UNIVERSITY OF CALIFORNIA
SANTA CRUZ

REPOSITORY FOR LONG-TAILED LEARNING WITH NOISY LABELS APPROACHES

A dissertation submitted in partial satisfaction of the
requirements for the degree of

MASTER OF COMPUTER SCIENCE & ENGINEERING

in

MACHINE LEARNING

by

Jinghao Shen

June 2023

The Dissertation of Jinghao Shen
is approved:

Professor Yang Liu, Chair

Professor Leilani Gilpin

Dean Alexander Wolf

Table of Contents

List of Figures

List of Tables

Abstract

Repository for Long-tailed Learning with Noisy Labels Approaches

by

Jinghao Shen

Real-world data inevitably has data bias. The data bias may come from two aspects: 1) class imbalance. I.e., a few classes account for most of the data, while most classes are under-represented. We often refer to this scenario as long-tail distribution; 2) label noise. I.e., some labels of the data are wrong annotated. For example, a ”dog” image may be wrongly annotated as ”wolf” by a person accidentally. In the literature on weakly supervised learning, ”long-tail learning” and ”learning with noisy labels” are two popular research areas. Researchers have devised many efficient solutions for dealing with these data biases. However, most of the works study these two data biases separately. They either assume the data is clean with long-tail distribution or assume the data is noisy but without a long-tail effect. There are fewer efforts to study long-tail and noisy labels together. Note that in practice, these two data biases may exist in real-world data at the same time. Thus it motivates us to understand how it affects the learning process when long-tail and noisy labels both exist in the data. In this project, we analyze several recently proposed methods for long-tailed with noisy label learning and build a codebase as the baseline for future research. The code is available at `https://github.com/jshen1s1/2023-MS-Project-Long-tailed-learning-with-noisy-labels`

Acknowledgments

I want to thank my family members for supporting me. Technical support from professor Yang Liu and my friend Hao Cheng.

Part I

First Part

Chapter 1

Introduction

Deep neural networks (DNNs) have been used in a wide range of applications and exhibited incredible performance in various machine learning tasks, such as natural language processing, information retrieval, and computer vision. The key to the remarkable success of DNNs often relies on the availability of large-scale datasets, high-performance-computing GPU, and the composition of many layers of vectors as representations [2]. However, data collection and annotation inevitably introduce label noise, and cleaning up the corrupted labels is extremely expensive and time-consuming. The existence of label noise can weaken the true correlation between features and labels as well as introduce artificial correlation patterns. Thus, mitigating the effects of noisy labels becomes a critical issue that needs careful treatment. Further, in real-world applications, data samples typically tend to follow a long-tail distribution where a few head (dominant) classes claim most of the examples, while most tail (minor) classes are associated with relatively few examples [30, 38, 27, 15, 31]. Due to this imbalanced sample distribution, training DNNs on such data is

challenging and often results in an undesirable bias toward dominant classes.

The presence of noisy labels and long-tailed distribution can greatly reduce the accuracy of models on the clean test data because DNNs can easily over-fit noisy labels and head classes [48, 52]. Popular regulation techniques, such as sample selection [13], robust loss function [53], and module improvement [17] have been applied to tackle specific problems in DNNs training, but they can not completely overcome the overfitting problem raised by long-tailed learning with noisy labels. Some recent works [42, 18] have demonstrated that combining approaches from 'long-tail learning' and 'learning with noisy labels' can achieve certain performance improvements. However, the methods are not theoretically verified and do not lead to a deeper understanding of the problem. Additionally, it is still unknown which bias can do more harm to DNNs training and which direction we should take to tackle this difficulty. Hence, an empirical analysis of the current methods is essential for future research.

Massive studies have been conducted to address the problem of noisy labels and long-tailed distribution separately but poorly studied the conditions when both biases are introduced to the dataset [27, 52, 36, 12]. The standard Empirical risk (ERR) minimization will not work well as the class distribution is highly skewed. Therefore, it remains a challenge to train DNNs on dataset with long-tailed distribution with noisy labels. There is a lack of systematic study and discussion of this existing problem. Thus, the goals of this project are to construct a codebase that consists of three categories (i.e., noisy labels, long-tailed learning, and long-tailed distribution with noisy labels) for future research in these areas and to analyze several state-of-the-art methods.

We summarize the key distribution of this project as follows:

- We build a code repository with commonly used approaches to long-tailed learning with noisy labels. This code repository has carefully designed dataset which supports the generation of many noise types (pariflip, symmetric, asymmetric, instance-dependent) and long-tail types (step, exponential) and incorporates many popular approaches from both long-tail learning (focal loss, class-balance loss, etc.) and leaning with noisy labels (gce, co-teaching loss, etc.).

- We review the state-of-the-art in handling long-tailed distribution with noisy labels and provide an in-depth comparison of current long-tailed learning with noisy label studies.

Chapter 2

Preliminaries

In this section, we review the basic conceptional methods and theoretical foundations that underlay deep learning robust against noisy labels and long-tailed distribution. Classification is a task that requires the use of machine learning algorithms to learn how to map an input feature to a class label.

In this project, we consider a classification problem on a set of N training samples denoted by $D := \{(x_n, y_n)\}_{n\in[N]}$, where $[N] := \{1, 2, \cdots, N\}$ is the set of the indices of these samples and each sample x_n has a corresponding class label y_n. Samples (x_n, y_n) are drawn according to random variables (X, Y) from a joint distribution $\mathcal{D}$. We assume that the label noise is corrupted by a noise transition matrix T, where each element T_{ij} represents the probability of mislabeling the ground-truth label $y = j$ to the noisy label $\tilde{y} = i$, i.e., $T_{ij} = \mathbb{P}(\widetilde{Y} = i|Y = j)$. The corresponding noisy dataset is denoted by $\widetilde{D} := \{(x_n, \tilde{y}_n)\}_{n\in[N]}$. We consider four types of label noise with a noise rate of $\gamma \in [0, 1]$ as described in [29, 39, 43, 36]. *Symmetric noise* is where

real labels are flipped to other labels with equal probability. On the other hand, *asymmetric noise* means real labels are flipped to a particular label based on fixed rules. *Pairflip noise* is where a real label is flipped into its adjacent classes. For more realistic noise, *instance noise* is where the probability of mislabeling an instance depends on both the data features and class labels.

The setting for long-tail learning is where the distribution $\mathbb{P}(Y)$ follows a long-tailed class distribution so that many labels have a very low probability of occurrence. The imbalance ratio ρ donates the ratio between sample sizes of the most frequent and least frequent class, i.e., $\rho = \max_y \mathbb{P}(y)/\min_y \mathbb{P}(y)$. It is assumed that the long-tailed imbalance follows an *exponential* decay in sample sizes, while for the *step imbalance* setting, all minority classes have the same sample size as do all dominant classes [4]. The deep neural networks $f(\cdot;\boldsymbol{\theta}) : \mathcal{X} \to \mathbb{R}^N$ learning from the long-tailed and noisy dataset is parameterized by the model parameter $\boldsymbol{\theta}$.

2.1 Noisy labels

2.1.1 Sample selection:

Sample selection can be viewed as an implicit re-weighting scheme that gives the selected samples with weight of 1 and unselected samples with weight of 0. The motivation for sample selection in learning with noisy labels is that the samples with noisy labels tend to have larger losses at the beginning of training deep neural networks. Mathematically, let l_i be the cross-entropy loss for sample i, then the sample selection scheme can be represented as:

$$L = \sum_{i=1}^{N} \mathbb{1}(l_i < \text{threshold}) \cdot l_i \tag{2.1}$$

where $\mathbb{1}$ is the indicator function that takes value 1 if the condition is satisfied and 0 otherwise. The threshold can be pre-defined or dynamically adjusted. A line of works adopts this sample selection scheme for learning with noisy labels including co-teaching [13], co-teaching+ [47], Dividemix [23], etc.

2.1.2 Robust loss design:

A very popular direction for learning with noisy labels is to design a loss function that is statistically robust to label noise. For example, we wish to design a loss function l that satisfy

$$\min_f \mathbb{E}_{X,Y}[l(f(X), Y)] \Longleftrightarrow \min_f \mathbb{E}_{X,\widetilde{Y}}[l(f(X), \widetilde{Y})] \tag{2.2}$$

where $\widetilde{Y}$ is the noisy labels. Equation (2.2) suggests that minimizing l with respect to the model function f on clean data is equivalent to minimizing l on the noisy data. A line of works design various loss functions that satisfy or approximately satisfy this condition such as MAE [11], GCE [53], SCE [40], APL [28]. For example, the loss formulation of GCE can be represented as:

$$l_{gce}(x, \tilde{y}) = \frac{1 - f_{\tilde{y}}(x)^q}{q} \tag{2.3}$$

where $q \in [0, 1]$ is a hyper-parameter. When $q = 0$, l_{gce} is MAE loss and when $q = 1$, l_{gce} is cross entropy loss.

However, the condition in Equation (2.2) may not be the best solution because the performance of designed loss on clean data may be worse than cross entropy. Another strong condition

is:

$$\min_f \mathbb{E}_{X,Y}[CE(f(X), Y)] \Longleftrightarrow \min_f \mathbb{E}_{X,\widetilde{Y}}[l(f(X), \widetilde{Y})] \tag{2.4}$$

I.e., minimizing l with respect to the model function f on noisy data is equivalent to minimizing CE on the clean data. A representative work to satisfy this condition is peer loss [26]. The formulation of peer loss can be represented as:

$$l_{peer}(x_i, \tilde{y_i}) = \text{CE}(f(x_i), \tilde{y_i}) - \alpha \text{CE}(f(x_{p,i}), \tilde{y_{p,i}}) \tag{2.5}$$

where α is a hyper-parameter and $x_{p,i}$ and $y_{p,i}$ are randomly sampled from the dataset.

2.2 Long-tailed learning

2.2.1 Focal loss for long-tail learning

Focal loss [24] was first proposed for solving the imbalanced object distribution in images. Later it was later used in the pure classification problem with long-tail distribution. Consider a binary classification problem where $y \in \{0, 1\}$. Define $p \in [0, 1]$ be the model's estimated probability for the class with label $y = 1$. Let $p_t = p$ if $y = 1$ or $p_t = 1 - p$ if $y = 0$. Then the focal loss can be represented as:

$$\text{FL}(p_t) = -\alpha_t (1 - p_t)^\gamma \log(p_t) \tag{2.6}$$

where α_t and γ are hyper-parameters. From Equation (2.6), focal loss (FL) is doing re-weighting for each training sample with the weight $\alpha_t(1 - p_t)^\gamma$. Intuitively, FL reduces the loss contribution

from easy examples and extends the range in which an example receives low loss. Thus, samples in the tail classes tend to have larger weights than samples in the head classes.

2.2.2 Class-balanced loss for long-tail learning:

class-balanced loss [9] proposes a new weighting scheme by considering the data overlapping. The authors find that as the number of samples increases, the additional benefit of a newly added data point will diminish. Thus, from this observation, the authors calculate the effective number of samples E_n for each class by using the random covering theory. E_n can be represented as:

$$E_n = \frac{1-\beta^n}{1-\beta}, \tag{2.7}$$

where $\beta = \frac{N-1}{N}$, N is the number of samples in a whole dataset and n is the number of samples in a specific class. After calculating E_n, we can re-weighting many losses such as cross entropy or focal loss to deal with long-tail learning. For example, when using E_n to further re-weight focal loss, the class-balanced focal loss can be presented as:

$$\text{CB}_{\text{focal}}(p_t, y) = \frac{1}{E_{n_y}} * \text{focal loss} = -\frac{1-\beta}{1-\beta^{n_y}}\alpha_t(1-p_t)^{\gamma}\log(p_t) \tag{2.8}$$

where n_y denotes the number of samples in class y. Experiments show that CB_{focal} achieves better performance than only using focal loss for dealing with long-tail data.

2.2.3 Logits-adjustment for long-tail learning

Both focal loss and class-balanced loss are effective in long-tail learning, but it is still not clear what is the clear objective function we are seeking for long-tail learning. For this purpose, Logits-adjustment [30] formulates the long-tail learning in terms of balanced error problem:

$$\text{BER}(f) = \frac{1}{L} \sum_{y \in [L]} \mathbb{P}_{x|y}(y \notin \arg\max_{y'} f_{y'}(x)) \tag{2.9}$$

This can be seen as implicitly using a balanced class-probability function $\mathbb{P}^{\text{bal}}(y|x) \propto \frac{1}{L} \cdot \mathbb{P}(x|y)$, as opposed to the native $\mathbb{P}(y|x) \propto \mathbb{P}(y) \cdot \mathbb{P}(x|y)$, that is employed in the vanilla misclassification error. Under this formulation, the authors propose a loss function that can achieve Baye Optimal solution for Equation (2.9). The loss can be expressed as:

$$l(y, f(x)) = -\log \frac{e^{f_y(x) + \tau \cdot \log \pi_y}}{\sum_{y' \in [L]} e^{f_{y'}(x) + \tau \cdot \log \pi_{y'}}} \tag{2.10}$$

where π_y is the sampling frequency in the dataset for label y. For $\tau = 0$, the loss degenerates to the vanilla cross entropy loss. For $\tau = 1$, the loss returns Bayes optimal solution for Equation (2.9).

Chapter 3

Previous Work

In the previous work, we have worked on learning a classifier $\widetilde{D} := \{(x_n, \tilde{y}_n)\}_{n\in[N]}$ that is consistent with Bayes optimal classifier for balanced empirical risk on the clean dataset such that the classifier can well alleviate the noisy label and long-tail effects from the biased dataset. We refer to such a classifier as Fisher consistent classifier [30].

To achieve fisher consistency, one common approach is to estimate $\mathbb{P}(Y)$[30, 8]. However, since the dataset contains noisy labels, it is non-trivial to estimate $\mathbb{P}(Y)$ from $\mathbb{P}(\widetilde{Y})$. An alternative approach is to select small loss samples as clean samples at the beginning of the training stage when using cross entropy to train the model. To make the sample selection more efficient, we proposed to use the confidence regularizer $\ell_{\text{CR}}(f)$ from [7] along with vanilla Cross Entropy (CE) loss to ease the sample selection.

$$\ell_{\text{CR}}(f(x_n)) := -\beta \cdot \mathbb{E}_{\mathcal{D}_{\widetilde{Y}|\widetilde{D}}}[\ell(f(x_n), \widetilde{Y})], \tag{3.1}$$

After we sieve out the clean samples, we also need to estimate $\mathbb{P}(Y)$ to satisfy fisher consistency. Then, we make the logits adjustment on the model output using estimated $\mathbb{P}(Y)$. Mathematically, Suppose the dataset has K classes. let $f(x)$ be the model output (before softmax layer) on the sample x where $f(x)$ is a K dimensional vector representing the score of each class. Then, our method can be represented as:

$$
\begin{aligned}
&\min_{f' \in \mathcal{F}, \boldsymbol{v} \in [0,1]^N} \sum_{n \in [N]} v_n \left[\ell(f'(x_n), \tilde{y}_n) + \ell_{\text{CR}}(f'(x_n)) - \alpha_n \right] \\
&\quad \text{s.t.} \quad f'(x_n) = f(x_n) + \tau \log \pi^Y \\
&\qquad \ell_{\text{CR}}(f'(x_n)) := -\beta \mathbb{E}_{\mathcal{D}_{\widetilde{Y}}} \ell(f'(x_n), \widetilde{Y}) \\
&\qquad \alpha_n := \frac{1}{K} \sum_{\tilde{y} \in [K]} \ell(\bar{f}'(x_n), \tilde{y}) + \ell_{\text{CR}}(\bar{f}'(x_n)) \\
&\qquad \pi^Y := [\frac{\sum_{n \in [N]} (\tilde{y}_n = 1 | v_n = 1)}{\sum_{n \in [N]} v_n = 1}, \cdots, \frac{\sum_{n \in [N]} (\tilde{y}_n = K | v_n = 1)}{\sum_{n \in [N]} v_n = 1}]
\end{aligned}
\tag{3.2}
$$

Where τ and β are hyper-parameters. π^Y is the estimated label frequency on selected clean samples. f' is the adjustment of f, following logits adjustment in [30]. $\bar{f}'$ is a copy of f' and does not contribute to the back-propagation. $\mathcal{F}$ is the search space of f'. For $v_n = 1$, sample n is clean. For $v_n = 0$, sample n contains noise. α_n controls which example should be sieved out. ℓ is chosen to be Cross Entropy Loss.

We evaluated our method on CIFAR-10 [21], which has 50000 samples for training and 10000 samples for evaluation. We used ResNet34 for model training. We performed the experiment on CIFAR-10 with symmetric label noise. The number of samples for each class decays

Table 3.1: Test accuracy for each method on CIFAR-10. The imbalance ratio is 10.

	CE	CB	focal loss	logits adjustment	ROLT	our approach
symmetric 0.1	79.32	78.03	78.32	79.55	86.24	**87.29**
symmetric 0.2	77.17	74.04	67.93	73.44	85.49	**85.54**
symmetric 0.3	70.59	66.46	63.68	70.99	**84.11**	83.4

exponentially with a fixed imbalance ratio.

Table 3.1 shows the empirical results on CIFAR10. We compare our approach with Cross Entropy loss (CE), class-balanced loss (CB) [9], Focal loss [24], logits adjustment [30], and ROLT [42]. CE, CB, focal loss, and logits adjustment loss are popular losses in long-tail learning. ROLT is a recent approach dealing with long-tail learning with noisy labels. We train DNN for 200 epochs with an initial learning rate of 0.1 and decayed by 0.1 at 60 and 120 epochs. The optimizer is vanilla SGD.

From Table 3.1, we can see our approach is very comparable to ROLT. It is worth noting that we do not spend much time tuning the hyper-parameters. Unlike ROLT, our approach does not employ semi-supervised learning on noisy samples. The result motivates us to compare our approach with more different methods. This is also the reason for this project. By constructing such a repository, we hope to obtain a further understanding of this problem, the performance of other standard approaches to this problem, and as a baseline for future research.

Part II

Second Part

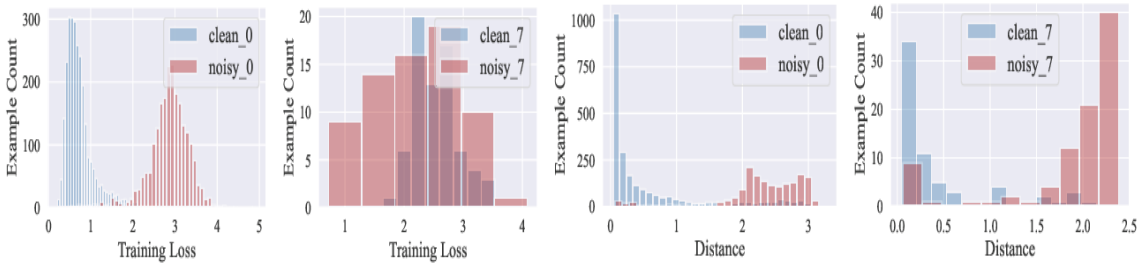
Chapter 4

Long-tailed Learning with Noisy Labels Methods

Methods for long-tailed learning with noisy labels usually utilize multiple techniques in biased learning. In this section, we will introduce some widely used techniques that are deployed in different approaches and provide insight into how these techniques are combined together. This analysis can help us better understand the efficiency of these techniques against the bias of imbalanced and noisy labels. Other not mentioned methods are listed in README file in the code repository.

4.1 Sample Selection and Re-balancing

Sample Selection, as a mainstream technique in biased data learning, seeks to assign different weights to different objects to exploit noisy data from the clean data in noisy label learning.



(a) Loss for head class (b) Loss for tail class (c) Distance for head class (d) Distance for tail class

Figure 4.1: (a-b) Training losses for examples of head class and tail class, respectively. (c-d) Distance distribution between examples and their class prototype for head and tail classes, respectively.

Typical noise detection employs the small-loss trick. However, with the presence of long-tailed distribution, sample loss belonging to the tail classes can also be large, as shown in 4.1(a-b) [42]. To solve this issue, several methods have been proposed.

• RoLT [42] proposed to use class prototypes for noise detection. They found that clean examples are more likely to be clustered around their prototypes, as demonstrated in 4.1(c-d). They fit a mixture of two Gaussian models [33] to estimate parameters from distance statistics for class-independent noise detection. For each class k, the prototype is computed as

$$\mathbf{c}_k \leftarrow \text{Normalize}(\frac{1}{|\mathcal{D}_k|} \sum_{x_i \in \mathcal{D}_k} f_{\boldsymbol{\theta}}(x_i)), \mathcal{D}_k = \{x_i | y_i = k\}, \tag{4.1}$$

where $f_{\boldsymbol{\theta}}(x_i)$ donates the extracted features of x. Then, the distances between $\mathbf{c}_k$ and examples of class k are calculated as

$$dist(\mathbf{c}_k, x_i) = \|\mathbf{c}_k - f_{\boldsymbol{\theta}}(x_i)\|_2^2, \forall x_i \in \mathcal{D}_k. \tag{4.2}$$

After fitting the Gaussian Mixture Model (GMM), i.e., $\max \sum_{i=1}^{|\mathcal{D}_k|} log(\sum_{j=1}^{2} \phi_j \mathbb{P}(d_i|\mu_j, \sigma_j)$,

where $d_i = dist(\mathbf{c}_k, x_i)$, the dataset is divided into clean set $\mathcal{X}_k$ and noisy set $\mathcal{S}_k$.

• CNLCU [44] proposed to use interval estimation instead of point estimation of losses for sample selection. It follows the co-teaching methods where the mentor network provides the student network with examples whose labels are likely to be true labels. The confidence interval estimation reduces not only the uncertainty of small-loss samples but also the uncertainty of large-loss samples. They introduced two robust mean estimators w.r.t the truncation level. Soft truncation exploits the non-decreasing *Taylor expansion of the exponential function* ψ as the influence function of the selection criterion.

$$\psi(X) = \log(1 + X + X^2/2), X \geq 0 \tag{4.3}$$

The robust mean estimator is calculated as follows:

$$\tilde{\mu_s} = \frac{1}{t}\sum_{i=1}^{t}\psi(\ell_i). \tag{4.4}$$

Hard truncation first exploits the K-nearest neighbor algorithm (KNN) to remove underlying outliers in training losses L_t, turning it to L_{t-t_o}. Then, the mean estimator can be calculated as follows:

$$\tilde{\mu_h} = \frac{1}{t - t_o}\sum_{\ell_i \in L_{t-t_o}} \ell_i. \tag{4.5}$$

• Sample selection with balanced loss [50] suggests using a regularization term $\mathcal{L}_{reg}$ during the warm up stage to prevent the influence of the long-tailed distribution.

$$\mathcal{L}_{reg} = \sum_{i=1}^{N}\frac{n_N}{n_i}\pi_i \log(\pi_i / \frac{1}{N}\sum_{j=1}^{N}\sum_{(x,\tilde{y})\in\tilde{D}_j}\frac{1}{n_j}p^i_{model}(x)), \tag{4.6}$$

where $p^i_{model}(x) = \frac{\exp f_i(x)}{\sum_{j=1}^{N} \exp f_j(x)}$ and $\pi_i = \frac{1}{N}$. Class-balanced weight $\frac{1}{n_j}$ is assigned to reduce the influence of head classes, and $\frac{n_N}{n_i}$ is used to strengthen tail classes. After warming up, the small-loss criterion and a two-component (g_1 as clean distribution, g_0 as noisy distribution) GMM are applied to select clean samples. The clean probability of sample $(x, \tilde{y})$ is given by $P(g_1|L(x, \tilde{y})$, where g_1 is the Gaussian component with a smaller loss.

• LNOR [10] proposed to employ the cross-augmentation matching to detect noisy labels. The classification model is built up with dual-path normalization, which processes weakly and strongly augmented images. The inference matching between weakly and strongly augmented images aims to split mislabeled samples and clean samples from tail classes. The loss criterion is formulated as:

$$\ell(x_i, y_i) = \ell^c(x_i, y_i) + \ell^c(\hat{x}_i, y_i) + \alpha\ell^c(\hat{x}_i, y'_i), \tag{4.7}$$

$$\ell^c(x_i, y_i) = -\mathrm{y}_i^T log(f_{\boldsymbol{\theta}}(x_i)), \tag{4.8}$$

where α is a weight constant, x_i and $\hat{x}_i$ denotes the weakly and strongly augmented samples respectively, y'_i denotes the most confident class prediction on x_i, and y_i is the one-hot format of y_i. Small-loss trick with threshold ϵ is used to obtain a clean dataset $\mathbb{C}$. The overall object function on $\mathbb{C}$ is

$$\mathcal{L}_c = \sum_{(x_i, y_i) \in \mathbb{C}} \ell(x_i, y_i). \tag{4.9}$$

• MFRW-MES [35] proposed using meta-learning to train an auxiliary advisor network that exploits an attention mechanism to concentrate on the useful parts of visual information. The concentrated information can then contribute to improving the overall generalization capacity of

the network regardless of the presence of noisy labels and long-tailed distribution. The process is referenced as Meta Feature Re-Weighting (MFRW). It is achieved by calculating a weighting mask W_f and element-wise multiplying it with the features of the network. Here, the mask W_f is one of the outputs of the meta-model.

The meta-model comprises a fully connected layer, which takes a feature f and a loss value $\mathcal{L}$ as input. Each input is projected in a fixed-size embedding space and then concatenated to form a larger common space. Two weight vectors W_f and s_k can be obtained through a fully connected layer followed by a sigmoid activation function.

4.1.1 Discussion

Noise detection is an essential strategy for long-tailed with noisy label learning. Most of them are theoretically guaranteed to separate clean labels from the noisy set under long-tail problems. The technique is well-motivated and works well in general. However, noise detection is relatively difficult to implement and computationally costly [36], which often requires carefully designed training frameworks and complicated algorithms to mitigate the negative effect of data imbalance. It can also suffer from the accumulation of incorrectly selected samples and heavily biased data distribution. To benefit most from even false-selected samples, it is usually combined with loss correction or semi-supervised learning techniques.

4.2 Module Improvement

Module improvement seeks to improve network modules from their architecture, which includes enhancing the feature extractor, the model classifier, or both. The resulting architectures usually yield better representation and higher prediction accuracy. We will introduce methods that utilize module improvement techniques in this section.

• RoLT [42] refine the detected noisy labels by generating a soft pseudo-label through both the Empirical Risk Minimization (ERM) classifier and the Nearest Class Mean (NCM) classifier. The ERM classifier is known to be biased toward head classes, but the NCM classifier has a more balanced classification boundary [17]. Both predictions $(\hat{y}^{erm}, \hat{y}^{ncm})$ and the original label y are used to generate soft pseudo-labels for class $k \in [K]$ as:

$$\tilde{y}_k = \begin{cases} \sum_{\hat{y} \in \mathcal{G}} \mathbb{P}(\hat{y} = y^*) \cdot \mathbb{I}(\hat{y} = k) & \text{if } k \in \mathcal{G} \\ \frac{1 - \sum_{\hat{y} \in \mathcal{G}} \mathbb{P}(\hat{y} = y^*)}{K - \mathcal{G}} & \text{otherwise.} \end{cases} \tag{4.10}$$

Where $\mathcal{G}$ is the guessing label set, $\mathbb{I}(\cdot)$ is an indicator function that returns 1 if the condition is true, and $\mathbb{P}(\hat{y} = y^*)$ denotes the probability that $\hat{y}$ matches the true label. Then, the loss function is calculated by combining the cross-entropy of clean samples with original labels and noisy samples with soft pseudo-labels.

• PCL [41] also leveraged the advantage of prototypes of classes that resist against class-imbalance but proposed using dynamic thresholds for noise detection. They introduced a Prototypical Classifier using the same equation to calculate prototypes as 4.1. But, for class $k \in 1 \dots K$,

the distance is obtained using cosine similarity, as:

$$\mathbb{P}_\theta(Y = k|x) = \frac{exp(f_{\boldsymbol{\theta}}(x)^T \mathbf{c}_k)}{\sum_{k'} exp(f_{\boldsymbol{\theta}}(x)^T \mathbf{c}_{k'})}. \tag{4.11}$$

The predicted labels are refined through a threshold τ_t, where t is the optimization iteration. Samples with confidence scores greater than the threshold are deemed clean samples. The dynamic threshold is set as an increasing function of t,

$$\tau_t = \gamma^t \tau_0. \tag{4.12}$$

• H2E [46] proposed a Hard-to-Easy framework that turns noise affected by the negative impact of content bias and class bias (hard noise) into identifiable noise (easy noise) through invariant multi-environment learning, thus removing identified 'easy' noise to obtain a cleaner representation. To better adapt to long-tailed distribution, the noise identifier combines LWS [17] and Logit adjustments [30] classifiers as:

$$g(\cdot) = f(\Phi(\cdot)) - w \cdot log\pi, \tag{4.13}$$

where $\Phi(\cdot)$ donates the extracted feature, $f(\cdot)$ projects feature vectors to the logit space, w is learnable parameters, π is the class distribution $p(y)$. To mitigate context bias, three environments with diverse classes and contexts are constructed, then an IRM optimization [1] is applied to regularize the identifier to be equally optimal across all environments. For noise removal, they use Mixup [49] strategy to fine-tune $\Phi(\cdot)$ and $f(\cdot)$ which generates mixed training pairs $(\tilde{x_{ij}}, \tilde{y_{ij}})$, combining

two images and labels with denoise weight δ proportion to the confidences $g(x_i)/g(x_j)$, i.e.,

$$\begin{aligned} \tilde{x_{ij}} &= \delta_{ij} x_i + (1 - \delta_{ij}) x_j, \\ \tilde{y_{ij}} &= \delta_{ij} y_i + (1 - \delta_{ij}) y_j. \end{aligned} \tag{4.14}$$

It is iteratively refined to learn better representations against noisy labels.

After the refine iteration of H2E, the model $f(Phi(\cdot))$ only needs to tackle the class bias in the clean dataset. Therefore, the model is trained with the balanced softmax loss [34] and optimized by re-weighting all samples from training data according to the weight parameter generated by the noise identifier.

4.2.1 Discussion

Module improvement as a fundamental problem for long-tailed with noisy label learning is worth further exploring. Representation learning and classifier design focused on producing better predictions, alleviating the negative effect of either long-tailed distribution or noisy labels. However, it usually focuses on only one side of the problem and requires other techniques to handle the rest. Despite this, the idea of classifier learning is conceptually simple and thus can be explored for new methods.

4.3 Loss Adjustment

Loss adjustment seeks to reduce the negative effect of biased data by adjusting loss values before updating the DNN. Loss functions designed for long-tailed learning tend to re-balance

sample weights, assigning higher loss to tail classes to allocate the model's attention to learning these classes. In noisy label learning, losses are usually specifically designed against the presence of noisy labels. All these losses are designed assuming that only one bias is presented. As a result, previous losses do not work effectively when the dataset is both imbalanced and noisy. However, the good news is different losses can be leveraged simultaneously to ensure robustness against both label noise and long-tailed distribution. Several novel losses are introduced for model fine-tuning.

• Two-stage training approach from [18] fine-tune their model with a combined loss of the logit adjustment loss [30] and the SuperLoss [6]. The model is first pre-trained with self-supervised learning approaches. The logit adjustment loss is formulated as Eq. 2.10. The obtained $\mathcal{L}_{LA}$ is then used in the SuperLoss, which is computed as follows:

$$\mathcal{L}_{LA+SL}(\mathcal{L}_{LA}, \sigma^*) = (\mathcal{L}_{LA} - \tau)\sigma^* + \lambda(log\sigma^*)^2, \tag{4.15}$$

where λ is a regularization trade-off, τ is the expected loss for the 'average' sample, and σ^* is the optimal confidence which can be computed in the formula as follows:

$$\sigma^*_\lambda(\mathcal{L}_{LA}) = exp(-W(\frac{1}{2}\max(\frac{\mathcal{L}_{LA} - \tau}{\lambda}, \frac{2}{e}))). \tag{4.16}$$

• PCL [41] combined CE with prototypical contrastive loss and MixUp unsupervised contrastive loss. The prototypical contrastive loss is used to improve the performance of noise detection, aiming to learn a reweighting of the input. The weighted loss is computed as follows:

$$\mathcal{L}_{pc} = \frac{-1}{\sum_{i=1}^{N} w_i} \sum_{i=1}^{N} w_i log \frac{exp(f_{\boldsymbol{\theta}}(x) \cdot \mathbf{c}_{y_i}/\tau)}{\sum_{k=1}^{K} exp(f_{\boldsymbol{\theta}}(x) \cdot \mathbf{c}_k/\tau)}, \tag{4.17}$$

where w_i is the weights that reflect the likelihood of samples being correctly-labeled, $\mathbf{c}_k$ is a weighted version of 4.1, and τ is a temperature parameter.

To prevent the negative effect of noise labels on representation learning, the model is trained to optimize the unsupervised contrastive loss using unbiased training labels. The idea is to pull together two embeddings of the same samples while pushing apart from others. Let $f'_{\boldsymbol{\theta}}(x_i)$ denotes the embedding of the augmented version of $f_{\boldsymbol{\theta}}(x_i)$, the MixUp unsupervised contrastive loss is computed as:

$$\mathcal{L}^i_{cc} = -log \frac{exp(f_{\boldsymbol{\theta}}(x) \cdot f'_{\boldsymbol{\theta}}(x_i)/\tau)}{\sum_{b=0}^{B} exp(f_{\boldsymbol{\theta}}(x) \cdot f'_{\boldsymbol{\theta}}(x_b)/\tau)}, \tag{4.18}$$

where B is the mini-batch size. The overall loss is written as: $\mathcal{L} = \mathcal{L}_{ce} + \lambda_1 \mathcal{L}_{pc} + \lambda_2 \mathcal{L}_{cc}$, where λ_1 and λ_2 are trade-off parameters.

• Sample selection with balanced loss [50] follows the semi-supervised learning methods MixMatch [3] but re-balance the clean label loss and re-exploit regularization term $\mathcal{L}_{reg}$ 4.6 from warm up stage. With the clean samples L and unlabeled sample U, MixMatch will transform them into L' and U' with soft labels $q \in [0,1]^N$. Then uses the cross-entropy loss and the mean square error to obtain losses on both samples as:

$$\mathcal{L}_L = \frac{1}{|L'|} \sum_{(x,q) \in L'} \sum_{i=1}^{N} q_i log \left[1 + \sum_{j \neq i} \frac{exp(f_j(x))}{exp(f_i(x))} \right], \tag{4.19}$$

$$\mathcal{L}_U = \frac{1}{|U'|} \sum_{(x,q) \in U'} \|q - p_{model}(x)\|_2^2. \tag{4.20}$$

Since the model bias may not directly relate to the class frequency and the true label frequency is unknown under noisy labels, they compute a matrix M to capture the model bias. The matrix M is calculated before each training epoch, where M_{ij} represents the probability

that the learning model predicts a sample from class i to class j. The matrix is further averaged with Exponentially Moving Average (EMA) [22] to obtain an averaged matrix $\tilde{M}$. With $R_{ij} = \tilde{M}_{ij}/\tilde{M}_{ji}, g(\cdot) = \gamma_{sup} \cdot \mathbb{I}(R_{ij} > 1) + \gamma_{rel} \cdot \mathbb{I}(R_{ij} \leq 1)$, where $g(\cdot)$ is a function of R_{ij}, γ_{sup} and γ_{rel} are hyper-parameters, they adjust Eq. 4.19 as:

$$\mathcal{L'}_L = \frac{1}{|L'|} \sum_{(x,q) \in L'} \sum_{i=1}^{N} q_i log \left[1 + \sum_{j \neq i} R_{ij}^{g(R_{ij})} \frac{exp(f_j(x))}{exp(f_i(x))} \right]. \tag{4.21}$$

The overall balanced loss is written as: $\mathcal{L} = \mathcal{L'}_L + \lambda_u \mathcal{L}_U + \lambda_{reg} \mathcal{L}_{reg}$, where λ_u and λ_{ref} are trade-off parameters.

• LNOR [10] proposed a leave-noise-out regularization loss for confident noisy samples combined with the penalty of online prior probabilities and the loss of strongly and weakly augmented samples. Given the confident noisy samples, they are used to regularize the network, preventing predictions of these noisy samples from being positive on their labels. The regularization term is computed as:

$$\mathcal{L}_n = - \sum_{(x_j, y_j) \in \mathbb{N}} log(1 - p_j[y_j]), \tag{4.22}$$

where $\mathbb{N}$ is the confident noisy sample space and $p_j = f_{\boldsymbol{\theta}}(\cdot)$.

The online prior probabilities reflect the fitting degree on classes, which is dynamically estimated as:

$$\boldsymbol{q} := (1 - \tau)\boldsymbol{q} + \tau \sum_{(x_i, y_i) \in \mathbb{C}} f_{\boldsymbol{\theta}}(x_i)/|\mathbb{C}|, \tag{4.23}$$

where $\mathbb{C}$ is the clean sample space, τ is a constant, and $\boldsymbol{q}$ is initialized by the ratios of samples in N classes. Then, the online prior penalty is calculated as follows:

$$v(x_i) = -(1 - \boldsymbol{q}^T) log(f_{\boldsymbol{\theta}}(x_i)). \tag{4.24}$$

Labels with larger prior probabilities are penalized harder, and the reduction of the target label is distributed to other labels in negative relation to their probabilities. After combing the clean set loss 4.9 with LNOR loss and penalty, the final loss is formed as: $\mathcal{L} = \mathcal{L}_c + \mathcal{L}_n + \lambda \sum_{(x_i,y_i)\in\mathbb{C}}[v(x_i) + v(\hat{x}_i)]$, where λ is the trade-off parameter, x_i and $\hat{x}_i$ is weakly and strongly augmented samples respectively.

• MFRW-MES [35] proposed a meta-learning equalization loss (MES) that formulates a new weight vector using the output of a meta-model to pass into the softmax cross-entropy from [37]. The original weighted softmax cross-entropy loss is calculated as:

$$\mathcal{L}_{SEQL} = -\sum_{j=1}^{N} y_j log(\frac{exp(z_j)}{\sum_{k=1}^{N} \tilde{w}_k exp(z_k)}), \qquad \tilde{w}_k = 1 - \beta T_\lambda(f_k)(1 - y_k) \tag{4.25}$$

where z is the network output score with the softmax function, $\tilde{w}_k$ is the weight term, $T_\lambda(f_k)$ is a threshold function $\in (0, 1)$ based on class frequency f_k, and β is a Bernoulli random variable. MES replaces the threshold function $T_\lambda(f_k)$ with the output weight vector s_k obtained from the meta-model training, described in 4.1, which is differentiated between classes and between samples.

4.3.1 Discussion

Loss adjustment is an efficient technique for reducing the negative impact of long-tailed distribution and noisy labels. This type of method seeks to assign larger weights to clean samples and tail samples, emphasizing these parts of samples to optimize the network better. It is difficult as clean samples tend to have smaller losses, and tail samples tend to have larger losses. Thus, loss

adjustment often requires pre-step like noise detection and redesigned classifier as the backbone.

Part III

Third Part

Chapter 5

Methodological Comparison

5.1 Codebase Implementation

The codebase has three main sections: noisy learning, long-tailed learning, and long-tailed learning with noisy labels. The implemented methods are listed as follows:

- Cross Entropy
- Long-tail learning
  - Focal Loss [24]
  - Class-balanced Loss [9]
  - Class-balanced Foacl Loss
  - LDAM Loss [5]
  - LADE Loss [15]
  - Logits Adjustments Loss [30]
  - IB Loss [32]
  - IB+Focal Loss [32]
  - BKD Loss [51]
  - VS Loss [20]
  - WVN+RS [19]
- Noisy Label
  - GCE [53]
  - Co-teaching [13]
  - Co-teaching+ [47]
  - Dual T [45]
  - DivideMix [23]
  - Early Learning Regularization [25]
  - Super Loss [6]
  - CORES [7]
- Long-tail with Noisy Label
  - CORES+Logits Adjustments
  - RoLT [42]
  - Self-supervised Two Staged Approach [18]
  - PCL [41]
  - CNLCU [44]

5.1.1 Datasets and Implementation Details

We introduce the codebase settings.

Datasets. We evaluate the implemented methodologies on two standard benchmarks, CIFAR-10 and CIFAR-100 [21], with simulated label noise and class imbalance. Both datasets contain 50000 samples for training and 10000 samples for evaluation with 10 and 100 classes, respectively. When the long-tailed distribution is applied, the number of training samples for each class decays on the long-tailed type and the imbalance ratio ρ. In our experiment, we use exponential long-tailed type and symmetric label noise with long-tailed rates $\rho = 10, 50$ and noise rates $\gamma = 0.1, 0.2, 0.3$.

Implementation Details. We implement all experiments in PyTorch. We have PreAct ResNet-18 and ResNet-34 [14] as the network backbones. All networks are trained for 200 epochs with SGD optimizer with a momentum of 0.9, batch size of 64, weight decay factor of 0.0001, and base learning rate of 0.1. The learning rate schedule is set as MultiStepLR, which decays the learning rate at 60 and 120 epochs. We use the same basic data augmentation ($i.e.$, Random resize and crop to 32, random horizontal flip, and normalization). For other method-related hyper-parameters, we set the values following the default settings in the original papers.

The results will be saved under the /results/dataset/loss_type folder, which is named based on noise type, noise rate, long-tail type, long-tailed rate, and specific training method. The results include a .txt file that records train accuracy, best test accuracy, and last test accuracy displayed in table form, a .pth file that saves the trained model parameters, a .best.path file that has the model

Table 5.1: Test accuracy results for each method on CIFAR-10. * indicates the results obtained by our implementation.

Type	Imbalance Ratio	10			50		
	Noise Level	0.1	0.2	0.3	0.1	0.2	0.3
Baseline	Cross-Entropy	83.65	78.49	77.85	69.9	65.94	55.73
NL	DivideMix	61.75	78.12	81.26	73.6	74.37	80.07
LT	Class-balanced	80.89	75.51	74.78	65.66	63.29	57.15
	Logits adjustment	82.28	75.24	71.28	71.51	62.46	65.79
LT_NL	RoLT*	84.46	81.25	80.50	72.61	69.96	60.08
	RoLT-DRW*	84.21	80.35	77.35	87.64	84.11	80.42
	PCL*	85.35	85.11	**85.92**	77.13	75.15	74.79
	CNLCU_soft*	87.20	85.30	84.04	**89.74**	88.15	**86.89**
	CNLCU_hard*	**87.28**	**85.69**	84.17	89.66	**88.24**	86.60

parameter when test accuracy reaches its maximum, and a .npy that saves the loss values. You can always resume the training from last time with the .pth file.

5.2 Results on all classes

Table 5.1 shows the empirical results on CIFAR-10. Table 5.2 shows the empirical results on CIFAR-100. We each select four methods from noisy label learning and long-tailed learning as baselines, including Co-teaching [13], Co-teaching+ [47], DivideMix [23], ELR [25], Class-balanced Loss [9], Focal loss [24], logits adjustment [30], and LDAM[5]. We select five methods for long-tailed distribution with noisy labels, including (RoLT, RoLT-DRW) [42], PCL [41], (CNLCU_soft, and CNLCU_hard) [44]. Other methods can be tested following procedures in the codebase.

The results show that most long-tailed with noisy label methods perform better than

Table 5.2: Test accuracy results for each method on CIFAR-100. * indicates the results obtained by our implementation.

Type	Imbalance Ratio	10			50		
	Noise Level	0.1	0.2	0.3	0.1	0.2	0.3
Baseline	Cross-Entropy	42.08	37.76	33.98	33.46	28.13	24.99
NL	DivideMix	62.26	61.3	58.98	49.58	48.46	44.55
LT	Class-balanced	42.24	34.35	32.44	26.17	22.32	19.05
	Logits adjustment	46.13	38.46	31.07	28.92	28.17	19.53
LT_NL	RoLT*	53.33	48.75	47.41	38.34	33.78	31.65
	RoLT-DRW*	53.85	48.21	44.15	58.54	55.32	49.94
	PCL*	**64.68**	**61.74**	**59.47**	**69.0**	**67.83**	59.13
	CNLCU_soft*	58.34	55.15	52.36	64.7	62.23	59.96
	CNLCU_hard*	58.17	55.66	52.44	64.57	62.97	**60.07**

previous methods in terms of accuracy. RoLT [42] performed varied under different imbalance ratios. We suspect that the accuracy of the detected noisy samples may be the cause of this. CNLCU [44] performed relatively stable despite the type of truncation; sometimes, even better accuracy under higher imbalance ratios. PCL [41] performed unexpectedly well under more classes. Overall, the average accuracy indicates that there is still room for improvement.

Part IV

Fourth Part

Chapter 6

Future Research Directions

In this section, we identify several drawbacks of current methods and discuss possible future research directions for long-tailed with noisy label learning.

Meta-Learning. Instead of solving part of the problem once at a time, meta-learning handle both problems simultaneously by learning a weighting function and feeding back the needed weight vectors to the main network. However, meta-learning requires unbiased data as meta data which is difficult to obtain under long-tailed distribution with noisy labels. The higher quality of the meta data, the more improvement the model can achieve [16]. Thus, it remains an open question about how to obtain unbiased meta data for the meta-learning process efficiently.

Noise Detection. One key challenge in long-tailed distribution with noisy label learning is the corruption of data samples. Separating noise labels from clean labels is a feasible and straightforward solution. Multiple methods have been proposed to solve this challenge. However, most existing methods require a high computational cost. Considering noise detection is funda-

mental for other advanced techniques, it is valuable to design simpler approaches with lower costs.

Chapter 7

Conclusion

In this project, we have reviewed deep long-tailed learning with noisy labels methods proposed before early-2023, according to the category of sample selection, module improvement, and loss adjustment. We have built a codebase of long-tailed with noisy label methods to evaluate to what extent they address the issue of long-tailed distribution with noisy labels and also as baselines for future research. We expect this project can provide a better understanding of long-tailed distribution with noisy label learning for the community.

Bibliography

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[2] Yoshua Bengio, Yann Lecun, and Geoffrey Hinton. Deep learning for ai. *Communications of the ACM*, 64(7):58–65, 2021.

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

[4] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

[5] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In H. Wallach, H. Larochelle, A. Beygelz-

imer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[6] Thibault Castells, Philippe Weinzaepfel, and Jerome Revaud. Superloss: A generic loss for robust curriculum learning. *Advances in Neural Information Processing Systems*, 33:4308–4319, 2020.

[7] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. In *International Conference on Learning Representations*, 2021.

[8] Guillem Collell, Drazen Prelec, and Kaustubh Patil. Reviving threshold-moving: a simple plug-in bagging ensemble for binary and multiclass imbalanced data. *arXiv preprint arXiv:1606.08698*, 2016.

[9] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019.

[10] Chaowei Fang, Lechao Cheng, Huiyan Qi, and Dingwen Zhang. Combating noisy labels in long-tailed image classification. *arXiv preprint arXiv:2209.00273*, 2022.

[11] Aritra Ghosh, Himanshu Kumar, and P Shanti Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[12] Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406*, 2020.

[13] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6626–6636, 2021.

[16] Shenwang Jiang, Jianan Li, Ying Wang, Bo Huang, Zhang Zhang, and Tingfa Xu. Delving into sample loss curve to embrace noisy and imbalanced data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7024–7032, 2022.

[17] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

[18] Shyamgopal Karthik, Jérome Revaud, and Boris Chidlovskii. Learning from long-tailed data with noisy labels. *arXiv preprint arXiv:2108.11096*, 2021.

[19] Byungju Kim and Junmo Kim. Adjusting decision boundary for class imbalanced learning. *IEEE Access*, 8:81674–81685, 2020.

[20] Ganesh Ramachandra Kini, Orestis Paraskevas, Samet Oymak, and Christos Thrampoulidis. Label-imbalanced and group-sensitive classification under overparameterization. *Advances in Neural Information Processing Systems*, 34:18970–18983, 2021.

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[22] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[23] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

[24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[25] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342, 2020.

[26] Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International conference on machine learning*, pages 6226–6236. PMLR, 2020.

[27] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2537–2546, 2019.

[28] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International conference on machine learning*, pages 6543–6553. PMLR, 2020.

[29] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pages 3355–3364. PMLR, 2018.

[30] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020.

[31] Malek Mouhoub, Samira Sadaoui, Otmane Ait Mohamed, and Moonis Ali. *Recent Trends and Future Technology in Applied Intelligence: 31st International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2018, Montreal, QC, Canada, June 25-28, 2018, Proceedings*, volume 10868. Springer, 2018.

[32] Seulki Park, Jongin Lim, Younghan Jeon, and Jin Young Choi. Influence-balanced loss for imbalanced visual classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 735–744, 2021.

[33] Haim Permuter, Joseph Francos, and Ian Jermyn. A study of gaussian mixture models of color and texture features for image classification and segmentation. *Pattern recognition*, 39(4):695–706, 2006.

[34] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020.

[35] Simone Ricci, Tiberio Uricchio, and Alberto Del Bimbo. Meta-learning advisor networks for long-tail and noisy labels in social image classification. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2023.

[36] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[37] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11662–11671, 2020.

[38] Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, 2017.

[39] Qizhou Wang, Bo Han, Tongliang Liu, Gang Niu, Jian Yang, and Chen Gong. Tackling instance-dependent label noise via a universal probabilistic model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10183–10191, 2021.

[40] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019.

[41] Tong Wei, Jiang-Xin Shi, Yu-Feng Li, and Min-Ling Zhang. Prototypical classifier for robust class-imbalanced learning. In *Advances in Knowledge Discovery and Data Mining: 26th Pacific-Asia Conference, PAKDD 2022, Chengdu, China, May 16–19, 2022, Proceedings, Part II*, pages 44–57. Springer, 2022.

[42] Tong Wei, Jiang-Xin Shi, Wei-Wei Tu, and Yu-Feng Li. Robust long-tailed learning under label noise. *arXiv preprint arXiv:2108.11569*, 2021.

[43] Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *International conference on learning representations*, 2021.

[44] Xiaobo Xia, Tongliang Liu, Bo Han, Mingming Gong, Jun Yu, Gang Niu, and Masashi

Sugiyama. Sample selection with uncertainty of losses for learning with noisy labels. *arXiv preprint arXiv:2106.00445*, 2021.

[45] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. *Advances in neural information processing systems*, 33:7260–7271, 2020.

[46] Xuanyu Yi, Kaihua Tang, Xian-Sheng Hua, Joo-Hwee Lim, and Hanwang Zhang. Identifying hard noise in long-tailed sample distribution. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 739–756. Springer, 2022.

[47] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173. PMLR, 2019.

[48] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[49] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[50] Lefan Zhang, Zhang-Hao Tian, and Wei Wang. Learning from long-tailed noisy data with sample selection and balanced loss. *arXiv preprint arXiv:2211.10906*, 2022.

[51] Shaoyu Zhang, Chen Chen, Xiyuan Hu, and Silong Peng. Balanced knowledge distillation for long-tailed learning. *Neurocomputing*, 2023.

[52] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[53] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018.