p(y = j|x) = \frac{p(x|y = j)}{p(x)} \frac{1}{k}; \quad \hat{\phi}_j = \hat{p}(y = j|x) = \frac{p(x|y = j)}{\hat{p}(x)} \frac{n_j}{n} \quad (11)$$

then

$$\log\frac{\phi_j}{\hat{\phi}_j} = \log\frac{n}{kn_j} + \log\frac{\hat{p}(x)}{p(x)} \quad (12)$$

Finally, bring Eqn. 12 back to Eqn. 10

$$\hat{\phi}_j = \frac{e^{\eta_j - \log\frac{n}{kn_j} - \log\frac{\hat{p}(x)}{p(x)}}}{\sum_{i=1}^k e^{\eta_i - \log\frac{n}{kn_i} - \log\frac{\hat{p}(x)}{p(x)}}} = \frac{n_j e^{\eta_j}}{\sum_{i=1}^k n_i e^{\eta_i}} \quad (13)$$

Theorem 1 essentially shows that applying the following Balanced Softmax function can naturally accommodate the label distribution shifts between the training and test sets. We define the Balanced Softmax function as

$$\hat{l}(\theta) = -\log(\hat{\phi}_y) = -\log\left(\frac{n_y e^{\eta_y}}{\sum_{i=1}^k n_i e^{\eta_i}}\right). \quad (14)$$

3.2 From Generalization Error Bound Perspective

Generalization error bound gives the upper bound of a model's test error, given its training error. With drastically fewer training samples, the tail classes have much higher generalization error bounds than the head classes, making high classification accuracy on tail classes unlikely. In this section, we show that optimizing Eqn. 47 is equivalent to minimizing the generalization upper bound.

Margin theory provides a bound based on the margins [5]. Margin bounds usually negatively correlate to the magnitude of the margin, i.e., a larger margin leads to lower generalization error. Consequently, given a constraint on the sum of margins of all classes, there would be a trade-off between minority classes and majority classes [2].

Locating such an optimal margin for multi-class classification is non-trivial. The bound investigated in [2] was established for binary classification using hinge loss. Here, we try to develop the margin bound for the multi-class Softmax regression. Given the previously defined ϕ and $\hat{\phi}$, we derive $\hat{l}(\theta)$ by minimizing the margin bound. Margin bound commonly bounds the 0-1 error:

$$err_{0,1} = \Pr \left[\theta_y^T f(x) < \max_{i \neq y} \theta_i^T f(x) \right]. \quad (15)$$

However, directly using the 0-1 error as the loss function is not ideal for optimization. Instead, negative log likelihood (NLL) is generally considered more suitable. With continuous relaxation of Eqn. 15, we have

$$err(t) = \Pr[t < \log(1 + \sum_{i \neq y} e^{\theta_i^T f(x) - \theta_y^T f(x)})] = \Pr[l(\theta) > t], \quad (16)$$

where $t \geq 0$ is any threshold, and $l_y(\theta)$ is a standard negative log-likelihood with Softmax. This new error is still a counter, but describes how likely the test loss will be larger than a given threshold. Naturally, we define our margin for class j to be

$$\gamma_j = t - \max_{(x,y) \in S_j} l_j(\theta). \quad (17)$$

where S_j is the set of all class j samples. If we force a large margin γ_j during training, i.e., force the training loss to be much lower than t , then $err(t)$ will be reduced.

Theorem 2. *Let $t \geq 0$ be any threshold, for all $\gamma_j > 0$, with probability at least $1 - \delta$, we have*

$$err_{bal}(t) \lesssim \frac{1}{k} \sum_{j=1}^k \left(\frac{1}{\gamma_j} \sqrt{\frac{C}{n_j}} + \frac{\log n}{\sqrt{n_j}} \right); \quad \gamma_j^* = \frac{\beta n_j^{-1/4}}{\sum_{i=1}^k n_i^{-1/4}}, \quad (18)$$

where $err_{bal}(t)$ is the error on the balanced test set, \lesssim is used to hide constant terms and C is some measure on complexity. With a constraint on $\sum_{j=1}^k \gamma_j = \beta$, Cauchy-Schwarz inequality gives us the optimal γ_j^* .

Proof. Firstly, we define f as,

$$f(x) - l(\theta) + t \quad (19)$$

where $l(\theta)$ and t is previously defined in submission. However, f does not have a specific semantic meaning as it is defined only to keep consistent with notations in [5].

Let $err_j(t)$ be the 0-1 loss on example from class j

$$err_j(t) = \Pr_{(x,y) \in S_j} [f(x) < 0] = \Pr_{(x,y) \in S_j} [l(\theta) > t] \quad (20)$$

and $err_{\gamma,j}(t)$ be the 0-1 margin loss on example from class j

$$err_{\gamma,j}(t) = \Pr_{(x,y) \in S_j} [f(x) < \gamma_j] = \Pr_{(x,y) \in S_j} [l(\theta) + \gamma_j > t] \quad (21)$$

Let $\hat{err}_{\gamma,j}(t)$ denote the empirical variant of $err_{\gamma,j}(t)$.

For any $\delta > 0$ and with probability at least $1 - \delta$, for all $\gamma_j > 0$, and $f \in \mathcal{F}$, Theorem 2 in [5] directly gives us

$$err_j(t) \leq \hat{err}_{\gamma,j}(t) + \frac{4}{\gamma_j} \hat{\mathfrak{R}}_j(\mathcal{F}) + \sqrt{\frac{\log(\log_2 \frac{4B}{\gamma_j})}{n_j}} + \sqrt{\frac{\log(1/\delta)}{2n_j}} \quad (22)$$

where $\sup_{(x,y) \in S} |l(\theta) - t| \leq B$ and $\hat{\mathfrak{R}}_j(\mathcal{F})$ denotes the empirical Rademacher complexity of function family \mathcal{F} . By applying [2]'s analysis on the empirical Rademacher complexity and union bound over all classes, we have the generalization error bound for the loss on a balanced test set

$$err_{bal}(t) \leq \frac{1}{k} \sum_{j=1}^k \left(\hat{err}_{\gamma,j}(t) + \frac{4}{\gamma_j} \sqrt{\frac{C(\mathcal{F})}{n_j}} + \epsilon_j(\gamma_j) \right) \quad (23)$$

where

$$\epsilon_j(\gamma_j) \triangleq \sqrt{\frac{\log(\log_2 \frac{4B}{\gamma_j})}{n_j}} + \sqrt{\frac{\log(1/\delta)}{2n_j}} \quad (24)$$

is a low-order term of n_j . To minimize the generalization error bound Eqn. 23, we essentially need to minimize

$$\sum_{j=1}^k \frac{4}{\gamma_j} \sqrt{\frac{C(\mathcal{F})}{n_j}} \quad (25)$$

By constraining the sum of γ as $\sum_{j=1}^k \gamma_j = \beta$, we can directly apply Cauchy-Schwarz inequality to solve the optimal γ

$$\gamma_j^* = \frac{\beta n_j^{-1/4}}{\sum_{i=1}^k n_i^{-1/4}}. \quad (26)$$

The optimal γ^* suggests that we need larger γ for the classes with fewer samples. In other words, to achieve the optimal generalization ability, we need to focus on minimizing the training loss of the tail classes. Therefore, for each class j , the desired training loss $\hat{l}_j^*(\theta)$ is

$$\hat{l}_j^*(\theta) = l_j(\theta) + \gamma_j^*, \quad \text{where } l_j(\theta) = -\log(\phi_j), \quad (27)$$

Corollary 1. $\hat{l}_j^*(\theta) = l_j(\theta) + \gamma_j^* = l_j(\theta) + \frac{\beta n_j^{-1/4}}{\sum_{i=1}^k n_i^{-1/4}}$ can be approximated by $\hat{l}_j(\theta)$ when:

$$\hat{l}_j(\theta) = -\log(\hat{\phi}_j); \quad \hat{\phi}_j = \frac{e^{\eta_j - \log \gamma_j^*}}{\sum_{i=1}^k e^{\eta_i - \log \gamma_i^*}} = \frac{n_j^{\frac{1}{4}} e^{\eta_j}}{\sum_{i=1}^k n_i^{\frac{1}{4}} e^{\eta_i}} \quad (28)$$

Proof. Notice that $\hat{l}_j^*(\theta) = l_j(\theta) + \gamma_j^*$ can not be achieved, because $-\log \hat{\phi}_j^* = -\log \phi_j + \gamma_j^*$ and $\gamma_j^* > 0$ implies

$$\hat{\phi}_j^* < \phi_j; \quad \sum_{j=1}^k \hat{\phi}_j^* < \sum_{j=1}^k \phi_j = 1 \quad (29)$$

The equation above contradicts with the definition that sum of $\hat{\phi}^*$ should be exactly equal to 1. To solve the contradiction, we introduce a term $\gamma_{base} > 0$, such that

$$-\log \hat{\phi}_j^* = -\log \phi_j - \gamma_{base} + \gamma_j^*; \quad \sum_{j=1}^k \hat{\phi}_j^* = 1 \quad (30)$$

To justify the new term γ_{base} , we recall the definition of error

$$err_{\gamma,j}(t) = \Pr_{(x,y) \in S_j} [l(\theta) + \gamma_j > t]; \quad err_{bal}(t) = \Pr_{(x,y) \in S_{bal}} [l(\theta) > t] \quad (31)$$

If we tweak the threshold t with the term γ_{base}

$$err_{\gamma,j}(t + \gamma_{base}) = \Pr_{(x,y) \in S_j} [l(\theta) + \gamma_j > t + \gamma_{base}] = \Pr_{(x,y) \in S_j} [(l(\theta) - \gamma_{base}) + \gamma_j > t] \quad (32)$$

$$err_{bal}(t + \gamma_{base}) = \Pr_{(x,y) \in S_{bal}} [l(\theta) > t + \gamma_{base}] = \Pr_{(x,y) \in S_{bal}} [(l(\theta) - \gamma_{base}) > t] \quad (33)$$

As γ^* is not a function of t , the value of γ^* will not be affected by the tweak. Thus, instead of looking for $\hat{l}_j^*(\theta) = l_j(\theta) + \gamma_j^*$ that minimizes the generalization bound for $err_{bal}(t)$, we are in fact looking for $\hat{l}_j^*(\theta) = (l_j(\theta) - \gamma_{base}) + \gamma_j^*$ that minimizes generalization bound for $err_{bal}(t + \gamma_{base})$

Compare the form we obtained from generalization bound Eqn. 28 to Eqn. 47, we have an additional constant 1/4 before $\log n_j$. We empirically find that setting 1/4 to 1 leads to the optimal results, which may suggest that Eqn. 23 is not necessarily tight. To this point, the label distribution shift and generalization bound of multi-class Softmax regression lead us to the same loss form: Eqn. 47.

3.3 Derivation of Balanced Sigmoid

For instance segmentation, a detector is normally required to predict a background class explicitly. However, the number of background class is uncountable and we need the number of training samples of each class in the training set for Balanced Softmax. Thus, we introduce Balanced Sigmoid to circumvent this problem. By virtue of Bayesian law, a similar strategy can be applied to the multiple binary logistic regressions.

Definition. Multiple Binary Logistic Regression uses k binary logistic regression to do multi-class classification. Same as Softmax regression, the predicted label is the class with the maximum model output.

$$y_{pred} = \arg \max_j (\eta_j). \quad (34)$$

The only difference is that ϕ_j is expressed by a logistic function of η_j

$$\phi_j = \frac{e^{\eta_j}}{1 + e^{\eta_j}} \quad (35)$$

and the loss function sums up binary classification loss on all classes

$$l(\theta) = \sum_{j=1}^k -\log \tilde{\phi}_j \quad (36)$$

where

$$\tilde{\phi}_j = \begin{cases} \phi_j, & \text{if } y = j \\ 1 - \phi_j, & \text{otherwise} \end{cases} \quad (37)$$

Setup. By the virtue of Bayes Rule, ϕ_j and $1 - \phi_j$ can be decomposed as

$$\phi_j = \frac{p(x|y=j)p(y=j)}{p(x)}; \quad 1 - \phi_j = \frac{p(x|y \neq j)p(y \neq j)}{p(x)} \quad (38)$$

and for $\hat{\phi}$ and $1 - \hat{\phi}$,

$$\hat{\phi}_j = \frac{p(x|y=j)\hat{p}(y=j)}{\hat{p}(x)}; \quad 1 - \hat{\phi}_j = \frac{p(x|y \neq j)\hat{p}(y \neq j)}{\hat{p}(x)} \quad (39)$$

Derivation. Again, we introduce the exponential family parameterization and have the following link function for ϕ_j

$$\eta_j = \log \frac{\phi_j}{1 - \phi_j} \quad (40)$$

Bring the decomposition Eqn. 38 and Eqn.39 into the link function above

$$\eta_j = \log\left(\frac{\hat{\phi}_j}{1 - \hat{\phi}_j} \cdot \frac{\phi_j}{\hat{\phi}_j} \cdot \frac{1 - \hat{\phi}_j}{1 - \phi_j}\right) \quad (41)$$

$$\eta_j = \log\left(\frac{\hat{\phi}_j}{1 - \hat{\phi}_j} \cdot \frac{p(x|y=j)p(y=j)/p(x)}{p(x|y=j)\hat{p}(y=j)/\hat{p}(x)} \cdot \frac{p(x|y \neq j)\hat{p}(y \neq j)/\hat{p}(x)}{p(x|y \neq j)p(y \neq j)/p(x)}\right) \quad (42)$$

Simplify the above equation

$$\eta_j = \log\left(\frac{\hat{\phi}_j}{1 - \hat{\phi}_j} \cdot \frac{p(y=j)}{\hat{p}(y=j)} \cdot \frac{\hat{p}(y \neq j)}{p(y \neq j)}\right) \quad (43)$$

Substitute the n_j in to the equation above

$$\eta_j = \log\left(\frac{\hat{\phi}_j}{1 - \hat{\phi}_j} \cdot \frac{n/k}{n_j} \cdot \frac{n - n_j}{n - n/k}\right) \quad (44)$$

Then

$$\eta_j - \log\left(\frac{n/k}{n_j} \cdot \frac{n - n_j}{n - n/k}\right) = \log\left(\frac{\hat{\phi}_j}{1 - \hat{\phi}_j}\right) \quad (45)$$

Finally, we have

$$\hat{\phi}_j = \frac{e^{\eta_j - \log\left(\frac{n/k}{n_j} \cdot \frac{n - n_j}{n - n/k}\right)}}{1 + e^{\eta_j - \log\left(\frac{n/k}{n_j} \cdot \frac{n - n_j}{n - n/k}\right)}} \quad (46)$$

3.4 TL;DR

The label distribution shift and generalization bound of multi-class Softmax regression lead us to the same Balanced Softmax loss form, for certain class y , we have:

$$\hat{l}(\theta) = -\log(\hat{\phi}_y) = -\log\left(\frac{n_y e^{\eta_y}}{\sum_{i=1}^k n_i e^{\eta_i}}\right). \quad (47)$$

We also provide the Balanced Sigmoid variant for broader usage:

$$\hat{l}(\theta) = -\log(\hat{\phi}_y) = -\log\left(\frac{e^{\eta_y - \log\left(\frac{n/k}{n_y} \cdot \frac{n - n_y}{n - n/k}\right)}}{1 + e^{\eta_y - \log\left(\frac{n/k}{n_y} \cdot \frac{n - n_y}{n - n/k}\right)}}\right) \quad (48)$$

4 Experiments

We first validate the proposed method on the long-tailed version of CIFAR-10 and CIFAR-100. Compared to LVIS, these two datasets are relatively small. Hence, they are suitable for fast validation of our ideas. After that, we show that our method leads to a $\sim 3\%$ improvement over the Repeat Factor Sampling baseline and outperforms current SOTAs by a clear margin on LVIS.

Dataset	CIFAR-10-LT			CIFAR-100-LT		
	200	100	10	200	100	10
Softmax	71.2 ± 0.3	77.4 ± 0.8	90.0 ± 0.2	41.0 ± 0.3	45.3 ± 0.3	61.9 ± 0.1
CBW	72.5 ± 0.2	78.6 ± 0.6	90.1 ± 0.2	36.7 ± 0.2	42.3 ± 0.8	61.4 ± 0.3
CBS	68.3 ± 0.3	77.8 ± 2.2	90.2 ± 0.2	37.8 ± 0.1	42.6 ± 0.4	61.2 ± 0.3
Focal Loss [7]	71.8 ± 2.1	77.1 ± 0.2	90.3 ± 0.2	40.2 ± 0.5	43.8 ± 0.1	60.0 ± 0.6
Class Balance Loss [3]	72.6 ± 1.8	78.2 ± 1.1	89.9 ± 0.3	39.9 ± 0.1	44.6 ± 0.4	59.8 ± 1.1
LDAM [2]	71.2 ± 0.3	77.2 ± 0.2	90.2 ± 0.3	41.0 ± 0.3	45.4 ± 0.1	62.0 ± 0.3
Equalization Loss [8]	72.8 ± 0.2	76.7 ± 0.1	89.9 ± 0.3	43.3 ± 0.1	47.3 ± 0.1	59.7 ± 0.3
Balanced Softmax	79.0 ± 0.8	83.1 ± 0.4	90.9 ± 0.4	45.9 ± 0.3	50.3 ± 0.3	63.1 ± 0.2

Table 1. Top 1 accuracy for CIFAR-10/100-LT. Softmax denotes the standard cross-entropy loss with Softmax, CBW denotes class-balanced weighting, and CBS denotes class-balanced sampling. Balanced Activation generally outperforms SOTA methods, especially when the imbalance factor is high. Note that for other methods, we reproduce a higher accuracy than the reported value in the original papers.

4.1 CIFAR-10/100-LT

We show results under different Imbalance Factors in Table 1. The Imbalance Factor is defined as the number of training instances in the largest class divided by that of the smallest. We also visualize the marginal likelihood $p(y)$ on CIFAR-100-LT to show that Balanced Softmax is more stable under a high imbalance factor. We follow [3] and use ResNet-32 as backbone in all experiments.

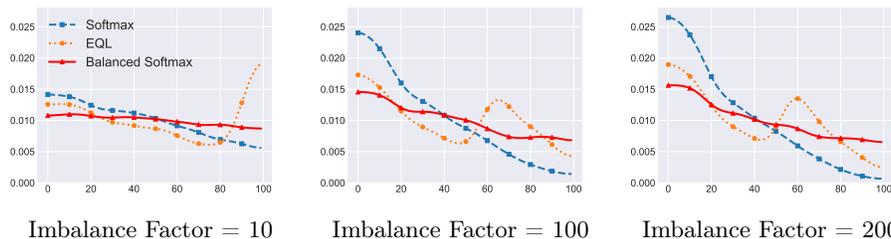


Fig. 1. Experiment on CIFAR-100-LT. the x-axis is the class labels with decreasing training samples and the y-axis is the marginal likelihood $p(y)$. Balanced Softmax is more stable under a high imbalance factor compared to the Softmax baseline and SOTA method, Equalization Loss (EQL).

4.2 LVIS-1.0

For all the experiments on LVIS-1.0, we use the off-the-shelf Mask RCNN and repeat factor sampling as our baseline. We apply scale jitter and random flip at training time (sampling image scale for the shorter side from 640, 672, 704, 736, 768, 800). For testing, images are resized to a shorter image edge of 800 pixels; no

Method	Backbone	AP_{mask}	AP_f	AP_c	AP_r	AP_{box}	Split
Repeat Factor Sampling	ResNet-50	22.1	27.8	21.1	11.6	23.6	Val
cRT	ResNet-50	23.2	27.8	22.5	14.7	24.8	Val
LWS	ResNet-50	23.3	27.8	22.5	14.9	24.8	Val
Equalization Loss	ResNet-50	24.2	27.5	24.2	16.5	25.8	Val
Balanced Sigmoid	ResNet-50	25.1	29.3	24.0	18.3	26.7	Val
Repeat Factor Sampling	ResNet-101	23.2	28.8	22.0	13.5	24.7	Val
cRT	ResNet-101	23.9	28.7	22.7	15.8	25.6	Val
LWS	ResNet-101	24.2	28.8	23.1	16.5	24.7	Val
Equalization Loss	ResNet-101	25.4	28.8	25.3	18.5	27.3	Val
Balanced Sigmoid	ResNet-101	26.4	30.2	25.4	20.1	28.4	Val
Balanced Sigmoid	ResNet-101	26.1	30.4	25.0	19.5	-	test-dev

Table 2. Comparison of various methods using ResNet-50 and ResNet-101 on LVIS-1.0. Balanced Activation generally outperforms SOTA methods by a healthy margin.

test-time augmentation is used. We train all the repeat factor sampling models for 25 epochs and apply decoupled training for 10 epochs using other methods (cRT, LWS, Equalization Loss and Balanced Sigmoid). Models are all trained on 8 GTX 1080Ti GPUs.

We compare our method with four existing methods and show improvements: **Repeat Factor Sampling [4]:** A simple and effective sampling strategy that increases the rate at which tail categories are observed by oversampling the images that contain them. Different from class-balance sampling, it does not force the sampling probability to be equal for all the classes.

Classifier Re-training (cRT) [6]: A straightforward approach, which re-train the classifier with class-balanced sampling. The detector is first trained under the instance balance scheme at the first stage. During the second stage, the feature extractor (backbone) is fixed and only the classifier (regression layer) is optimized during training.

Learnable weight scaling (LWS)[6]: An efficient approach to re-balance the decision boundaries of classifiers. The training setup is similar to **cRT**. The only difference is that there is one more regression layer inserted before the classifier to rebalance the decision boundary.

Equalization Loss [8]: The solution of the winner of LVIS Challenge 2019. It pointed out that randomly dropping some scores of tail classes in the Softmax function can effectively help, by balancing the positive gradients and negative gradients flowing through the scoring output.

5 Conclusion

We have introduced Balanced Activation (Balanced Sigmoid and Balanced Softmax) for long-tail visual recognition tasks. Balanced Activation tackles the distribution shift between training and testing, combining with generalization error bound theory: it optimizes a Balanced Activation which theoretically minimizes

the generalization error bound. We show the method outperforms the current SOTA method without introducing extra parameters.

References

1. Byrd, J., Lipton, Z.C.: What is the effect of importance weighting in deep learning? arXiv preprint arXiv:1812.03372 (2018)
2. Cao, K., Wei, C., Gaidon, A., Arechiga, N., Ma, T.: Learning imbalanced datasets with label-distribution-aware margin loss. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) *Advances in Neural Information Processing Systems* 32, pp. 1567–1578. Curran Associates, Inc. (2019)
3. Cui, Y., Jia, M., Lin, T.Y., Song, Y., Belongie, S.J.: Class-balanced loss based on effective number of samples. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* pp. 9260–9269 (2019)
4. Gupta, A., Dollar, P., Girshick, R.: LVIS: A dataset for large vocabulary instance segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019)
5. Kakade, S.M., Sridharan, K., Tewari, A.: On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In: *Advances in neural information processing systems*. pp. 793–800 (2009)
6. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. *International Conference on Learning Representations* **abs/1910.09217** (2020)
7. Lin, T.Y., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)* pp. 2999–3007 (2017)
8. Tan, J., Wang, C., Li, B., Li, Q., Ouyang, W., Yin, C., Yan, J.: Equalization loss for long-tailed object recognition. *ArXiv* **abs/2003.05176** (2020)
9. Wang, Y.X., Ramanan, D., Hebert, M.: Learning to model the tail. In: *Advances in Neural Information Processing Systems*. pp. 7029–7039 (2017)