




# Output Layer Multiplication for Class Imbalance Problem in Convolutional Neural Networks

Zhao Yang<sup>1</sup> · Yuanxin Zhu<sup>1</sup> · Tie Liu<sup>1</sup> · Sai Zhao<sup>1</sup> · Yunyan Wang<sup>2</sup> · Dapeng Tao<sup>3</sup> 

Accepted: 4 October 2020 / Published online: 19 October 2020  
© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

Convolutional neural networks (CNNs) have demonstrated remarkable performance in the field of computer vision. However, they are prone to suffer from the class imbalance problem, in which the number of some classes is significantly higher or lower than that of other classes. Commonly, there are two main strategies to handle the problem, including dataset-level methods via resampling and algorithmic-level methods by modifying the existing learning frameworks. However, most of these methods need extra data resampling or elaborate algorithm design. In this work we provide an effective but extremely simple approach to tackle the imbalance problem in CNNs with cross-entropy loss. Specifically, we multiply a coefficient  $\alpha > 1$  to output of the last layer in a CNN model. With this modification, the final loss function can dynamically adjust the contributions of examples from different classes during the imbalanced training procedure. Because of its simplicity, the proposed method can be easily applied in the off-the-shelf models with little change. To prove the effectiveness on imbalance problem, we design three experiments on classification tasks of increasing complexity. The experimental results show that our approach could improve the convergence rate in the training stage and/or increase accuracy for test.

**Keywords** Convolutional neural networks · Imbalance learning · Output layer multiplication

## 1 Introduction

Convolutional neural networks (CNNs) have obtained increasing attention in the computer vision community, due to the state-of-the-art performance in kinds of vision problems. It

---

✉ Dapeng Tao  
dapeng.tao@gmail.com

<sup>1</sup> School of Mechanical and Electric Engineering, Guangzhou University, Guangzhou, People's Republic of China

<sup>2</sup> School of Electrical and Electronic Engineering, Hubei University of Technology, Wuhan, People's Republic of China

<sup>3</sup> School of Information Science and Engineering, Yunnan University, Kunming, People's Republic of China

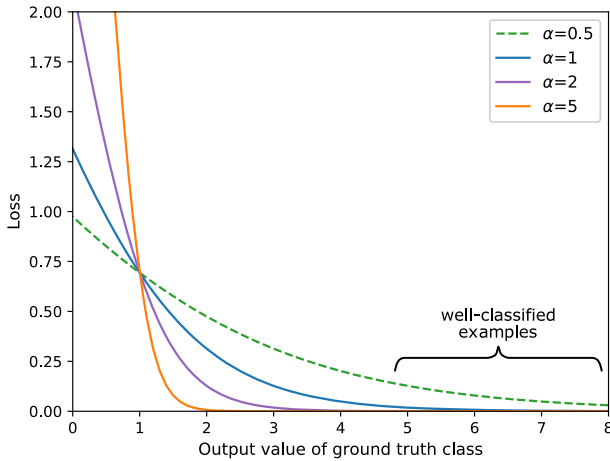
includes tasks such as image classification [21, 29], object detection [36, 37], semantic segmentation [9, 38], and so on. Despite the success, it has been shown that CNNs are prone to the class imbalance problem [5], which exists in various practical applications. For instance, in object detection tasks [34, 39, 40], there is an inevitable foreground–background class imbalance, because the vast majority of the bounding boxes are labeled as background and few boxes contain specific objects. In facial attribute recognition [12, 24, 30], there is a severe imbalance between different attributes, as most of the datasets are drawn from the Internet using search engines without elaborate manual selections. Besides, many real-world datasets [10, 11, 32, 49, 56] exhibit roughly imbalanced or skewed distributions with a long tail, as it is impractical or difficult to collect equal examples for all classes with the increasing number of classes.

These class imbalance problems may bring out a detrimental effect on training traditional models, including convergence characteristics during the training phase and generalization in the test stage [5, 23]. To tackle the problem, a lot of approaches have been proposed in related research works. Among these approaches, the most common and direct way is resampling [5] technique which operates the original dataset to achieve balance between majority classes and minority classes. Commonly, there are two kinds of resampling manners. The first manner is to synthesize new examples from the data of minority classes by oversampling [1, 7, 19]. The second is undersampling technique which reduces some examples from the majority classes [3–53]. Compared with undersampling, oversampling is more widely used and proven to be helpful and robust in some imbalance problems [22].

Another effective way to tackle the problem of class imbalance is based on algorithmic-level by modifying the existing learning algorithms. Typical methods include cost-sensitive learning [2, 14, 27, 35, 54], ensemble learning [8, 16, 33, 44] and reweighting [26, 34, 42]. Cost-sensitive learning approaches assign a higher cost value to the misclassified examples from minority classes, in this way obtaining better generalization for minority examples. Ensemble learning methods usually incorporate resampling methods with boosting or bagging algorithms to increase the accuracy. In reweighting based methods, weight coefficients are directly assigned to different samples by considering the balance of the total loss function.

Although the above mentioned methods have shown satisfactory results in several imbalance problems, most of these methods need extra data resampling or elaborate algorithm design. In this work, we present an effective yet extremely simple approach for the imbalance problem, and it can be easily adopted in existing CNN models. The proposed method is motivated by the focal loss [34], which is used to solve the imbalance problem in object detection. In the focal loss, the basic cross-entropy is modified by adding an adjustable factor  $(1 - p_t)^\gamma$ , which could dynamically reduce the contributions of easy examples and focal on hard examples during the imbalanced training. But very different from the focal loss, we only multiply a coefficient  $\alpha > 1$  to output of the last layer instead of any change on the cross-entropy loss function. By the modification, the final loss function achieves a similar effect with the focal loss, which could dynamically adjust the contributions of examples from majority classes and minority classes.

The effectiveness of coefficient  $\alpha$  can be seen from Fig. 1.  $\alpha = 1$  degenerates to the basic cross-entropy loss. When  $\alpha > 1$ , the final loss function can reduce the relative cost for well-classified examples and concentrate on hard-classified examples.  $\alpha = 0.5$  is a negative situation that is only used for checking the trend of parameter. To show the performance of our proposed approach for the imbalance problem, we provide experiments based on three different common datasets. The experimental results indicate that our proposed approach could improve the convergence rate in the training stage and/or improve the generalization of the model in the test stage to a certain degree.



**Fig. 1** Olymp loss curves with different coefficient  $\alpha$ . On the right part the figure, a large x-coordinate value corresponds to a low loss, which can be regarded as a well-classified example. When setting  $\alpha > 1$ , the final loss function can reduce the relative cost for well-classified examples and concentrate on hard-classified examples. So the coefficient could automatically decrease the contributions of easy samples and make the model concentrate on hard examples during the imbalanced training procedure.  $\alpha = 1$  degenerates to the basic cross-entropy loss, and  $\alpha = 0.5$  is a negative situation that is only used for checking the trend of the parameter. Best viewed in color

The main contributions of this work are three-fold: (1) We propose an effective but extremely concise approach called Olymp to tackle the imbalance problem for softmax loss based classification tasks. (2) We provide a theoretical analysis to illustrate the principle of the Olymp method for the problem of class imbalance. (3) Our proposed method could tackle the imbalance problem in object detection tasks in an extremely simple way. It could be taken as a trick to improve the imbalance problem for other researchers.

The rest of the paper is arranged as follows. Section 2 provides a review of methods to tackle the imbalance problem and we present our output layer multiplication method as well as theoretical analysis in Sect. 3. Experiments on three different imbalance problems are provided in Sect. 4, followed by which there are discussions in Sect. 5. Finally, we draw conclusions in Sect. 6.

## 2 Related Work

The method of tackling the class imbalance problem has been well researched for classical machine learning frameworks [7, 8, 16, 19, 25, 44, 53], and obtained increasing attention in deep learning networks [5, 11, 23, 26, 34, 42] in recent years. For both classical machine learning frameworks and deep learning architectures, the methods could be mainly divided into two categories: dataset-level methods and algorithmic-level methods [5].

Dataset-level methods are mainly based on the resampling technique, which includes generating more examples from the minority classes (oversampling) and removing some examples from the majority classes (undersampling) to achieve data balance between classes. They can decrease the impact of imbalanced data with a preprocessing step. For oversampling [7, 19, 48], a simple implementation is to duplicate random examples from the minority classes. Besides, some advanced methods have been proposed to generate new samples from

existing ones, such as SMOTE [7, 18], ADASYN [19] and so on. SMOTE tries to generate synthetic minority class examples by linearly interpolating adjacent class samples. As an extension of SMOTE, ADASYN could adaptively generate synthetic examples for minority classes according to their contributions. In addition, GANs-based methods [50, 51] are also applied to create synthetic examples in recent years. For undersampling [3, 46, 53], random deleting some examples from majority classes is a straightforward way to get balance and it has been proved effectively in sometimes [13]. However, an obvious weakness of this approach is that it can discard a portion of potentially useful data and it is not feasible for the situation with extremely imbalanced samples [6]. To overcome this shortage, some improved approaches have been proposed to carefully choose examples that need to be deleted. For example, Yen et al. [53] presented a cluster-based sampling approach to improve the accuracy of minority classes by selecting the representative data. Batuwita et al. [3] proposed an efficient resampling method whose main idea was to select the most informative examples by SVM from the imbalanced dataset and then used those selected examples to balance the classes.

Algorithmic-level methods try to modify existing models or propose new frameworks to tackle imbalance problem, including cost-sensitive learning [14, 27, 35, 54], ensemble learning [8, 16, 33, 44, 55], and reweighting [26, 34, 42]. Cost-sensitive learning imposes heavier costs for the misclassification of minority classes with respect to majority classes [20]. The different costs are often specified as a cost matrix  $C$ , where  $C(i, j)$  indicates the cost for predicating class  $i$  when the true class is  $j$ . Given an sample  $\mathbf{x}$ , the learning framework seeks to minimize the conditional risk  $R(i|\mathbf{x}) = \sum_j C(i, j)P(j|\mathbf{x})$ , where  $P(j|\mathbf{x})$  indicates the posterior probability of each class  $j$  for the sample  $\mathbf{x}$ . By optimizing the overall cost on the whole training data, the learned model can improve the classification performance for the imbalanced problem. For example, Zhang et al. [54] incorporated a cost matrix into a deep belief network (DBN) model and utilized an evolutionary algorithm to optimize the cost matrix. Finally the best found cost matrix was applied to the output layer of model for imbalanced classification tasks. Khan et al. [27] used a cost matrix to modify the output layer of a convolutional neural network and jointly optimized the class-dependent costs and network parameters to solve the imbalance problem in image classification.

Ensemble learning method is also a popular solution for imbalance by combing resampling methods with ensemble learning techniques [16]. They can be mainly categorized into two groups: boosting-based and bagging-based methods. Boosting-based methods embed resampling techniques into boosting algorithms. For instance, Chawla et al. [8] proposed a SMOTEBoost approach where SMOTE was used to increase the weights of misclassified minority classes by generating new samples. Similar to SMOTEBoost, Seiffert et al. [44] proposed a faster and alternative method called RUSBoost where random undersampling technique was applied to remove samples from the majority classes in each iteration of the boosting procedure. Bagging-based methods take into account the balance of the subsets when they are drawn from the original dataset in the generation step of bagging, such as OverBagging [52] and UnderBagging [4]. OverBagging carries out an oversampling process before training each weak classifier, while UnderBagging uses an undersampling technique to obtain diverse subsets instead of oversampling.

Reweighting method assigns weights to different samples by modifying the loss function. It can be represented as  $\sum_{i=1}^N w_i \ell(\mathbf{x}_i, y_i; \theta)$ , where  $\ell(\mathbf{x}_i, y_i; \theta)$  is the loss function for a sample  $\mathbf{x}_i, y_i$  with model parameter  $\theta$  and  $w_i$  is the corresponding weight coefficient. The simplest reweighting method is to reweight each example proportionally to the inverse frequency of its corresponding class. Recently, some more advanced reweighting methods have been widely applied to tackle the imbalance problem, as it can be easily embedded in

deep learning architectures. For instance, Ren et al. [42] introduced an online meta-learning method to reweight training examples, where the optimal weights are learned based on their gradient directions. It can be applied to any deep learning architecture without any extra hyper-parameters. But it needed an additional small balanced validation set to assign importance weights for each sample in every iteration. Lin et al. [34] proposed focal loss to solve the significant foreground–background class imbalance problem in tasks of object detection. The authors improved the cross-entropy loss by introducing an extra weight  $(1 - p_t)^\gamma$ , where  $p_t$  indicated the prediction probability of the ground truth class and  $\gamma$  was a tunable parameter. Then the focal loss was presented as  $-(1 - p_t)^\gamma \log(p_t)$ . It can automatically adjust the contributions of different examples according to their importance, which are inversely correlated with the value of  $p_t$ .

### 3 Method

#### 3.1 Preliminary

We assume that there is a classification task with  $C$  classes, and the amount of training sample for a mini-batch is  $N$ . Assume that  $\mathbf{x}_i$  is the  $i$ -th example and its corresponding label is  $y_i$ , where  $y_i \in \{1, \dots, C\}$ . Let  $\mathbf{o}_i = [o_{i1}, o_{i2}, \dots, o_{ij}, \dots, o_{iC}]$  be the output value of  $\mathbf{x}_i$  at the last layer. The softmax activation value can be expressed as

$$\mathbf{p}_i = [p_{i1}, p_{i2}, \dots, p_{ij}, \dots, p_{iC}], p_{ij} = \frac{e^{o_{ij}}}{\sum_{k=1}^C e^{o_{ik}}}. \tag{1}$$

If we use  $\mathbf{t}_i = [t_{i1}, t_{i2}, \dots, t_{ij}, \dots, t_{iC}]$  to indicate the ground truth, the cross-entropy loss can be expressed as

$$l_i = - \sum_{j=1}^C t_{ij} \log(p_{ij}). \tag{2}$$

For a single-label multi-class classification problem,  $\mathbf{t}_i$  is a one-hot vector, where there is only one nonzero value 1 in the place of label index  $y_i$ . Then the cross-entropy loss of  $\mathbf{x}_i$  can be rewritten as

$$l_i = - \log(p_{iy_i}) = - \log \frac{e^{o_{iy_i}}}{\sum_{j=1}^C e^{o_{ij}}}. \tag{3}$$

Conventionally, the losses from all samples or each mini-batch are averaged to represent the final softmax loss,

$$L = - \frac{1}{N} \sum_{i=1}^N l_i = - \frac{1}{N} \sum_{i=1}^N \log \frac{e^{o_{iy_i}}}{\sum_{j=1}^C e^{o_{ij}}}. \tag{4}$$

In the following sections, we will neglect the subscript  $i$  for the purpose of simplicity if it does not bring out any misunderstanding.

#### 3.2 Output Layer Multiplication

In Eq. (4), coefficient  $1/N$  can be also understood as each input sample has equal weights. This is practical effective in most of traditional classification tasks. However, if the dataset

is imbalanced, it makes the loss towards the majority class and finally influences the training performance.

So directly giving weights to different classes or different examples in the loss function is an intuitionistic and effective way to suppress the imbalance. The general weighting principle is that well-classified examples in majority classes have smaller weights than those hard examples in minority classes. Nevertheless, directly assigning weights to different classes is simple and straightforward but not sufficient. To reweight each sample separately, it would involve elaborate algorithm design or careful tuning of additional hyper parameters.

In this paper, we propose a simple approach called output layer multiplication (Olymp) to reweight different examples in an implicit way. The weights of examples are dynamically adapted according to the output value of the last layer. More formally, we multiply an amplification coefficient  $\alpha > 1$  to output of the last layer, without adding any explicit weights to the loss function. The new loss function, which will be named as Olymp loss function, is expressed as

$$L_\alpha = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{\alpha o_{y_i}}}{\sum_{j=1}^C e^{\alpha o_{ij}}}. \tag{5}$$

Intuitively, the effect of coefficient  $\alpha$  is abrupt and obscure, as it is placed in a strange position. To explore its purpose, we plot the loss curves with different  $\alpha$  in Fig. 1. The y-coordinate is the loss value, and the x-coordinate indicates the output value corresponding to the ground truth class, that is  $o_{y_i}$  in Eq. (5). On the right part of the figure, for all curves large  $o_{y_i}$  corresponds to low loss, which can be regarded as well-classified examples. With setting  $\alpha > 1$ , the relative loss value for well-classified examples is reduced with respect to hard examples. So the coefficient could automatically decrease the contributions of easy samples and make the model concentrate on hard examples. This adaptive mechanism is very helpful in the face of imbalance learning, because in the case of imbalanced datasets, the model is easily biased towards the majority classes at the early stage. As a consequence, most examples from the majority classes are easily classified and examples from the minority classes are becoming hard examples. So with the proposed output layer multiplication method, the contributions of examples from the majority or minority classes could be properly adjusted during the imbalanced training procedure.

It is worth mentioning that the proposed method is extremely simple, as it only multiplies a coefficient  $\alpha > 1$  to output of the last layer in a CNN model without any other operations. So it could be applied in off-the-shelf deep learning models without bringing out obvious computational costs.

### 3.3 Theoretical Analysis

In the above sections, we demonstrate the effects of the Olymp method with loss curves in Fig. 1. In next, we will give a theoretical analysis on the characteristics of the Olymp loss curve.

**(1) Theorem 1** *The Olymp loss is monotonically decreasing with the output value of ground truth, when  $\alpha > 1$ .*

**Proof** We rewrite the softmax activation value  $p_j$  as the function of coefficient  $\alpha$  and output value of ground truth  $o_k$  as follow,

$$f(\alpha, o_k) = \frac{e^{\alpha o_k}}{\sum_{j=1}^C e^{\alpha o_j}} = \frac{e^{\alpha o_k}}{e^{\alpha o_k} + \sum_{j=1, j \neq k}^C e^{\alpha o_j}} = \frac{1}{1 + \sum_{j=1, j \neq k}^C e^{\alpha(o_j - o_k)}}. \tag{6}$$

By computing its derivative with  $o_k$ , we get

$$\frac{\partial f(\alpha, o_k)}{\partial o_k} = -\frac{1}{\left(1 + \sum_{j=1, j \neq k}^C e^{\alpha(o_j - o_k)}\right)^2} \sum_{j=1, j \neq k}^C e^{\alpha(o_j - o_k)}(-\alpha). \tag{7}$$

For well-classified examples, there is  $o_k > o_j, j = 1, 2, \dots, C$  and  $j \neq k$ . So Eq. (7) is always positive with  $\alpha > 0$ , which indicates  $f(\alpha, o_k)$  is an increasing function with  $o_k$ . On the other hand, the cross-entropy function  $l = -\log(p)$  is always monotonically decreasing. Therefore the Olymp loss function is also monotonically decreasing with  $o_k$ , when  $\alpha > 1$ .

**(2) Theorem 2** *The Olymp loss is monotonically decreasing with  $\alpha$  for well-classified examples.*

**Proof** By computing derivative of  $f(\alpha, o_k)$  with  $\alpha$ , we can get

$$\frac{\partial f(\alpha, o_k)}{\partial \alpha} = -\frac{1}{\left(1 + \sum_{j=1, j \neq k}^C e^{\alpha(o_j - o_k)}\right)^2} \sum_{j=1, j \neq k}^C e^{\alpha(o_j - o_k)}(o_j - o_k). \tag{8}$$

For well-classified examples, there is  $o_k > o_j, j = 1, 2, \dots, C$  and  $j \neq k$ . So Eq. (8) is always positive, which indicates  $f(\alpha, o_k)$  is an increasing function with  $\alpha$ . Similar to Theorem 1, we can conclude that the Olymp loss is monotonically decreasing with  $\alpha$  for well-classified examples.

**(3) Theorem 3** *The ratio of Olymp loss to softmax loss for a sample is monotonically decreasing with the output value of ground truth, when  $\alpha > 1$ .*

**Proof** Let  $l_{Olymp}$  and  $l_{softmax}$  denote the Olymp loss and the softmax loss for a sample respectively. We denote the ratio of the two losses as  $w$ , and we have

$$w = \frac{l_{Olymp}}{l_{softmax}} = -\log\left(\frac{e^{\alpha o_k}}{\sum_{j=1}^C e^{\alpha o_j}}\right) / -\log\left(\frac{e^{o_k}}{\sum_{j=1}^C e^{o_j}}\right). \tag{9}$$

By computing its derivative with  $o_k$ , we get

$$\frac{\partial w}{\partial o_k} = \frac{\alpha \left(1 - \frac{e^{\alpha o_k}}{\sum_{j=1}^C e^{\alpha o_j}}\right) \log\left(\frac{e^{o_k}}{\sum_{j=1}^C e^{o_j}}\right) - \left(1 - \frac{e^{o_k}}{\sum_{j=1}^C e^{o_j}}\right) \log\left(\frac{e^{\alpha o_k}}{\sum_{j=1}^C e^{\alpha o_j}}\right)}{\left(\log\left(\frac{e^{o_k}}{\sum_{j=1}^C e^{o_j}}\right)\right)^2}. \tag{10}$$

From the proof that provided in the Appendix, we have that Eq. (10) is always negative when  $\alpha > 1$ , which indicates  $w$  is an decreasing function with  $o_k$ .

Theorems 1 and 2 are used to illustrate the characteristics of the Olymp loss curve, while Theorem 3 could explain the principle of our method for class imbalance problem. By reformulating Eq. (9) we can get

$$l_{Olymp} = w l_{softmax}, \tag{11}$$

which indicates that the Olymp method is equivalent to adding a modulating factor  $w$  to the basic softmax loss. Since  $w$  is monotonically decreasing with  $o_k$ , it assigns a small weight for a large  $o_k$  (easy samples) and a large weight for a small  $o_k$  (hard samples). So the Olymp method could dynamically adjust the contributions of examples from different classes during the imbalanced training procedure.

## 4 Experiments

In this section, we design three different class imbalance experiments to show the effectiveness of the proposed approach. First, we verify the performance of Olymp on binary classification tasks with imbalanced subsets of MNIST dataset [31]. Then we conduct multi-class classification tasks with different imbalanced settings based on CIFAR-10 dataset [28]. Finally, we apply Olymp to classic imbalance problem in object detection task with VOC 2007 detection dataset [15]. All the experiments are conducted with the Pytorch package [41].

### 4.1 MNIST 4-9 Binary Classification

MNIST dataset consists of 70,000  $28 \times 28$  gray images with 10 classes of handwritten digits. There are 60,000 samples for training and 10,000 samples for test. We choose two class digits from MNIST dataset to simulate a class imbalance binary classification task. Similar to [42], a total of 5000 images from class 4 and 9 are used as training set, where digit 9 is the majority class and digit 4 is the minority class. The original test images are used for test. For preprocessing, all the images are resized to  $32 \times 32$  and converted to tensors followed with normalization in the Pytorch. We utilize a typical LeNet-5 [31] architecture for experiments and compare our approach with two basic tricks for class imbalance: (1) *Proportion*, reweights each example by the inverse frequency of each class (2) *Resampling*, generates a class-balanced mini-batch for each epoch, in which training data is inverse to its class appearing probability. The network is trained using SGD for 80 epochs with a batch size of 128. The momentum is 0.9 and we set the learning rate as 0.001 throughout the training procedure.

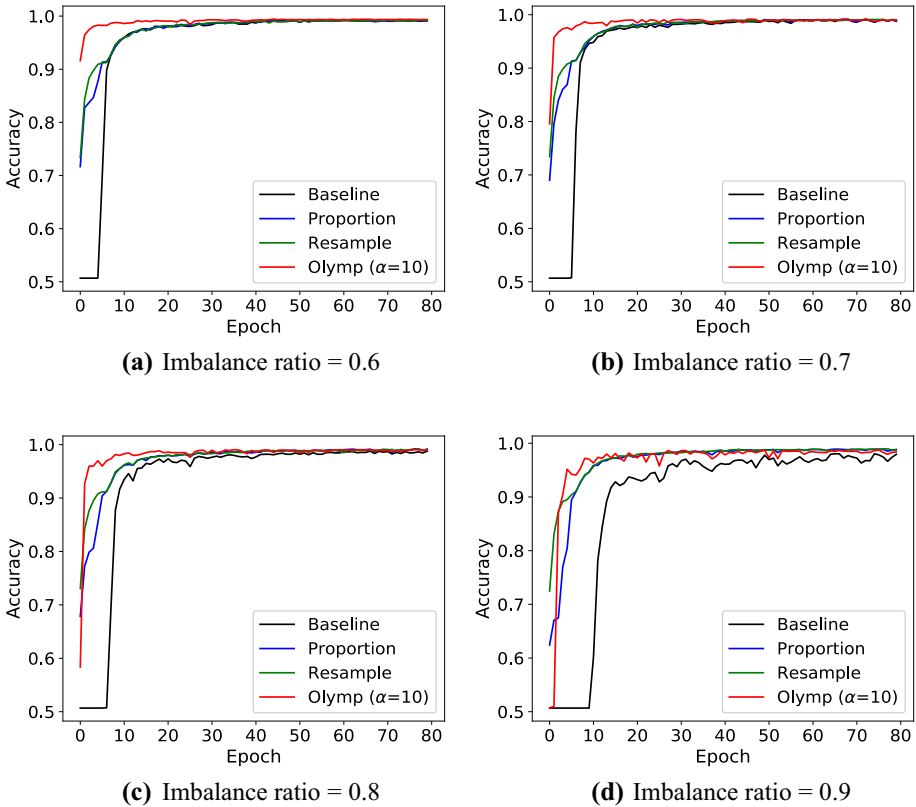
Figure 2 shows the comparative evolutions of model accuracy with some imbalance techniques under different imbalance ratios. The “Baseline” means the model is trained using the basic softmax loss without any imbalance skills. From Fig. 2a–d, we can find that Olymp ( $\alpha = 10$ ) can improve the convergence rate in all ratio settings and have better performance than the *Proportion* and *Resampling* methods except for (d).

### 4.2 CIFAR-10 Classification

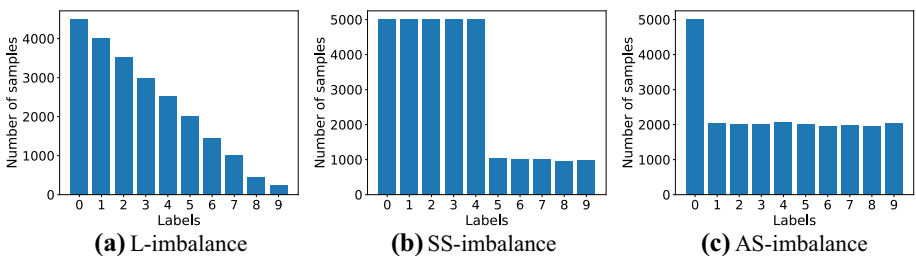
In comparison to MNIST, CIFAR-10 is a relatively more complicated dataset. There are 60,000 color images from ten classes of natural objects with a resolution of  $32 \times 32$ . The number of examples is distributed uniformly, and there are 5000 training examples and 1000 test examples for each class.

To conduct the imbalance experiment on CIFAR-10, we generate imbalanced subsets from the whole training datasets by considering three imbalanced settings [5]. (1) *Linear imbalance* (L-imbalance). In each class the number of examples is probably increasing or decreasing linearly. (2) *Symmetric step imbalance* (SS-imbalance). In majority class and minority class, the number of classes is equivalent. In addition, the number of examples within majority class is almost identical, and there are also similar numbers within the minority class. (3) *Asymmetric step imbalance* (AS-imbalance). In majority class and minority class, the number of classes is extreme unbalanced. However, the number of examples within majority classes or minority classes is almost equal. The intuitional example distributions that used in our experiment are illustrated in Fig. 3.



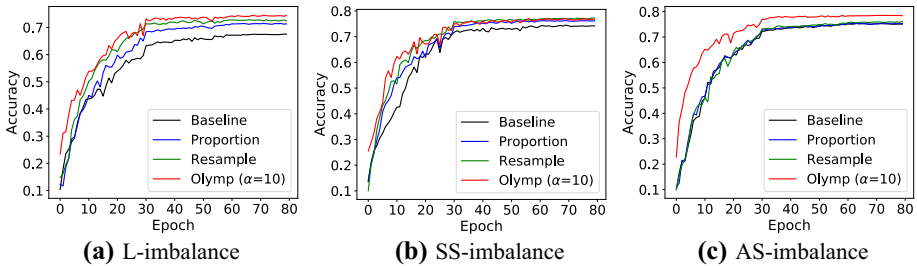


**Fig. 2** Comparative evolutions of model accuracy for some imbalance techniques with different imbalance ratios on MNIST 4-9 imbalanced dataset. The “Baseline” means the model is trained using the basic softmax loss without any imbalance skills. Best viewed in color



**Fig. 3** Three types of imbalanced settings on CIFAR-10 dataset. **a** L-imbalance, **b** SS-imbalance, **c** AS-imbalance

We use AlexNet [29] to conduct the imbalanced experiments. For dataset preprocessing, RandomCrop and RandomHorizontalFlip (Pytorch functions) are used for training data augmentation, and all images are converted to tensors followed with normalization. The network is trained using SGD for 80 epochs by setting the batch size as 128. The initial value of learning rate is 0.005 and it is decayed by 0.1 every 30 step sizes. Along the training process, all images from test set of CIFAR-10 are used to evaluate the classification performance.



**Fig. 4** Comparative evolutions of model accuracy for some imbalance techniques with three types of imbalanced settings on CIFAR-10 imbalanced classification task. The “Baseline” means the model is trained using the basic softmax loss without any imbalance skills. Best viewed in color

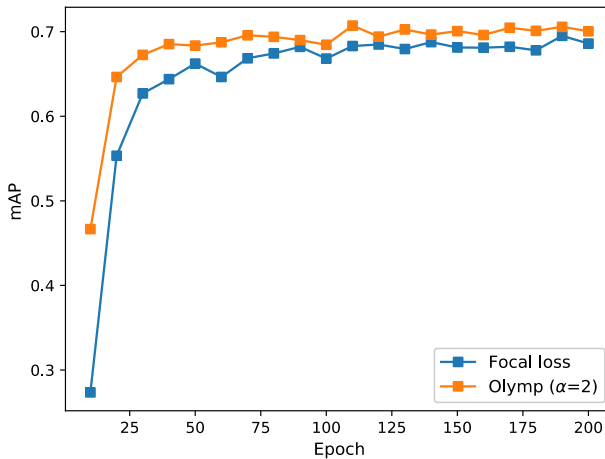
The evolutions of model accuracy with some imbalance techniques in three different imbalances setting are illustrated in Fig. 4. The “Baseline” means to train imbalanced dataset using the basic softmax loss directly. From Fig. 4a–c, we can find that Olymp improves the convergence rate and test accuracy obviously. For linear imbalance in Fig. 4a, Olymp has better performance than *Proportion* and *Resampling* methods. It achieves comparative results in symmetric step imbalance that is shown in Fig. 4b. However, in asymmetric step imbalance situation of Fig. 4c, Olymp still improves the performance obviously, while *Proportion* and *Resampling* seem to be no conspicuous improvement.

### 4.3 Imbalance in Object Detection

The Imbalance problem in object detection tasks has obtained significant attention in last several years [39]. The imbalance is mainly reflected in the extreme inequality of foreground and background examples [34]. It is inevitable because most bounding boxes belong to the background and only a few bounding boxes contain candidate objects in a given image. When dealing with the imbalanced situation with a common CNN based architecture, the process of training is dominated by the easily well-classified examples of majority classes, and finally will greatly influence detection accuracy.

To deal with the problem, various types of approaches have been proposed [39]. It includes hard sampling methods [17, 43], soft sampling methods [36, 37] and generative methods [45, 47]. Among those methods, the focal loss [34] is a straightforward and excellent approach. It dynamically assigns more weights to the hard example by  $(1 - p_t)^\gamma$ , where  $p_t$  indicates the prediction probability of the ground truth class and  $\gamma$  is a modulating factor.

By considering that our proposed Olymp method is very closely related to the focal loss. Here we applied Olymp to resolve the imbalance problem in object detection task. Similar to [34], RetinaNet is used as the one-stage detection framework, which includes a ResNet architecture and two tasks-based sub networks. Training and validation images from VOC 2007 detection dataset are used to train the whole network. The corresponding test set is used to check the detection performance with mAP indicator. The model network is optimized using SGD for 200 epochs with a batch size of 18, and the learning is 0.001 throughout the experiment. Weight decay and momentum are set as 0.0001 and 0.9 respectively during the training process. We set coefficient  $\alpha = 2$  for our proposed Olymp method, and parameters of the focal loss are set as  $\gamma = 2$ ,  $\alpha = 0.25$  [34]. The comparative results are illustrated in Fig. 5, from which we can see that Olymp has slightly better performance than the focal loss.



**Fig. 5** Comparative object detection performance by using the focal loss and Olymp ( $\alpha = 2$ ). Best viewed in color

It is worth to mention that the RetinaNet will fail quickly if the standard cross-entropy function is used without any modification. However, with our method where only one coefficient is multiplied to the output layer, the RetinaNet can be well trained to tackle the imbalance problem in above object detection task.

## 5 Discussions

### 5.1 Parameter Analysis

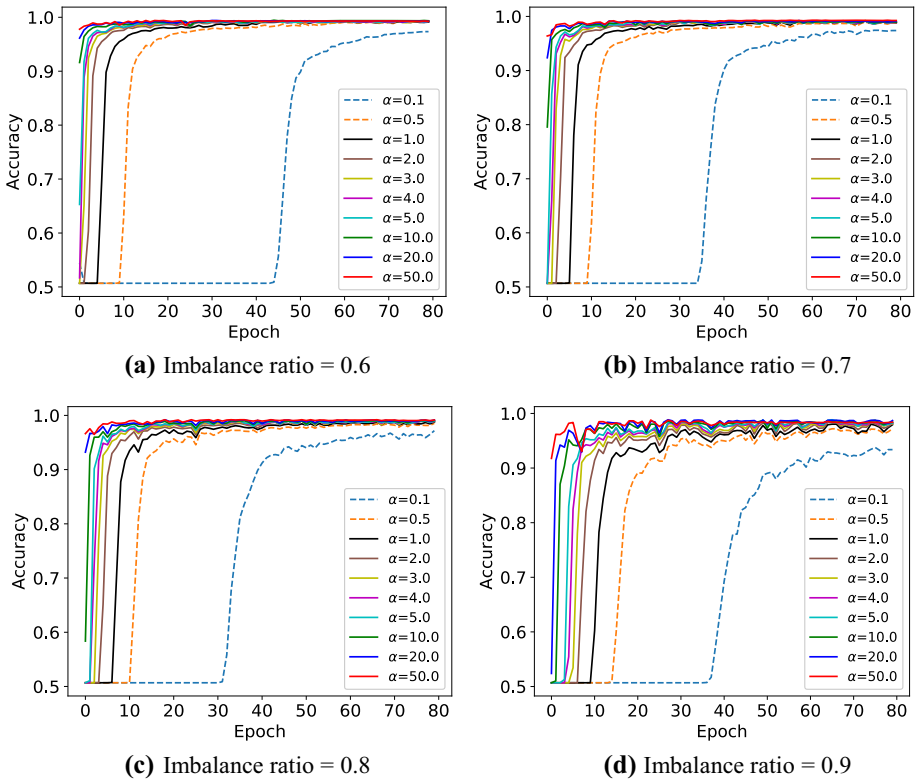
In this section, we empirically inspect the influence of the Olymp parameter  $\alpha$  on the performance of classification. For binary classification tasks on MNIST dataset, we vary  $\alpha$  from  $\{0.1, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0, 10.0, 20.0\}$  with different imbalance ratios. The corresponding experiment results are illustrated in Fig. 6.

In those figures,  $\alpha = 1.0$  means Olymp degenerates to basic softmax loss. The curves with  $\alpha < 1.0$  are negative situations that are used for checking the trend of the parameter. From the curves with  $\alpha > 1.0$  (our proposed method), the training performance is gradually improved expect when  $\alpha = 50.0$ .

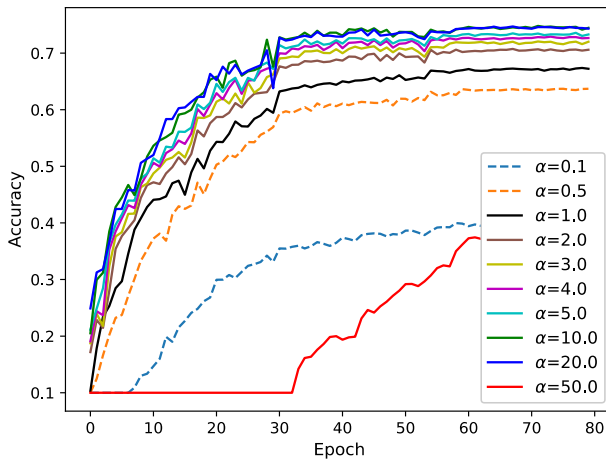
For multi-class classification on CIFAR-10, we only provide the Olymp parameter analysis on liner imbalance situation. The same parameter selection list is used and the results are plotted in Fig. 7. We can see that the performance is improved gradually from  $\alpha$  equals 1.0 to 10.0. There is no obvious improvement with  $\alpha = 20.0$ , and the performance is progressively decreasing when  $\alpha = 50.0$ .

### 5.2 Relation with the Focal Loss

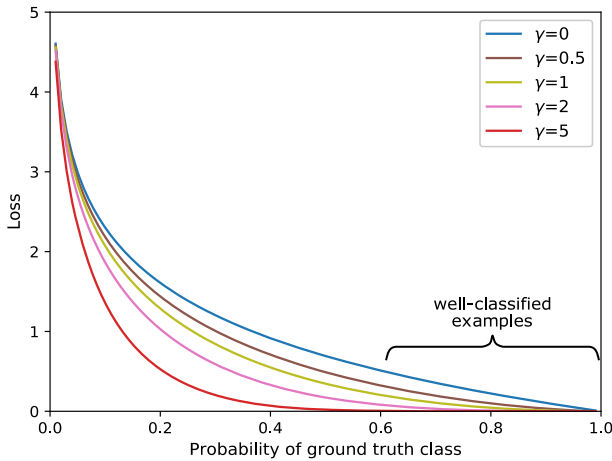
As mentioned above in Sect. 4, the proposed Olymp method is related to the focal loss. To better illustrate the similarity and difference of this two methods, we present the expressions of the basic softmax loss, the focal loss and the Olymp loss as follow,



**Fig. 6** Evolutions of model accuracy for different Olymp parameter  $\alpha$  with different imbalance ratios on MNIST 4-9 dataset. The performance is improved gradually from  $\alpha$  equals 1.0 to 20.0 expect when  $\alpha = 50.0$ . Best viewed in color



**Fig. 7** Evolutions of model accuracy for different Olymp parameter  $\alpha$  with linear imbalance on CIFAR-10 dataset. The performance is improved gradually from  $\alpha$  equals 1.0 to 10.0. Best viewed in color



**Fig. 8** Loss curves with the probability output of ground truth class in the focal loss. Setting  $\gamma > 0$  means reducing loss for well-classified examples with  $p_t > 0.5$ , putting more focus on hard-classified examples [34]. Best viewed in color

$$l_{softmax} = -\log(p_t), p_t = \frac{e^{o_{y_i}}}{\sum_{j=1}^C e^{o_j}}, \tag{12}$$

$$l_{focal} = -(1 - p_t)^\gamma \log(p_t), p_t = \frac{e^{o_{y_i}}}{\sum_{j=1}^C e^{o_j}}, \tag{13}$$

$$l_{Olymp} = -\log(p_t), p_t = \frac{e^{\alpha o_{y_i}}}{\sum_{j=1}^C e^{o_j}}. \tag{14}$$

By comparison, the focal loss has an adjustable factor for the softmax loss to reweight different examples dynamically, while the Olymp loss only has a coefficient  $\alpha$  on the output layer instead of any adjustable weight on the loss function. Although there is no explicit weight in Eq. (14), Olymp could still automatically reweight different examples as the focal loss. Because they have similar curves for well-classified examples, which can be clearly found by comparing loss curves of the focal loss in Fig. 8 with Olymp loss in Fig. 1. The similarity of curves for well-classified examples means that they have a homologous effect on imbalanced dataset training. However, we achieve this purpose in an extremely simple way and it can be easily adopted in existing CNN models with only multiplying a coefficient to the output layer.

## 6 Conclusions

In this work, we provide a concise but useful technique called output layer multiplication (Olymp) to solve the imbalance problem in softmax loss based CNN model. Olymp can be regarded as an implicit reweighting method, which tackles the imbalance by assigning different weights to different examples. In particular, by multiplying a coefficient to the output layer, the loss function automatically reduces the contributions of well-classified examples and concentrates on hard examples during the imbalanced training procedure. Compared to other methods, Olymp is very simple and can be easily applied in off-the-shelf CNN models with the slightest modification to the codes. Experiments on classifications of MNIST,

CIFAR-10 imbalanced subsets, and VOC 2007 object detection task are conducted to show the superior performance over some basic imbalance techniques.

**Acknowledgements** This research was supported by NSFC (No. 61501177, 61772455, U1713213, 41601394, 61902084), Guangzhou University’s training program for excellent new-recruited doctors (No. YB201712), Major Science and Technology Project of Precious Metal Materials Genetic Engineering in Yunnan Province (No. 2019ZE001-1, 202002AB080001), Yunnan Natural Science Funds (No. 2018FY001(-013), 2019FA-045), Yunnan University Natural Science Funds (No. 2018YDJQ004), the Project of Innovative Research Team of Yunnan Province (No. 2018HC019), Guangdong Natural Science Foundation (No. 2017A030310639), and Featured Innovation Project of Guangdong Education Department (No. 2018KTSCX174).

### Appendix

Proof for that Eq. (10) is negative with  $\alpha > 1$ .

Assume that there is a function

$$f(x) = \frac{\log(x)}{1-x}, x \in (0, 1). \tag{15}$$

By computing its derivation with  $x$ , we get

$$\frac{\partial f(x)}{\partial x} = \frac{\frac{1}{x} - 1 + \log(x)}{(1-x)^2}. \tag{16}$$

Let  $g(x)$  denote the numerator of Eq. (16), we have

$$g(x) = \frac{1}{x} - 1 + \log(x), x \in (0, 1). \tag{17}$$

By computing its derivation with  $x$ , we have

$$\frac{\partial g(x)}{\partial x} = \frac{1}{x} \left(1 - \frac{1}{x}\right). \tag{18}$$

Since Eq. (18) is always negative with  $x \in (0, 1)$ , we can conclude that  $g(x)$  is a decreasing function. The minimum value of  $g(x)$  approaches zero, as  $g(1) = 0$ . Thus,  $g(x)$  is always positive with  $x \in (0, 1)$ . It indicates that  $f(x)$  is an increasing function with  $x \in (0, 1)$ .

Let  $p_1 = e^{\alpha k} / \sum_{j=1}^C e^{\alpha_j}$ ,  $p_\alpha = e^{\alpha \alpha_k} / \sum_{j=1}^C e^{\alpha \alpha_j}$  for short. There are  $p_1, p_\alpha \in (0, 1)$  and  $p_1 < p_\alpha$  with  $\alpha > 1$ , which can be inferred from the proof of Theorem 2. By considering the monotonicity of  $f(x)$ , we have

$$\frac{\log(p_1)}{1-p_1} < \frac{\log(p_\alpha)}{1-p_\alpha}, \tag{19}$$

which can be transformed as

$$(1-p_\alpha)\log(p_1) < (1-p_1)\log(p_\alpha). \tag{20}$$

Because both sides of Eq. (20) are negative, we can obtain

$$\alpha(1 - p_a)\log(p_1) < (1 - p_1)\log(p_a), \alpha > 1. \quad (21)$$

So we can conclude that Eq. (10) is negative with  $\alpha > 1$ .

## References

- Alejo R, Garcia V P-SJ (2015) An efficient over-sampling approach based on mean square error back-propagation for dealing with the multi-class imbalance problem. *Neural Process Lett* 42:603–617
- Aurelio YS, De Almeida GM, De Castro CL, Braga AP (2019) Learning from imbalanced data sets with weighted cross-entropy function. *Neural Process Lett* 50:1937–1949
- Batuwita R, Palade V (2010) Efficient resampling methods for training support vector machines with imbalanced datasets. In: IEEE international joint conference on neural networks, pp 1–8
- Barandela R, Valdivinos RM, Sanchez JS (2003) New applications of ensembles of classifiers. *Pattern Anal Appl* 6(3):245–256
- Buda M, Maki A, Mazurowski MA (2018) A systematic study of the class imbalance problem in convolutional neural networks. *Neural Netw.* <https://doi.org/10.1016/j.neunet.2018.07.011>
- Cao K, Wei C, Gaidon A, Arechiga N, Ma T (2019) Learning imbalanced datasets with label-distribution-aware margin loss. In: *Neural information processing systems*, pp 1567–1578
- Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res* 16:321–357
- Chawla NV, Lazarevic A, Hall LO, Bowyer KW (2003) SMOTEBoost: improving prediction of the minority class in boosting. In: *European conference on principles and practice of knowledge discovery in databases*, pp 107–119
- Chen L, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018) DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans Pattern Anal Mach Intell* 40(4):834–848
- Chen Q, Huang J, Feris R, Brown LM, Dong J, Yan S (2018) Deep domain adaptation for describing people based on fine-grained clothing attributes. In: *IEEE Conference on computer vision and pattern recognition*, pp 5315–5325
- Cui Y, Jia M, Lin TY, Song Y, Belongie S (2019) Class-balanced loss based on effective number of samples. In: *IEEE conference on computer vision and pattern recognition*
- Dong Q, Gong S, Zhu X (2019) Imbalanced deep learning by minority class incremental rectification. *IEEE Trans Pattern Anal Mach Intell* 41(6):1367–1381
- Drummond C, Holte RC (2003) C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In: *ICML workshop on learning from imbalanced data II*, pp 1–8
- Elkan C (2001) The foundations of cost-sensitive learning. In: *International joint conference on artificial intelligence*, pp 973–978
- Everingham M, Gool LV, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. *Int J Comput Vision* 88(2):303–338
- Galar M, Fernandez A, Barrenechea E, Bustince H, Herrera F (2012) A review on ensembles for the class imbalance problem: gaggling-, boosting-, and hybrid-based approaches. *IEEE Trans Syst Man Cybern Part C* 42(4):463–484
- Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: *IEEE conference on computer vision and pattern recognition*, pp 580–587
- Guo S, Liu Y, Chen R, Sun X, Wang X (2019) Improved SMOTE algorithm to deal with imbalanced activity classes in smart homes. *Neural Process Lett* 50:1503–1526
- He H, Bai Y, Garcia EA, Li S (2008) ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: *IEEE international joint conference on neural networks*, pp 1322–1328
- He H, Garcia EA (2009) Learning from imbalanced data. *IEEE Trans Knowl Data Eng* 21(9):1263–1284
- He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *IEEE conference on computer vision and pattern recognition*, pp 770–778
- Hensman P, Masko D (2015) The impact of imbalanced training data for convolutional neural networks. Degree project, KTH Royal Institute of Technology
- Huang C, Li Y, Loy CC, Tang X (2016) Learning deep representation for imbalanced classification. In: *IEEE conference on computer vision and pattern recognition*, pp 5375–5384

24. Huang C, Li Y, Loy CC, Tang X (2018) Deep imbalanced learning for face recognition and attribute prediction. arXiv: 1806.00194
25. Huang K, Zhang R, Yin XC (2015) Learning imbalanced classifiers locally and globally with one-side probability machine. *Neural Process Lett* 41:311–323
26. Katharopoulos A, Fleuret F (2018) Not all samples are created equal: deep learning with importance sampling. In: *International conference on machine learning*, pp 2525–2534
27. Khan SH, Hayat M, Bennamoun M, Sohel FA, Togneri R (2018) Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE Trans Neural Netw Learn Syst* 29(8):3573–3587
28. Krizhevsky A, Hinton GE (2009) Learning multiple layers of features from tiny images. Ms. thesis, University of Toronto
29. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: *Neural information processing systems*, pp 1097–1105
30. Kumar N, Berg A, Belhumeur PN, Nayar S (2011) Describable visual attributes for face verification and image search. *IEEE Trans Pattern Anal Mach Intell* 33(10):1962–1977
31. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. *Proc IEEE* 86(11):2278–2324
32. Li S, Deng W (2016) Real world expression recognition: a highly imbalanced detection problem. In: *IEEE international conference on biometrics*, pp 1–6
33. Lim P, Goh CK, Tan KC (2017) Evolutionary cluster-based synthetic oversampling ensemble (eco-ensemble) for imbalance learning. *IEEE Trans Cybern* 47(9):2850–2861
34. Lin TY, Goyal P, Girshick R, He K, Dallar P (2014) Focal loss for dense object detection. In: *IEEE international conference on computer vision*, pp 2999–3007
35. Ling CX, Sheng VS (2017) Cost-sensitive learning. *Encyclopedia of machine learning and data mining*. Springer, Boston
36. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2018) SSD: single shot multibox detector. In: *European conference on computer vision*, pp 21–37
37. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: *IEEE conference on computer vision and recognition*, pp 779–788
38. Shelhamer E, Long J, Darrell T (2017) Fully Convolutional networks for semantic segmentation. *IEEE Trans Pattern Anal Mach Intell* 39(4):640–651
39. Oksuz K, Cam BC, Kalkan S, Akbas E (2019) Imbalance problems in object detection: a review. arXiv: 1909.00169
40. Pang J, Chen K, Shi J, Feng H, Ouyang W, Lin D (2019) Libra R-CNN: towards balanced learning for object detection. arXiv: 1904.02701
41. Pytorch, <https://pytorch.org/>
42. Ren M, Zeng W, Yang B, Urtasun R (2018) Learning to reweight examples for robust deep learning. In: *International conference on machine learning*, pp 4334–4343
43. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 39(6):1137–1149
44. Seiffert C, Khoshgoftaar TM, Hulse JV, Napolitano A (2010) RUSBoost: a hybrid approach to alleviating class imbalance. *IEEE Trans Syst Man Cybern Part A Syst Humans* 40(1):185–197
45. Tripathi S, Chandra S, Agrawal A, Tyagi A, Rehg JM, Chari V (2019) Learning to generate synthetic data via compositing. In: *IEEE conference on computer vision and pattern recognition*, pp 461–470
46. Vuttipittayamongkol P, Elyan E (2020) Neighbourhood-based undersampling approach for handling imbalanced and overlapped data. *Inf Sci* 509:47–70
47. Wang X, Shrivastava A, Gupta A (2017) A-fast-rcnn: hard positive generation via adversary for object detection. In: *IEEE conference on computer vision and pattern recognition*, pp 3039–3048
48. Wang J, Xu M, Wang H, Zhang J (2006) Classification of imbalanced data by using the SMOTE algorithm and locally linear embedding. In: *International conference on signal processing*, <https://doi.org/10.1109/icsp.2006.345752>
49. Wang P, Su F, Zhao Z, Guo Y, Zhao Y, Zhuang B (2019) Deep class-skewed learning for face recognition. *Neurocomputing* 363:35–45
50. Wang Q, Gao J, Lin W, Yuan Y (2019) Learning from synthetic data for crowd counting in the wild. In: *IEEE conference on computer vision and pattern recognition*, pp 8190–8199
51. Wang Q, Gao J, Li X (2019) Weakly supervised adversarial domain adaptation for semantic segmentation in urban scenes. *IEEE Trans Image Process* 28(9):4376–4386
52. Wang S, Yao X (2009) Diversity analysis on imbalanced data sets by using ensemble models. In: *IEEE symposium on computational intelligence and data mining*, pp 324–331
53. Yen SJ, Lee YS (2009) Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Syst Appl* 36:5718–5727



54. Zhang C, Tan KC, Ren R (2016) Training cost-sensitive deep belief networks on imbalance data problems. In: International joint conference on neural networks, pp 4362–4367
55. Zhang L, Zhang Q, Zhang L, Tao D, Huang X, Du B (2015) Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding. *Pattern Recogn* 48:3102–3112
56. Zheng L, Shen L, Tian L, Wang S, Wang J, Tian Q (2015) Scalable person re-identification: a benchmark. In: IEEE international conference on computer vision, pp 1116–1124

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.