Learning with Noisy Labels via Sparse Regularization

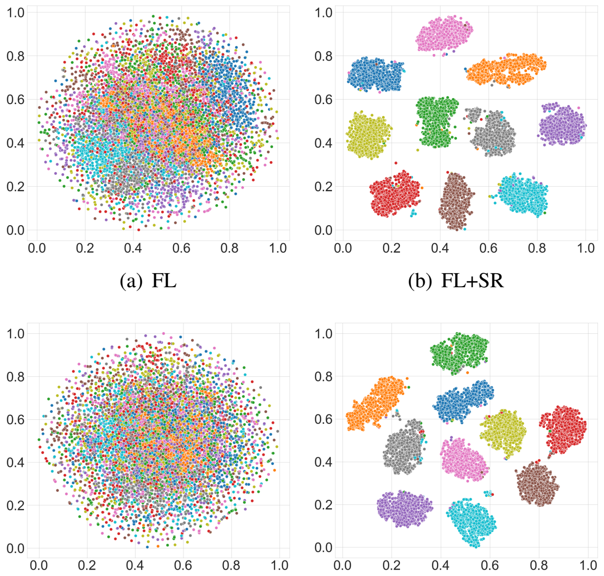
Xiong Zhou[1,2] Xianming Liu[1,2*] Chenyang Wang[1] Deming Zhai[1] Junjun Jiang[1,2] Xiangyang Ji[3]
[1]Harbin Institute of Technology [2]Peng Cheng Laboratory [3]Tsinghua University
{cszx,csxm,cswcy,zhaideming,junjunjiang}@hit.edu.cn xyji@tsinghua.edu.cn

Abstract

Learning with noisy labels is an important and challenging task for training accurate deep neural networks. Some commonly-used loss functions, such as Cross Entropy (CE), suffer from severe overfitting to noisy labels. Robust loss functions that satisfy the symmetric condition were tailored to remedy this problem, which however encounter the underfitting effect. In this paper, we theoretically prove that ***any loss can be made robust to noisy labels*** *by restricting the network output to the set of permutations over a fixed vector. When the fixed vector is one-hot, we only need to constrain the output to be one-hot, which however produces zero gradients almost everywhere and thus makes gradient-based optimization difficult. In this work, we introduce the sparse regularization strategy to approximate the one-hot constraint, which is composed of network output sharpening operation that enforces the output distribution of a network to be sharp and the ℓ_p-norm ($p \leq 1$) regularization that promotes the network output to be sparse. This simple approach guarantees the robustness of arbitrary loss functions while not hindering the fitting ability. Experimental results demonstrate that our method can significantly improve the performance of commonly-used loss functions in the presence of noisy labels and class imbalance, and outperform the state-of-the-art methods. The code is available at https://github.com/hitcszx/lnl_sr.*

1. Introduction

Deep neural networks (DNNs) have achieved remarkable success on various computer vision tasks, such as image classification, segmentation, and object detection [7]. The most widely used paradigm for DNN training is the end-to-end supervised manner, whose performance largely relies on massive high-quality annotated data. However, collecting large-scale datasets with fully precise annotations (or called clean labels) is usually expensive and time-consuming, and sometimes even impossible. Noisy labels, which are systematically corrupted from ground-truth labels, are ubiquitous in many real-world applications, such as online queries [19], crowdsourcing [1], adversarial attacks [30], and medical images analysis [12]. On the other hand, it is well-known that over-parameterized neural networks have enough capacity to memorize large-scale data with even completely random labels, leading to poor performance in generalization [28, 1, 11]. Therefore, robust learning with noisy labels has become an important and challenging task in computer vision [25, 8, 12, 31].



(a) FL (b) FL+SR

(c) GCE (d) GCE+SR

Figure 1. Visualization of learned representations on MNIST with 0.8 symmetric label noise. The representations learned by the proposed sparse regularization (SR)-enhanced methods are significantly discriminative than that learned by original losses, which are with more separated and clearly bound margin.

To prevent over-fitting to mislabeled data, many strategies have been presented in the literature, among which robust loss function design is one of the most popular approaches since it enjoys simplicity and universality. Ghosh *et al.* [6] theoretically proved that a loss function would be inherently tolerant to symmetric label noise as long as

[*] Correspondence to: Xianming Liu (csxm@hit.edu.cn)

it satisfies the symmetric condition. However, the derived loss functions according to this design principle, such as MAE [6] and Reverse Cross Entropy (RCE) [27], suffer from the underfitting effect on complicated datasets [3, 20]. As demonstrated in [29], the robustness of MAE can concurrently cause increased difficulty in training, leading to performance drops. On the other hand, the commonly-used CE and focal loss (FL) [18] enjoy the advantage of sufficient learning where the optimizer puts more emphasis on ambiguous samples, but they tend to overfit on noisy labels.

How to achieve robustness and learning sufficiency simultaneously? This question has motivated a large amount of work to design new loss functions that are robust to noisy labels as well as easy to fit clean ones. For instance, Zhang *et al.* proposed a generalization of cross entropy (GCE) [29], which behaves like a generalized mixture of MAE and CE. Wang *et al.* proposed the Symmetric Cross Entropy (SCE) loss [27], which combines RCE with CE. However, both GCE and SCE just perform the trade-off between symmetric loss and CE, which are only partially robust to noisy labels. Ma *et al.* [20] theoretically proved that by applying a simple normalization, any loss can be made robust to noisy labels. However, the normalization operation actually changes the form of loss functions, which no longer preserve the original fitting ability. The authors further proposed the Active Passive Loss (APL) to remedy this problem, which is the combination of two symmetric losses.

As reviewed above, all existing methods attempt to design new loss functions that satisfy the symmetric condition [6] to achieve robustness and achieve learning sufficiency meanwhile by combining multiple forms of loss function, such as symmetric loss and CE in [29, 27] and two symmetric losses in [20]. In this work, we propose a novel perspective to understand the symmetric condition, and prove in theory that any loss can be made robust through restricting the hypothesis class. Specifically, we demonstrate that label noise under risk minimization can be mitigated by restricting the network output to a permutation set of a fixed one-hot vector instead of modifying the loss function. This discrete process, however would result in many zero gradients, making the gradient-based optimization difficult. We then propose a sparse regularization strategy to approximate the one-hot constraint, which includes network output sharpening operation and ℓ_p-norm ($p \leq 1$) regularization in risk minimization. Experimental results on synthesis and real-world datasets demonstrate that our method can significantly improve the performance of commonly-used loss functions in the presence of noisy labels, as illustrated in Fig. 1, and outperform the state-of-the-art robust loss functions. Moreover, we evaluate the sparse regularization strategy on the long-tailed and step-imbalanced image classification tasks, which demonstrates that SR can also mitigate class imbalance well.

The main contributions of our work are highlighted as follows:

- To the best of our knowledge, we are the first work in the literature to meet the symmetric condition by restricting the hypothesis class. It offers a novel perspective of understanding the symmetric condition and an alternative approach for robust learning.
- We theoretically prove that any loss function can be made robust to noisy labels by restricting the network output to the set of permutations over a fixed vector.
- We propose a simple but effective approach for robust training through sparse regularization.
- We provide a principled approach to simultaneously achieve robust training and preserve the fitting ability of commonly-used losses such as CE.

2. The Proposed Method

In this section, we first introduce some preliminaries about robust learning. Subsequently, we present our finding in theory that any loss function can achieve noise tolerance via network output permutation. Furthermore, we offer a simple but effective approach for robust learning by introducing sparse regularization in network training. Finally, we provide analysis about the merit of our scheme—achieve better tradeoff between robustness and sufficient learning.

2.1. Preliminaries

Risk Minimization. Assume $\mathcal{X} \subset \mathbb{R}^d$ is the feature space from which the examples are drawn, and $\mathcal{Y} = [k] = \{1, ..., k\}$ is the class label space, *i.e.*, we consider a k-classification problem. In a typical classifier learning problem, we are given a training set, $\mathcal{S} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$, where $(\mathbf{x}_i, y_i)$ is drawn i.i.d. according to an unknown distribution, $\mathcal{D}$, over $\mathcal{X} \times \mathcal{Y}$. The classifier is a mapping function from feature space to label space $h(\mathbf{x}) = \arg\max_i f(\mathbf{x})_i$, where $f : \mathcal{X} \rightarrow \mathcal{C}$ denotes an approximation of $p(\cdot|\mathbf{x})$, and $\mathcal{C} \subset [0, 1]^k$, $\forall \mathbf{c} \in \mathcal{C}$, $\mathbf{1}^T\mathbf{c} = 1$. In deep learning, $f(\mathbf{x})$ is usually modeled by a neural network ending with a softmax layer.

The loss function is defined as a mapping $L : \mathcal{C} \times \mathcal{Y} \rightarrow \mathbb{R}^+$, where $\arg\min_{\mathbf{u}\in\mathcal{C}} L(\mathbf{u}, y) = \mathbf{e}_y$, and $\mathbf{e}_y$ denotes the one-hot vector. In this work, we consider the loss functional $\mathcal{L}$, where $\forall L \in \mathcal{L}$, $L(\mathbf{u}, i) = \ell_1(u_i) + \sum_{j\neq i} \ell_2(u_j)$, and ℓ_1 as well as ℓ_2 are two basic functions. For example, CE can be expressed by $\ell_1(u_i) = -\log u_i$ and $\ell_2(u_j) = 0$. Given any loss function L, and a classifier f, the L-risk of f is

$$R_L(f) = \mathbb{E}_{\mathcal{D}}[L(f(\mathbf{x}), y)] = \mathbb{E}_{\mathbf{x},y}[L(f(\mathbf{x}), y)], \quad (1)$$

where $\mathbb{E}$ denotes expectation. Under the risk minimization framework, the objective is to learn an optimal classifier, f^*, which is a global minimum of $R_L(f)$.

Noise Tolerance. We define the noise corruption process as that a clean label y is flipped into a noisy version $\tilde{y}$ with probability $\eta_{\mathbf{x},\tilde{y}} = p(\tilde{y}|y,\mathbf{x})$ (more label noise settings can be found in [6]). The corresponding noisy L-risk is

$$R_L^\eta(f) = \mathbb{E}_\mathcal{D}[(1-\eta_\mathbf{x})L(f(\mathbf{x}),y) + \sum_{i\neq y}\eta_{\mathbf{x},i}L(f(\mathbf{x}),i)],$$

where $\eta_\mathbf{x} = \sum_{i\neq y}\eta_{\mathbf{x},i}$ denotes the noise rate. Risk minimization under a given loss function is noise-tolerant if $R_L^\eta(f)$ shares the same global minimum as $R_L(f)$.

Symmetric Loss Functions. A symmetric loss function [22, 6] is proved to be noise-tolerant for a k-class classification under symmetric noise if the noise rate $\eta < \frac{k-1}{k}$ and the loss function satisfies

$$\sum_{i=1}^{k} L(f(\mathbf{x}),i) = C,\ \forall\mathbf{x}\in\mathcal{X},\forall f\in\mathcal{H}, \tag{2}$$

where C is a constant, and $\mathcal{H}$ is the hypothesis class.

The symmetric condition (2) stated above guarantees the noise tolerance by risk minimization on a symmetric loss function, *i.e.*, the classifier trained in noisy case has the same misclassification probability as that trained in noise-free case under the specified assumption. Moreover, if $R_L(f^*) = 0$, L is also noise-tolerant under an asymmetric noise, where f^* is a global risk minimum of R_L.

2.2. Noise Tolerance via Output Permutation

The symmetric condition (2) theoretically guarantees that a symmetric loss can lead to robust training. However, the derived loss functions according to this design principle usually suffer from underfitting [27, 3, 20]. On the other hand, the existing methods all pay attention to designing new loss functions, but never attempt to restrict the hypothesis class $\mathcal{H}$ to satisfy the symmetric condition. In this work, we propose to restrict the hypothesis class $\mathcal{H}$ such that any losses satisfy the symmetric condition (2) and thus become robust to label noise. Furthermore, we provide a theoretical analysis to demonstrate the noise tolerance of our scheme. The proofs can be seen in the supplementary materials.

First of all, we provide the definition of permutation operation, which plays an important role in our derivation.

Definition 1. *For a vector $\mathbf{v}\in\mathbb{R}^k$, the permutation operation on it is defined as [24]:*

$$\mathbf{v}_\pi = P_\pi\mathbf{v}, \tag{3}$$

where $P_\pi = [\mathbf{e}_{\pi_1}, \mathbf{e}_{\pi_2}, ..., \mathbf{e}_{\pi_k}]^T$ is the permutation matrix, and $\{\pi_1, \pi_2, ..., \pi_k\} = [k]$.

According to this definition, it is easy to find that $\mathbf{v}$ and $\mathbf{v}_\pi$ share the same space, *i.e.*, $\{v_{\pi_1}, v_{\pi_2}, ..., v_{\pi_k}\} = \{v_1, v_2, ..., v_k\}$. For instance, when $k=3$, $\mathbf{v} = \{v_1, v_2, v_3\}$ and $\pi = [3,1,2]$, then $P_\pi = [\mathbf{e}_3, \mathbf{e}_1, \mathbf{e}_2]^T$, and the vector after permutation operation is $\mathbf{v}_\pi = (v_3, v_1, v_2)^T$.

More generally, let $\mathcal{P}_\mathbf{v}$ denote the permutation set over $\mathbf{v}$, we have

$$\sum_{i=1}^{k}\ell(u_i) = \sum_{i=1}^{k}\ell(v_i), \forall\,\mathbf{u}\in\mathcal{P}_\mathbf{v}. \tag{4}$$

According to the above definition and derivation, we arrive at Lemma 1:

Lemma 1. *Given a vector $\mathbf{v}$, $\forall L\in\mathcal{L}$, we have*

$$\sum_{i=1}^{k} L(\mathbf{u},i) = C, \quad \forall\,\mathbf{u}\in\mathcal{P}_\mathbf{v}, \tag{5}$$

where $C = \sum_{i=1}^{k} L(\mathbf{v},i)$ is a constant when $\mathbf{v}$ is fixed.

Lemma 1 indicates that, when the network output $\mathbf{u}$ is restricted to belong to a permutation set $\mathcal{P}_\mathbf{v}$ of a fixed vector $\mathbf{v}$, any loss functions in $\mathcal{L}$ satisfy the symmetric condition. We further have the following theorems for symmetric and asymmetric noise that can be proved similarly as [6]:

Theorem 1 (Noise tolerance under symmetric noise)**.** *In a multi-class classification problem, $\forall L\in\mathcal{L}$, L is noise-tolerant under symmetric label noise if $\eta < 1-\frac{1}{k}$ and $f: \mathcal{X}\to\mathcal{P}_\mathbf{v}$, i.e.,*

$$\arg\min_{f:\mathcal{X}\to\mathcal{P}_\mathbf{v}} R_L(f) = \arg\min_{f:\mathcal{X}\to\mathcal{P}_\mathbf{v}} R_L^\eta(f), \tag{6}$$

where $\mathbf{v}$ is a fixed vector.

Theorem 2 (Noise tolerance under asymmetric noise)**.** *In a multi-class classification problem, let $f: \mathcal{X}\to\mathcal{P}_\mathbf{v}$, where $\mathbf{v}$ is a fixed vector, and suppose $L\in\mathcal{L}$ satisfy $0\le L(f(\mathbf{x}),i)\le\frac{C}{k-1}$, $\forall i\in[k]$. If $R_L(f^*) = 0$, then L is noise-tolerant under asymmetric or class-conditional noise when $\eta_{y,i} < 1-\eta_y$ with $\sum_{k\neq y}\eta_{y,i} = \eta_y$, $\forall\mathbf{x}$.*

Theorem 1 and 2 inspire us that label noise under risk minimization can be mitigated by restricting the network output to a permutation set instead of changing the loss function. This offers an alternative principle approach to achieve robust learning. However, the optimization is nontrivial when using gradient-based strategy because the constraint that $\mathcal{P}_\mathbf{v}$ is a discrete mapping produces many zero gradients. Instead, we turn to approximate the constraint by relaxing the output restriction of the hypothesis class with an error bound ε, *i.e.*, $\mathcal{H}_{\mathbf{v},\varepsilon} = \{f: \min_{\mathbf{u}\in\mathcal{P}_\mathbf{v}}\|f(\mathbf{x})-\mathbf{u}\|_2\le\varepsilon, \forall\mathbf{x}\}$. We can derive the risk bound as follows:

Theorem 3. *In a multi-class classification problem, if the loss function $L\in\mathcal{L}$ satisfies $|\sum_{i=1}^{k}(L(\mathbf{u}_1,i) - L(\mathbf{u}_2,i))|\le\delta$ when $\|\mathbf{u}_1-\mathbf{u}_2\|_2\le\varepsilon$, and $\delta\to 0$ as $\varepsilon\to 0$,* *then for symmetric label noise satisfying $\eta < 1 - \frac{1}{k}$, the risk bound for $f \in \mathcal{H}_{\mathbf{v},\varepsilon}$ can be expressed as*

$$R_L(f_\eta^*) - R_L(f^*) \leq 2c\delta,$$

where $c = \frac{\eta}{(1-\eta)k-1}$, f_η^* and f^* denote the global minimum of $R_L^\eta(f)$ and $R_L(f)$, respectively.

Theorem 3 indicates that when restricting the output of the network to belong to $\mathcal{H}_{\mathbf{v},\varepsilon}$, the noisy minimum f_η^*, compared to the clean minimum f^*, has a risk error bound $2c\delta$. And when $\varepsilon \to 0$, the bound also tends to 0. This implies that, by shrinking $\mathcal{H}_{\mathbf{v},\varepsilon}$, $f_\eta^* = \arg\min_{f \in \mathcal{H}_{\mathbf{v},\varepsilon}} R_L^\eta(f)$ converges to f^*.

2.3. Robust Learning via Sparse Regularization

Upon the above theoretical analysis, we propose a simple but effective approach for robust learning, which can make any losses robust to noisy labels by introducing sparse regularization on network output. Specifically, we consider the fixed vector $\mathbf{v}$ as a one-hot vector, *i.e.*, we restrict the network output to one-hot vectors $\mathcal{P}_{0-1}$. This discrete process would result in many zero gradients, making the optimization difficult. To approximate the one-hot constraint, we propose the sparse regularization strategy, which is composed of two modules: network output sharpening and ℓ_p-norm ($p \leq 1$) regularization in risk minimization.

Network Output Sharpening. The output sharpening module is to make the network output closer to a one-hot vector. One popular way to approximate a one-hot vector by the continuous mapping is to use a temperature-dependent softmax function, *i.e.*,

$$\sigma_\tau(\mathbf{z})_i = \frac{\exp(z_i/\tau)}{\sum_{j=1}^k \exp(z_j/\tau)}, \tag{7}$$

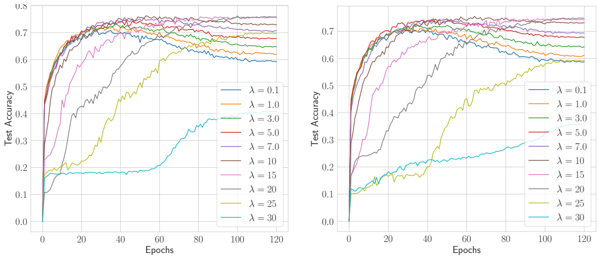
where $0 \leq \tau \leq 1$, and $\sigma_\tau(\mathbf{z})_i$ is a point in the probability simplex. Note that, in the limit situation where $\tau \to 0$, $\sigma_\tau(\mathbf{z})$ converges to a one-hot vector. In other words, with low temperatures, the distribution spends essentially all of its probability mass in the most probable state. Meanwhile, we may limit the value of z_i to the range $[-1, 1]$ by performing ℓ_2 normalization before the output sharpening to prevent it from trivial scaling solution.

ℓ_p-norm Regularization. We further introduce ℓ_p-norm regularization into risk minimization to promote the sparsity of network output. Specifically, we perform the following constrained risk minimization in network training:

$$\min_{f \in \mathcal{H}} R_L(f) \quad \text{s.t.} \quad \|f\|_p \leq \gamma, \tag{8}$$

where $0 \leq p \leq 1$, and γ is an appropriately selected parameter. In practice, we can convert to train a neural network by minimizing the following form:

$$\sum_{i=1}^{N} L(f(\mathbf{x}_i), \tilde{y}_i) + \lambda \|f(\mathbf{x}_i)\|_p^p. \tag{9}$$



(a) CE+SR (b) FL+SR

Figure 2. Test accuracy curve of different λ on CIFAR-10 with 0.6 symmetric label noise. Robustness is improved when increasing the weight of ℓ_p-norm for both CE and FL.

It is worth noting that, if we define $L(\mathbf{u}, i) = -\alpha \log u_i - \lambda u_i$ and $p = 1$, then Eq. (9) is equivalent to the Symmetric Cross Entropy (SCE) loss [27]. Thus, SCE can be regarded as a special case of a loss function with the ℓ_p-norm regularization.

For effective learning, the regularization parameter λ in Eq. (9) cannot be set too large, since the network would tend to minimize $\lambda \|f(\mathbf{x}_i)\|_p^p$ rather than $L(f(\mathbf{x}_i), \tilde{y}_i)$. As shown in Fig. 2(a), when $\lambda = 25$ and 30, although the curves look robust, they suffer from the underfitting effect. On the other hand, λ should not be set too small, otherwise robustness cannot be guaranteed (see $\lambda \leq 7$ in Fig. 2(a)). We need a large enough λ to maintain robustness. An effective strategy in practical implementation is to gradually increase the value of λ during training, *i.e.*, $\lambda_t = \lambda_0 \cdot \rho^{\lfloor t/r \rfloor}$ ($\rho \geq 1$), where t denotes the training epoch, and r denotes the updating rate of λ.

2.4. On the Robustness and Learning Sufficiency

In the following, we provide analysis about the robustness and learning sufficiency of the proposed scheme.

To obtain enough robustness, we restrict the output of the network to be one-hot, which naturally satisfies the symmetric condition. As for the output sharpening process, the derivative of $\sigma_\tau(\mathbf{z})_j$ with respect to z_i can be derived as

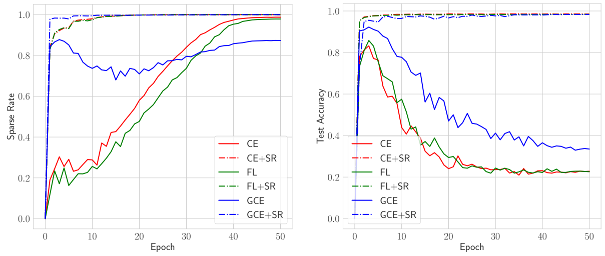
$$\frac{\partial \sigma_\tau(\mathbf{z})_j}{\partial z_i} = \frac{1}{\tau} \sigma_\tau(\mathbf{z})_i (\delta_{ij} - \sigma_\tau(\mathbf{z})_j), \tag{10}$$

where $\delta_{ij} = \mathbb{I}(i = j)$, and $\mathbb{I}(\cdot)$ is the identity function. We can see that the derivative is a scaled derivative of the original softmax function, so it would not change the optimization direction but change the step size. A larger step size in Eq. (10) would speed up the convergence to one-hot vectors. We can achieve this purpose by choosing an appropriate value for τ. On the other hand, we have $\lim_{\tau \to 0^+} \frac{\partial \sigma_\tau(\mathbf{z})_j}{\partial z_i} = 0$, which shows that the gradient would disappear if τ is small, so τ cannot be overly small to prevent underfitting. We fix τ in our implementation for simplicity, but we suggest to gradually decay τ in training, which can be regarded as an early-stopping strategy [16].

Moreover, consider the loss $L(\sigma_\tau(\mathbf{z}), y) = -\log \sigma_\tau(\mathbf{z})_y$, we have the derivative of $-\log \sigma_\tau(\mathbf{z})_y + \lambda \|\sigma_\tau(\mathbf{z})\|_p^p$ with respect to $\mathbf{z}$ as follows

$$
\begin{aligned}
&\frac{\partial - \log \sigma_\tau(\mathbf{z})_y}{\partial \mathbf{z}} + \lambda \frac{\partial \|\sigma_\tau(\mathbf{z})\|_p^p}{\partial \mathbf{z}} \\
&= - \frac{1}{\sigma_\tau(\mathbf{z})_y} \cdot \frac{\partial \sigma_\tau(\mathbf{z})_y}{\partial \mathbf{z}} + \lambda p \sum_{i=1}^{k} \frac{1}{[\sigma_\tau(\mathbf{z})_i]^{1-p}} \cdot \frac{\partial \sigma_\tau(\mathbf{z})_i}{\partial \mathbf{z}} \\
&= \underbrace{-\left(\frac{1}{\sigma_\tau(\mathbf{z})_y} - \frac{\lambda p}{[\sigma_\tau(\mathbf{z})_y]^{1-p}}\right) \cdot \frac{\partial \sigma_\tau(\mathbf{z})_y}{\partial \mathbf{z}}}_{\text{fitting term}} \\
&\quad \underbrace{+ \lambda p \sum_{i \neq y} \frac{1}{[\sigma_\tau(\mathbf{z})_i]^{1-p}} \cdot \frac{\partial \sigma_\tau(\mathbf{z})_i}{\partial \mathbf{z}}}_{\text{complementary term}}
\end{aligned}
$$

The fitting term denotes the gradient of learning towards the target y, while the complementary term limits the increase of $\sigma_\tau(\mathbf{z})_i$, $\forall i \neq y$. In the early phase of training, we guarantee enough fitting power by setting $\lambda p < 1$. As λ increases, the fitting term becomes weaker to mitigate label noise, but the complementary term still maintains a certain amount of fitting power through minimizing $\sigma_\tau(\mathbf{z})_i, \forall i \neq y$ to passively maximize $\sigma_\tau(\mathbf{z})_y$.

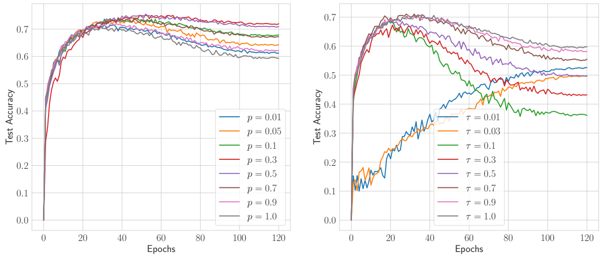


(a) Sparse Rate

(b) Test Accuracy

Figure 3. Sparse rate and test accuracy of different methods on MNIST with 0.8 symmetric label noise. One-hot constraint significantly improves the robustness while not hindering fitting ability.

On the other hand, we can regard the ℓ_p-norm regularized loss function $L(\mathbf{u}, i) + \lambda \|\mathbf{u}\|_p^p$ as a new loss function. If there exists $\alpha > 0$, such that $L(\mathbf{u}, i) + \alpha u_i^p$ is monotonically decreasing on u_i, then the new loss can be divided into the active loss $L(\mathbf{u}, i) + \alpha u_i^p$ and the passive loss $(\lambda - \alpha) u_i^p + \lambda \sum_{j \neq i} u_j^p$. In fact, α always exists for commonly-used loss functions, for example, when $L(\mathbf{u}, i) = -\log u_i$, we have $\alpha < \frac{1}{p}$ such that $-\log u_i + \alpha u_i^p$ keeps monotonically decreasing on u_i. Therefore, our proposed ℓ_p-norm regularization coincides with the Active Passive Loss proposed in [20]. The analysis demonstrates that our scheme achieves the best of both worlds of robustness and sufficient learning.



(a) CE with different p

(b) CE with different τ

Figure 4. Test accuracy curve of different parameters on CIFAR-10 with 0.6 symmetric label noise.

3. Experiments

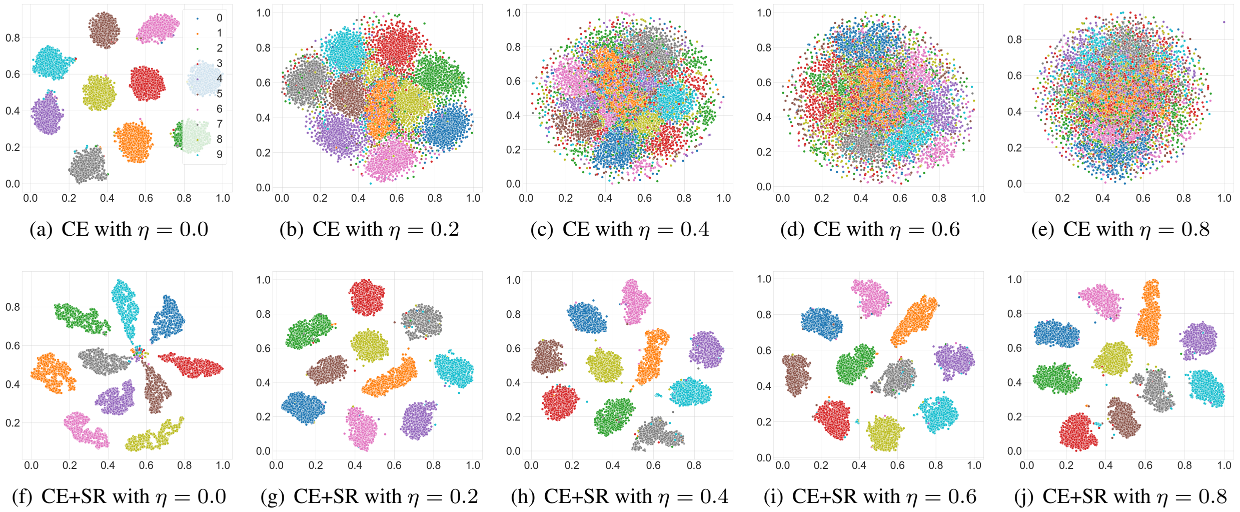
In this section, we empirically investigate the effectiveness of sparse regularization on synthetic datasets, including MNIST [15], CIFAR-10/-100 [14], and a real-world noisy dataset WebVision [17].

3.1. Empirical Analysis

One-hot Constraint Means Robustness. We first run a set of experiments on MNIST with 0.8 symmetric label noise to analyse the sparse rate and test accuracy during training, where the sparse rate is formulated as $\frac{\sum_{i=1}^{N} \mathbb{I}(\max_j f(\mathbf{x}_i)_j > 1 - 0.01)}{N}$, and $f(\mathbf{x}_i)$ is performed by the output sharpening with $\tau = 0.1$. If $\max_j f(\mathbf{x}_i)_j > 1 - 0.01$, then $\min_{\mathbf{u} \in \mathcal{P}_{0-1}} \|f(\mathbf{x}_i) - \mathbf{u}\|_2 \leq \sqrt{2} \cdot 0.01 = \varepsilon$. We add the sparse regularization strategy to CE, FL and GCE, the results are shown in Fig. 3. As we can see, with the role of SR, the sparse rate usually maintains a high value after several epochs, and the test accuracy curves of CE+SR, FL+SR and GCE+SR show enough robustness and learning efficiency for model to mitigate label noise, while the original losses have low sparse rate and poor accuracy. This validates that the noise tolerance can be obtained by restricting the output of a network to one-hot vectors.

Sparse Regularization can Mitigate Label Noise. As shown in Fig. 2, when we add SR to enhance the performance of CE and FL, the training process is increasingly robust as λ increases while not hindering the fitting ability ($\lambda \leq 20$). This demonstrates that learning with sparse regularization can be both robust and effective when mitigating label noise. The projected representations on MNIST are illustrated in Fig. 1. Under both settings, the representations learned by the SR-enhanced method are of significantly better quality than those learned by the original losses with more separated and clearly bounded clusters.

Parameter Analysis. We choose the different parameters τ, λ and p for sparse regularization to CE. The experiments are conducted on CIFAR-10 with 0.6 symmetric noise. We



(a) CE with $\eta = 0.0$ (b) CE with $\eta = 0.2$ (c) CE with $\eta = 0.4$ (d) CE with $\eta = 0.6$ (e) CE with $\eta = 0.8$

(f) CE+SR with $\eta = 0.0$ (g) CE+SR with $\eta = 0.2$ (h) CE+SR with $\eta = 0.4$ (i) CE+SR with $\eta = 0.6$ (j) CE+SR with $\eta = 0.8$

Figure 5. Visualization of learned representations on MNIST with different symmetric label noise ($\eta \in [0.0, 0.2, 0.4, 0.6, 0.8]$).

first tested $\tau \in \{0.01, 0.03, 0.1, 0.3, 0.5, 0.7, 0.9, 1\}$ while $\lambda = 0$, and the results can be show in Fig. 4(b). When τ is small (≤ 0.003), the curve is very robust, but it also suffer from significant underfitting problem. As τ increases, the curve falls into overfitting. Then we tested $\lambda \in \{0.1, 0.3, 1.0, 3.0, 5.0, 7.0, 10, 15, 20, 25, 30\}$ while $p = 0.1$ and $\tau = 1$. As shown in Fig. 2, curve gets more robustness as λ increases, but when $\lambda \geq 7$, it encounters severe underfitting since the optimization pays more attention on minimizing $\lambda\|f(\mathbf{x})\|_p^p$. Moreover, we tuned $p \in \{0.01, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1\}$ while $\tau = 1$ and $\lambda = 5$. The results in Fig. 4(a) show that both small p and large p tend to cause more overfitting to label noise, so we need choose an appropriate value.

Remark. As for parameter tunings, a simple principled approach for parameter setting is that parameters with strong regularization are selected for simple datasets, and parameters with weak regularization are selected otherwise. Specifically, this can be achieved by setting proper λ: the larger λ, the stronger regularization effect. However, as shown in Fig. 2, the large initial λ leads to underfitting since much attention is paid on minimizing $\lambda\|f(\mathbf{x})\|_p^p$, especially when $\lambda \cdot p > 1$. Instead, we turn to gradually increase λ, *i.e.*, $\lambda_t = \lambda_0 \cdot \rho^{\lfloor t/r \rfloor}$, where t is the iteration number and $r = 1$; $p = 0.1$ and 0.01, $\rho = 50^{1/120} \approx 1.03$ and $50^{1/200} \approx 1.02$ for CIFAR-10 and CIFAR-100 (where 120 and 200 represent the training epochs). The only parameter that requires careful tuning is λ_0, which is set as $\lambda_0 < \frac{1}{p}$. We use the similar parameter setting strategy for CIFAR-10, CIFAR-100, WebVision, and all achieve satisfactory results.

3.2. Evaluation on Benchmark Datasets

Experimental Details. The benchmark datasets, noise generation, networks, training details, parameter settings, more comparisons, and more experimental results can be found in the supplementary materials.

Baselines. We experiment with the state-of-the art methods GCE [29], SCE [27], NLNL [13], APL [20], and two effective loss functions CE and Focal Loss (FL) [18] for classification. Moreover, we add the proposed sparse regularization mechanism to CE, FL and GCE, i.e., CE+SR, FL+SR and GCE+SR. All the implementations and experiments are based on PyTorch.

Results. The test accuracies (mean ± std) under symmetric label noise are reported in Table 1. As we can see, our proposed SR mechanism significantly improves the robustness of CE, FL, and GCE, which achieve the top 3 best results in most test cases across all datasets. In the scenarios of serious noise, our CE+SR, FL+SR have a very obvious improvement over the original losses. For example, on MNIST with 0.8 symmetric noise, CE+SR outperforms CE by more than 52%. On CIFAR-10 with 0.8 symmetric noise, CE+SR outperforms CE by more than 31%. As for CIFAR-100, GCE and APL outperform our method by a small gap for $\eta = 0.6$ and 0.8, but they failed in the cases with small noise rates. The reason is that the fitting ability is not enough, which can be derived according to the experiments in the clean case. When APL and GCE meet a complicated dataset CIFAR-100 in a clean setting, their test accuracies are worse than the commonly used losses CE and FL, while the SR-enhanced methods outperform the original losses and achieve an improvement of ≥ 0.86. Therefore, our methods not only have good robustness, but also guarantee and even improve the fitting ability.

Results for asymmetric noise are reported in Table 2. Again, our methods significantly improve the robustness of the original version across all datasets and achieve top 3 best results in most cases. On MNIST, CE+SR and FL+SR outperform all the state-of-the-art methods over all asymmetric noise by a clear margin. More Surprisingly, the test

Table 1. Test accuracies (%) of different methods on benchmark datasets with clean or symmetric label noise ($\eta \in [0.2, 0.4, 0.6, 0.8]$). The results (mean±std) are reported over 3 random runs and the top 3 best results are **boldfaced**.

Datasets	Methods	Clean ($\eta = 0.0$)	Symmetric Noise Rate (η)			
			0.2	0.4	0.6	0.8
MNIST	CE	99.15 ± 0.05	91.62 ± 0.39	73.98 ± 0.27	49.36 ± 0.43	22.66 ± 0.61
	FL	99.13 ± 0.09	91.68 ± 0.14	74.54 ± 0.06	50.39 ± 0.28	22.65 ± 0.26
	GCE	99.27 ± 0.05	98.86 ± 0.07	97.16 ± 0.03	81.53 ± 0.58	33.95 ± 0.82
	SCE	99.23 ± 0.10	98.92 ± 0.12	97.38 ± 0.15	88.83 ± 0.55	48.75 ± 1.54
	NLNL	98.85 ± 0.05	98.33 ± 0.03	97.80 ± 0.07	96.18 ± 0.11	86.34 ± 1.43
	APL	99.34 ± 0.02	**99.14 ± 0.05**	98.42 ± 0.09	95.65 ± 0.13	72.97 ± 0.34
	CE+SR	99.33 ± 0.02	**99.22 ± 0.06**	**99.16 ± 0.04**	**98.85 ± 0.02**	**98.06 ± 0.86**
	FL+SR	99.35 ± 0.05	**99.25 ± 0.01**	**99.10 ± 0.10**	**98.81 ± 0.06**	**97.00 ± 1.28**
	GCE+SR	99.27 ± 0.06	99.13 ± 0.07	**99.06 ± 0.02**	**98.84 ± 0.09**	**98.37 ± 0.26**
CIFAR-10	CE	90.48 ± 0.11	74.68 ± 0.25	58.26 ± 0.21	38.70 ± 0.53	19.55 ± 0.49
	FL	89.82 ± 0.20	73.72 ± 0.08	57.90 ± 0.45	38.86 ± 0.07	19.13 ± 0.28
	GCE	89.59 ± 0.26	87.03 ± 0.35	82.66 ± 0.17	67.70 ± 0.45	26.67 ± 0.59
	SCE	91.61 ± 0.19	87.10 ± 0.25	79.67 ± 0.37	61.35 ± 0.56	28.66 ± 0.27
	NLNL	90.73 ± 0.20	73.70 ± 0.05	63.90 ± 0.44	50.68 ± 0.47	29.53 ± 1.55
	APL	89.17 ± 0.09	86.98 ± 0.07	83.74 ± 0.10	76.02 ± 0.16	46.69 ± 0.31
	CE+SR	90.06 ± 0.02	**87.93 ± 0.07**	**84.86 ± 0.18**	**78.18 ± 0.36**	**51.13 ± 0.51**
	FL+SR	89.86 ± 0.11	**87.94 ± 0.19**	**84.65 ± 0.05**	**77.85 ± 0.74**	**52.42 ± 0.76**
	GCE+SR	90.02 ± 0.40	**87.93 ± 0.27**	**84.82 ± 0.06**	**77.65 ± 0.05**	**51.97 ± 1.13**
CIFAR-100	CE	71.33 ± 0.43	56.51 ± 0.39	39.92 ± 0.10	21.39 ± 1.17	7.59 ± 0.20
	FL	70.06 ± 0.70	55.78 ± 1.55	39.83 ± 0.43	21.91 ± 0.89	7.51 ± 0.09
	GCE	63.09 ± 1.39	61.57 ± 1.06	56.11 ± 1.35	**45.28 ± 0.61**	**17.42 ± 0.06**
	SCE	70.64 ± 0.05	56.07 ± 0.26	39.88 ± 0.67	21.16 ± 0.65	7.63 ± 0.15
	NLNL	68.72 ± 0.60	46.99 ± 0.91	30.29 ± 1.64	16.60 ± 0.90	11.01 ± 2.48
	APL	67.95 ± 0.21	64.21 ± 0.24	57.70 ± 0.64	**45.20 ± 0.75**	**24.91 ± 0.42**
	CE+SR	72.19 ± 0.06	**67.51 ± 0.29**	**60.70 ± 0.25**	**44.95 ± 0.65**	**17.35 ± 0.13**
	FL+SR	72.08 ± 0.31	**67.64 ± 0.10**	**60.67 ± 0.48**	44.76 ± 0.08	17.16 ± 0.24
	GCE+SR	72.11 ± 0.26	**67.03 ± 0.46**	**60.68 ± 0.90**	44.66 ± 0.84	17.35 ± 0.42

accuracy (99.36 ± 0.05) of FL+SR under 0.4 asymmetric noise is higher than the clean case. On CIFAR-10 with 0.1 asymmetric noise, SCE has the best accuracy, but it loses the superiority in the other three cases where our method works much better than all other baselines with at least 1% increase. On CIFAR-100, the enhanced loss functions show particularly superior performance in all cases.

Representations. We further investigate the representations learned by CE+SR compared to those learned by CE. We extract the high-dimensional features at the second last full-connected layer, then project all test samples' features into 2D embeddings by t-SNE [26]. The projected representations on MNIST with different symmetric noise are illustrated in Fig. 5. As can be observed, CE encounters severe overfitting on label noise, and the embeddings look completely mixed when $\eta = 0.8$. On the contrary, CE+SR learns good representations with more separated and clearly bounded clusters in all noisy cases.

3.3. Evaluation on Real-world Noisy Dataset

Here, we evaluate our sparse regularization method on large-scale real-world noisy dataset WebVision 1.0 [17]. It contains 2.4 million images with real-world noisy labels, which crawled from the web using 1,000 concepts in ImageNet ILSVRC12 [5]. Since the dataset is very big, for quick experiments, we follow the training setting in [11] that only takes the first 50 classes of the Google resized image subset. We evaluate the trained network on the same 50 classes of WebVision 1.0 validation set, which can be considered as a clean validation set. We add sparse regularization to CE and GCE. The training details follow [20], where for each loss , we train a ResNet-50 [10] by using SGD for 250 epochs with initial learning rate 0.4, nesterov momentum 0.9, weight decay 3×10^{-5}, and batch size 512. The learning rate is multiplied by 0.97 after every epoch of training. All the images are resized to 224×224. Typical data agumentations including random width/height shift, color jittering, and random horizontal flip are applied. As shown in in Table 3, our proposed SR mechanism obviously enhances the performance of CE and FL, which outperform the existing loss functions SCE and APL with a clear margin ($\geq 2.2\%$). This verifies the effectiveness of SR against real-world label noise.

More Comparison. We also compare with Co-teaching[9], which is the representative work of sample selection, and PHuber-CE [23], which is a simple variant of gradient clip-

Table 2. Test accuracies (%) of different methods on benchmark datasets with clean or asymmetric label noise ($\eta \in [0.1, 0.2, 0.3, 0.4]$). The results (mean±std) are reported over 3 random runs and the top 3 best results are **boldfaced**.

Datasets	Methods	Asymmetric Noise Rate (η) 0.2	 0.3	 0.4
MNIST	CE	94.56 ± 0.22	88.81 ± 0.10	82.27 ± 0.40
	FL	94.25 ± 0.15	89.09 ± 0.25	82.13 ± 0.49
	GCE	96.69 ± 0.12	89.12 ± 0.24	81.51 ± 0.19
	SCE	98.03 ± 0.05	93.68 ± 0.43	85.36 ± 0.17
	NLNL	98.35 ± 0.01	97.51 ± 0.15	95.84 ± 0.26
	APL	98.89 ± 0.04	96.93 ± 0.17	91.45 ± 0.40
	CE+SR	**99.27 ± 0.06**	**99.24 ± 0.08**	**99.23 ± 0.07**
	FL+SR	**99.31 ± 0.02**	**99.23 ± 0.02**	**99.36 ± 0.05**
	GCE+SR	**99.22 ± 0.02**	**99.13 ± 0.05**	**99.09 ± 0.02**
CIFAR-10	CE	83.32 ± 0.12	79.32 ± 0.59	74.67 ± 0.38
	FL	83.37 ± 0.07	79.33 ± 0.08	74.28 ± 0.44
	GCE	85.93 ± 0.23	80.88 ± 0.38	74.29 ± 0.43
	SCE	86.20 ± 0.37	81.38 ± 0.35	75.16 ± 0.39
	NLNL	84.74 ± 0.08	81.26± 0.43	76.97 ± 0.52
	APL	86.50 ± 0.31	83.34 ± 0.39	77.14 ± 0.33
	CE+SR	**87.70 ± 0.19**	**85.63 ± 0.07**	**79.29 ± 0.20**
	FL+SR	**87.56 ± 0.29**	**85.10 ± 0.23**	**79.07 ± 0.50**
	GCE+SR	**87.55 ± 0.08**	**84.69 ± 0.46**	**79.01 ± 0.18**
CIFAR-100	CE	58.11 ± 0.32	50.68 ± 0.55	40.17 ± 1.31
	FL	58.05 ± 0.42	51.15 ± 0.84	41.18 ± 0.68
	GCE	59.35 ± 1.10	53.83 ± 0.64	40.91 ± 0.57
	NLNL	50.19 ± 0.56	42.81 ± 1.13	35.10 ± 0.20
	SCE	58.16 ± 0.73	50.98 ± 0.33	41.54 ± 0.52
	APL	62.80 ± 0.05	56.74 ± 0.53	42.61 ± 0.24
	CE+SR	**64.79 ± 0.01**	**59.09 ± 2.10**	**49.51 ± 0.59**
	FL+SR	**64.61 ± 0.67**	**58.94 ± 0.33**	**46.94 ± 1.68**
	GCE+SR	**64.35 ± 0.78**	**57.22 ± 0.80**	**49.51 ± 1.31**

Table 3. Top-1 validation accuracies (%) on WebVision validatoin set of ResNet-50 models trained on WebVision using different loss functions, under the Mini setting in [11].

Loss	CE	FL	SCE	APL	**CE+SR**	**FL+SR**
Acc	66.96	63.80	66.92	66.32	**69.12**	**70.28**

ping. As shown in Table 4, our method works better than Co-teaching and PHuber-CE.

Table 4. A comparison with other methods

Dataset	Method	Label Noise Type			
		sy 0.6	sy 0.8	asy 0.3	asy 0.4
CIFAR-10	CE	38.70	19.55	79.32	74.67
	Co-teaching	65.74	38.01	64.01	51.26
	PHuber-CE	75.44	41.18	76.06	55.78
	CE+SR	**78.18**	**51.13**	**85.63**	**79.29**
CIFAR-100	CE	21.39	7.59	50.68	40.17
	Co-teaching	34.28	7.94	42.82	33.67
	PHuber-CE	21.54	9.33	26.91	23.43
	CE+SR	**44.95**	**17.35**	**59.09**	**49.51**

3.4. Additional Experiments

Imbalanced Classification. As shown in Table 3, our method achieves the best result on WebVision that has a certain class imbalance. To better show the performance on class imbalance, we additionally test the ability of SR on the pure imbalanced classification task. We first follow the controllable data imbalance in [21] to create the imbalanced CIFAR-10/-100 by reducing the number of training examples and keeping the validation set unchanged. We also consider two imbalance types: long-tailed imbalance[4] and step imbalance [2]. The results shown in Table 5 are encouraging, where CE with sparse regularization achieves a very significant improvements compared with CE in all cases. Albeit simple, these additional experiments demonstrate that SR can also mitigate class imbalance well.

Table 5. Validation accuracy on imbalanced CIFAR-10/-100.

Dataset	Method	Imbalanced Type			
		lt-0.01	lt-0.1	step-0.01	step-0.1
CIFAR-10	CE	64.16	81.81	57.44	79.35
	CE+SR	**69.78**	**84.49**	**61.03**	**82.11**
CIFAR-100	CE	35.17	51.43	37.92	53.43
	CE+SR	**41.24**	**59.51**	**40.21**	**58.42**

4. Conclusion and Future Work

In this paper, we presented a novel method for learning with noisy labels. We first provided a theoretical conclusion that any loss can be made robust to noisy labels by restricting the output of a network to a permutation set of any fixed vector. According to this principle, subsequently, we proposed a simple but effective strategy for robust learning through sparse regularization, which is the approximation of the constraint of one-hot permutation. The meanings of sparse regularization are two-fold: the network output sharpening operation is designed to enforce the output distribution of a network to be sharp, and the ℓ_p-norm ($p \leq 1$) regularization is tailored to promote the network output to be sparse. Experimental results demonstrated the superior performance of the proposed method over the SOTA methods on both synthetic and real-world datasets. Moreover, we additionally experiment with imbalanced classification, and the results are encouraging, which demonstrates that sparse regularization can also mitigate class imbalance well.

Overall, this paper has investigated the one-hot constraint, *i.e.*, restricting the output to a permutation set over a one-hot vector. In future research, a promising direction is replacing the one-hot vector with a fixed smoothing vector.

Acknowledgments

This work was supported by National Natural Science Foundation of China under Grants 61922027, 62071155 and 61932022.

References

[1] Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR, 2017.

[2] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

[3] Nontawat Charoenphakdee, Jongyeong Lee, and Masashi Sugiyama. On symmetric losses for learning from corrupted labels. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 961–970, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[4] 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.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Aritra Ghosh, H. Kumar, and P. S. Sastry. Robust loss functions under label noise for deep neural networks. In *AAAI*, 2017.

[7] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[8] B. Han, Quanming Yao, T. Liu, Gang Niu, I. Tsang, James T. Kwok, and M. Sugiyama. A survey of label-noise representation learning: Past, present and future. *ArXiv*, abs/2011.04406, 2020.

[9] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018.

[10] 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.

[11] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313. PMLR, 2018.

[12] Davood Karimi, Haoran Dou, Simon K Warfield, and Ali Gholipour. Deep learning with noisy labels: Exploring techniques and remedies in medical image analysis. *Medical Image Analysis*, 65:101759, 2020.

[13] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. Nlnl: Negative learning for noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 101–110, 2019.

[14] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Computer Science Department, University of Toronto, Tech. Rep*, 1, 01 2009.

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[16] Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 4313–4324. PMLR, 2020.

[17] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. Webvision database: Visual learning and understanding from web data. 08 2017.

[18] 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.

[19] Wei Liu, Yu-Gang Jiang, Jiebo Luo, and Shih-Fu Chang. Noise resistant graph ranking for improved web image search. In *CVPR 2011*, pages 849–856. IEEE, 2011.

[20] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *ICML*, 2020.

[21] Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150, 2011.

[22] N. Manwani and P. S. Sastry. Noise tolerance under risk minimization. *IEEE Transactions on Cybernetics*, 43(3):1146–1151, 2013.

[23] Aditya Krishna Menon, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Can gradient clipping mitigate label noise? In *International Conference on Learning Representations*, 2020.

[24] Sebastian Prillo and Julian Eisenschlos. SoftSort: A continuous relaxation for the argsort operator. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7793–7802. PMLR, 13–18 Jul 2020.

[25] Hwanjun Song, Minseok Kim, Dongmin Park, and Jae Gil Lee. Learning from noisy labels with deep neural networks: A survey. 2020.

[26] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[27] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *IEEE International Conference on Computer Vision*, 2019.

[28] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representation*, 2017.

[29] Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, 2018.

[30] Mingyi Zhou, Jing Wu, Yipeng Liu, Shuaicheng Liu, and Ce Zhu. Dast: Data-free substitute training for adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 234–243, 2020.

[31] Xiong Zhou, Xianming Liu, Junjun Jiang, Xin Gao, and Xiangyang Ji. Asymmetric loss functions for learning with noisy labels. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12846–12856. PMLR, 18–24 Jul 2021.