Joint Negative and Positive Learning for Noisy Labels

Youngdong Kim Juseung Yun Hyounguk Shon Junmo Kim
School of Electrical Engineering, KAIST, South Korea
{ydkim1293, juseung_yun, hyounguk.shon, junmo.kim}@kaist.ac.kr

Abstract

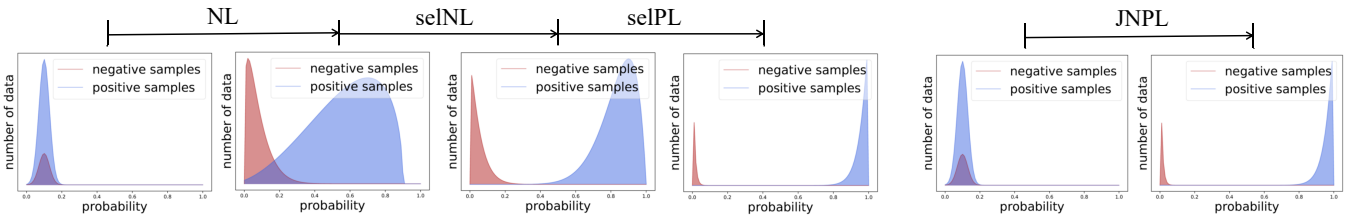
Training of Convolutional Neural Networks (CNNs) with data with noisy labels is known to be a challenge. Based on the fact that directly providing the label to the data (Positive Learning; PL) has a risk of allowing CNNs to memorize the contaminated labels for the case of noisy data, the indirect learning approach that uses complementary labels (Negative Learning for Noisy Labels; NLNL) has proven to be highly effective in preventing overfitting to noisy data as it reduces the risk of providing faulty target. NLNL further employs a three-stage pipeline to improve convergence. As a result, filtering noisy data through the NLNL pipeline is cumbersome, increasing the training cost. In this study, we propose a novel improvement of NLNL, named Joint Negative and Positive Learning (JNPL), that unifies the filtering pipeline into a single stage. JNPL trains CNN via two losses, NL+ and PL+, which are improved upon NL and PL loss functions, respectively. We analyze the fundamental issue of NL loss function and develop new NL+ loss function producing gradient that enhances the convergence of noisy data. Furthermore, PL+ loss function is designed to enable faster convergence to expected-to-be-clean data. We show that the NL+ and PL+ train CNN simultaneously, significantly simplifying the pipeline, allowing greater ease of practical use compared to NLNL. With a simple semi-supervised training technique, our method achieves state-of-the-art accuracy for noisy data classification based on the superior filtering ability.

1. Introduction

Convolutional Neural Networks (CNNs) have led to great improvements in many supervised tasks. However, CNNs' performance relies heavily on the quality of labels, and accurately labeling a huge amount of data is expensive and time-consuming. Furthermore, accurate labeling is done by hand, which can eventually lead to mismatched labeling. Therefore, the robust training of CNNs with noisy data is of great practical importance. There are many approaches regarding this issue. For example, there are methods that design noise-robust loss [4, 3, 29, 18], use two neural networks to select clean labels [6, 33, 30], and utilize label correction [22, 31]. These existing approaches commonly use the given labels in a direct manner, i.e., "input image belongs to this label" (*Positive Learning; PL*). This behavior carries the risk of providing faulty information to the CNNs when noisy labels are involved.

Motivated by this reason, *Negative Learning for Noisy Labels; NLNL* [12], which is an indirect learning method for training CNNs, has been proposed recently. *Negative Learning* (NL) uses randomly chosen complementary labels and trains the CNN that "input image does not belong to this complementary label," reducing the risk of providing the wrong information because of the high chance of not selecting a true label as a complementary label. Additionally, NLNL proposed three-stage pipeline for filtering noisy data from training data (Figure 1 (a)). Each stage is composed of NL $\rightarrow$ NL while discarding data of low confidence (*Selective NL; SelNL*) $\rightarrow$ PL while only retaining data of high confidence (*Selective PL; SelPL*), enabling more convergence after NL. However, the fundamental problem that NL loss function causes underfitting to the overall training data still remains. This is the reason that NL requires an additional sequential step, SelNL. Furthermore, the three-stage pipeline for filtering noisy data is quite inefficient, extending the time for training CNNs.

In this study, we propose a novel version of NLNL: *Joint Negative Learning and Positive Learning; JNPL* which has a unified single-stage pipeline for filtering noisy data (Figure 1 (b)). JNPL is composed of two losses to train CNN, NL+ and PL+ losses, dedicated to filtering noisy data from training data. Each is developed from NL and PL loss functions, respectively. Firstly, our paper focuses on analyzing the NL loss function to understand the cause for underfitting. Then we develop a new loss function NL+ that resolves the issue, which produces a gradient appropriate for convergence on a noisy training dataset. Our study demonstrates the effectiveness of NL+, showing improved convergence across various label noise types and noise rates. Secondly, while we utilize PL to aid in training with noisy data, PL+ loss function is also newly designed to enable faster



(a) NLNL (b) JNPL

Figure 1: Comparison between Negative Learning for Noisy Labels (NLNL) and Joint Negative and Positive Learning (JNPL) for filtering noisy data from training data, demonstrated with histograms showing the distribution of noisy training data . (a): NLNL is a 3-stage pipeline (NL→SelNL→SelPL). (b): JNPL is a single-stage pipeline, in which two loss functions (NL+ and PL+) train CNN simultaneously.

training with expected-to-be-clean data. Our paper shows the effectiveness of the PL+ loss function compared to the previous PL loss function. Finally, as both loss functions of our method (NL+ and PL+) jointly train the model through a *single stage*, it is simple and easier to use than NLNL. Our experiments show that JNPL successfully filters noisy data in a single stage, thereby providing significantly faster training of CNN as well as better filtering compared to NLNL.

After filtering noisy data from the training data we perform pseudo-labeling for noisy data classification. We achieve state-of-the-art accuracy across various settings in CIFAR10, CIFAR100 [13], and Clothing1M [31] datasets, proving the superior filtering ability of JNPL.

The main contributions of this paper are as follows:

- We propose an improved version of NLNL, named *"Joint Negative and Positive Learning (JNPL),"* featuring a single-stage pipeline for filtering noisy data, therefore enabling easier usage compared to NLNL.
- Two novel loss functions are newly designed, each named NL+ loss and PL+ loss. NL+ solves the underfitting problem of the NL loss, and provides better convergence on various types and ratios of label noises in the training data. Moreover, PL+ enables faster training compared to the previous PL loss function.
- Our method filters noisy data, more robust across different types and ratios of noise than NLNL. Our method also achieves state-of-the-art noisy data classification results when used along with pseudo-labeling.
- Prior knowledge of the type or number of noisy data is not required for our method. It does not require any hyper-parameter tuning that depend on prior knowledge, allowing our method to be applicable in practice.

The remainder of this paper is organized as follows. Section 3 describes NLNL method in depth, which is targeted throughout the whole paper, and discusses the cause of the underfitting problem of the method. Section 4 describes our proposed method, JNPL, and explains in detail on NL+ loss and PL+ loss terms. Section 5 demonstrates the overall comparison between JNPL and NLNL, showing the distinct advantages of JNPL over NLNL. Section 6 discusses the evaluations of our method in comparison to baseline methods. Finally, we summarize and conclude in Section 7.

2. Related works

Several methods that aim to mitigate label noise have been proposed. Here, we summarize some of the recent approaches to noise-robust learning.

Designing noise-robust loss The commonly used cross-entropy (CE) loss is known to be prone to overfitting when there is noise in the labels. Therefore, a family of studies aims to design novel loss functions that are tolerant of label noise. Ghosh *et al*. [4, 3] showed that the mean absolute error (MAE) loss is theoretically robust against label noise. Zhang *et al*. [35] proposed Generalized Cross Entropy loss, which is a generalized function that can interpolate between the forms of CE and MAE, which enables it to adjust trade-offs between robust loss and non-robust loss.

However, in many cases, such noise-robust losses carry the problem of underfitting, which motivates the combination of a robust loss with a non-robust loss to improve convergence. Wang *et al*. [29] proposed Symmetric Cross Entropy loss, which combines CE loss with Reverse Cross Entropy loss. Recently, Ma *et al*. [18] proposed a loss normalization technique that transforms a non-robust loss function into a robust loss function. They also showed that such normalized loss used in combination with another robust loss function improves convergence and coined the term Active Passive Loss (APL).

Weighting samples In some researches, each sample in the training set is weighted by the reliability of the label [10, 24, 15]. Moreover, other methods proposed meta-learning algorithms that predicts the weights for each sample [10, 24]. However, these methods require a clean validation set, which is often difficult to guarantee in practice.

Correction methods Some other researches used correction methods [21, 27, 9, 31, 28, 17]. They assume that

prior knowledge like noise rate or noisy transition matrix is known or that some clean data is accessible. However, in a practical case, prior knowledge and clean data is usually hard to obtain. Some other works used CNN with additional layer [25, 11, 5], and noise transition matrix is approximated to correct loss. Many efforts gradually change the data label to the prediction value of the network [23, 26, 19, 32]. Arazo *et al*. [1], fits a mixture of beta distributions that models the loss of clean and noisy samples during training.

Selecting clean labels Some attempted to identify clean labels from a noisy dataset [6, 2, 20]. Ding *et al*. [2] proposed a selection of clean examples based on predicted likelihoods. The labels of the remaining samples are discarded, and the network is trained by semi-supervision. Some of the successful approaches train two deep neural networks simultaneously and let them teach each other [6, 33, 30]. Each network selects possibly clean data and trains the other network with this data.

Use of complementary labels Kim *et al*. [12] proposed a noise-robust learning method where instead of maximizing the log-likelihood on the target position, it minimizes the log-likelihood on the complementary positions, termed Negative Learning (NL). They employ a three-stage pipeline based on NL that separates the clean data from the noisy data. Finally, the network is trained using standard CE loss with semi-supervision by treating the noisy set as unlabeled.

Other approaches Li *et al*. [16] uses meta-learning to obtain weights that can be easily fine-tuned to a given noisy dataset. Zhang *et al*. [34] proposed to learn confidence scores of each samples from the relationship between noisy samples in the feature space, then use the confidence scores to generate cleaner representations. Harutyunyan *et al*. [7] proposed training algorithm based on mutual information between weights and labels to regularize the memorization of labels.

3. Negative Learning for Noisy Labels (NLNL)

Throughout this paper, we consider the problem of c-class classification. Let $\boldsymbol{x} \in \mathcal{X}$ be an input, $y, \overline{y} \in \mathcal{Y} = \{1, ..., c\}$ be its label and complementary label, respectively, and $\boldsymbol{y}, \overline{\boldsymbol{y}} \in \{0, 1\}^c$ be their one-hot vector. Suppose the CNN $f(x; \theta)$ maps the input space to the c-dimensional score space $f : \mathcal{X} \rightarrow \mathbb{R}^c$, where θ is the set of network parameters. If f passes through the softmax function, the output can be interpreted as a probability vector $\boldsymbol{p} \in \Delta^{c-1}$, where Δ^{c-1} denotes the c-dimensional simplex.

NL [12] is an indirect learning method for training CNNs with noisy data. Instead of using given labels, it chooses random complementary label $\overline{y}$ and train CNNs as in "input image does not belong to this complementary label." The loss function following this definition is as below, along with the classic PL loss function for comparison:

$$\mathcal{L}_{PL}(f, y) = -\sum_{k=1}^{c} \boldsymbol{y}_k \log \boldsymbol{p}_k \tag{1}$$

$$\mathcal{L}_{NL}(f, \overline{y}) = -\sum_{k=1}^{c} \overline{\boldsymbol{y}}_k \log(1 - \boldsymbol{p}_k). \tag{2}$$

To improve convergence after NL, SelNL is performed as a subsequent step. SelNL trains the CNNs only with the data having confidence over $\frac{1}{c}$ ($\boldsymbol{p}_y > \frac{1}{c}$). Since data involved in training tend to be less noisy than before, CNNs converge better after SelNL. Furthermore, PL is considered a faster and more accurate method than NL, only if training data is assumed to be clean. After training with NL and SelNL, SelPL train CNNs only with data that has confidence above γ ($= 0.5$), assuming that such data are clean. After filtering noisy data with these three steps (NL→SelNL→SelPL), semi-supervised learning (pseudo-labeling [14]) is performed utilizing labeled expected-to-be-clean data and unlabeled noisy data.

As mentioned in Section 1, the fundamental problem of underfitting of NL still remains. To analyze the root of this phenomenon, we observe the gradient resulting from the NL loss function (Eq 2) as follows:

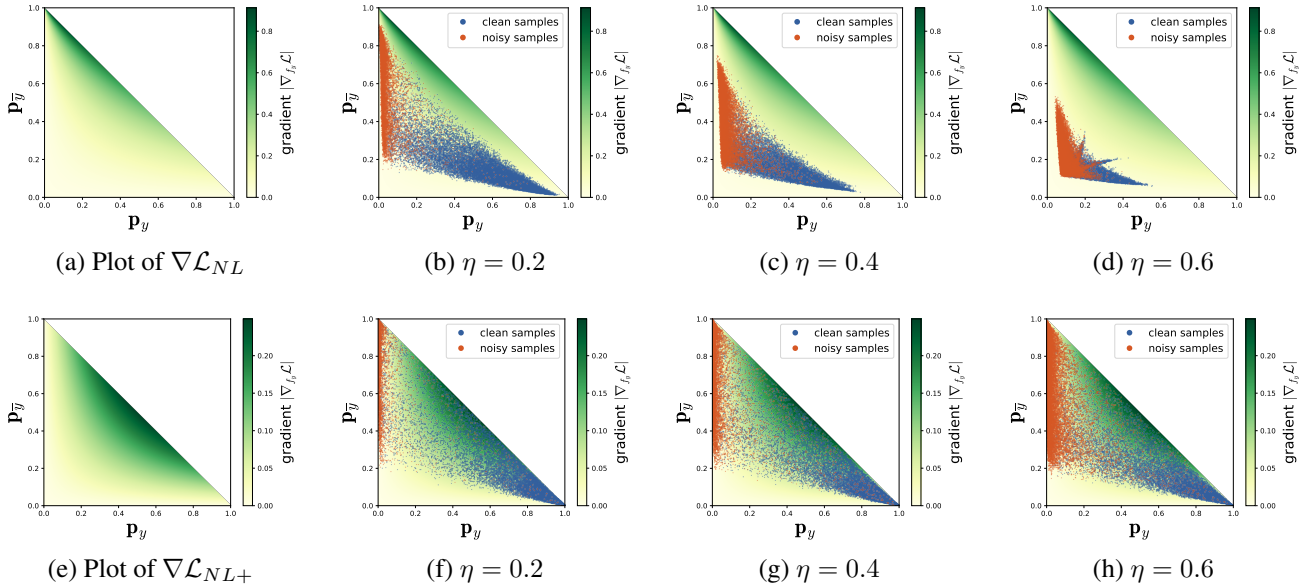
$$\nabla \mathcal{L}_{NL} = \frac{\partial \mathcal{L}_{NL}(f, \overline{y})}{\partial f_i} = \begin{cases} \boldsymbol{p}_i & \text{if } i = \overline{y} \\ -\frac{\boldsymbol{p}_{\overline{y}}}{1 - \boldsymbol{p}_{\overline{y}}} \boldsymbol{p}_i & \text{if } i \neq \overline{y}. \end{cases} \tag{3}$$

Eq 3 states that at classes except for $\overline{y}$ receives gradient of $-\frac{\boldsymbol{p}_{\overline{y}}}{1 - \boldsymbol{p}_{\overline{y}}} \boldsymbol{p}_i$ ($\nabla \mathcal{L}_{NL(i \neq \overline{y})}$). Figure 2 (a) shows 2D gradient map of $\nabla \mathcal{L}_{NL(i \neq \overline{y})}$, and Figure 2 (b)-(d) shows the distribution of training data after NL in diverse noise ratio. Each training data is distributed in gradient map with respect to its $\boldsymbol{p}_y$ (when $i = y$) and $\boldsymbol{p}_{\overline{y}_{max}}$. As the training with NL progresses, clean data tend to have high $\boldsymbol{p}_y$ and low $\boldsymbol{p}_{\overline{y}}$ (lower-right region in Figure 2 (a)), while noisy data tend to have low $\boldsymbol{p}_y$ and high $\boldsymbol{p}_{\overline{y}}$ (upper-left region in Figure 2 (a)). However, considering noisy data, ground-truth labels may be chosen as $\overline{y}$. In this case, all classes, except for ground truth label, receive high $\nabla \mathcal{L}_{NL(i \neq \overline{y})}$ because of high $\boldsymbol{p}_{\overline{y}}$, resulting in underfitting of that data as the confidence of classes other than the ground-truth label increases. In Section 4.1, we describe the developed loss function of NL (NL+) that resolves this underfitting issue.

4. Joint Negative and Positive Learning (JNPL)

The loss function of the proposed method, JNPL, is composed of two loss functions:

$$\mathcal{L}_{JNPL} = \mathcal{L}_{NL+} + \lambda \mathcal{L}_{PL+}. \tag{4}$$



(a) Plot of $\nabla\mathcal{L}_{NL}$ (b) $\eta = 0.2$ (c) $\eta = 0.4$ (d) $\eta = 0.6$

(e) Plot of $\nabla\mathcal{L}_{NL+}$ (f) $\eta = 0.2$ (g) $\eta = 0.4$ (h) $\eta = 0.6$

Figure 2: Comparison between NL and NL+ with CIFAR10 with *symm* noise. (a), (e): Gradient map of NL and NL+, respectively. (b)-(d): Training data distribution with 20%, 40%, 60% noise after training with NL. (f)-(h): Training data distribution with 20%, 40%, 60% noise after training with NL+

Each of which is dedicated to filtering noisy data from training data. $\mathcal{L}_{NL+}$ is the advanced version of NL, which resolves underfitting issue. $\mathcal{L}_{PL+}$ is other newly designed loss for PL that trains on expected-to-be-clean data, empowering training on data of higher confidence. λ is added to scale the overall magnitude of PL+ so that it does not overwhelm the magnitude of NL+. We set $\lambda = 0.01$ throughout the whole paper. These two losses enable successful filtering of noisy data. Finally, noisy data classification is done in semi-supervised manner, utilizing these filtered noisy data confidence as pseudo-label. In the following sections, we further introduce each of the loss functions and describe the concept and implementation respectively.

4.1. NL+

As discussed in Section 3, we argue that the cause of the underfitting problem with NL is due to the nature of its gradient $\nabla\mathcal{L}_{NL(i\neq\overline{y})}$ (Figure 2 (a)). This is more pronounced as the noise rate increases, as shown in Figure 2 (b)-(d). This problem occurs when noisy data receives high gradient to classes except for $\overline{y}$ when the confidence of $\overline{y}$ is high, $\overline{y}$ being most likely to be ground truth label. To solve this issue, we propose a modification to the NL loss function, named NL+ loss, as follows:

$$\mathcal{L}_{NL+}(f, \overline{y}) = -(1 - \boldsymbol{p}_{\overline{y}}) \sum_{k=1}^{c} \overline{\boldsymbol{y}}_k \log(1 - \boldsymbol{p}_k). \quad (5)$$

It should be noted that $\left(1 - \boldsymbol{p}_{\overline{y}}\right)$ acts as a constant weighting factor. Intuitively, this factor has the effect of decreasing the loss for noisy data when corresponding $\boldsymbol{p}_{\overline{y}}$ is high, $\overline{y}$ being most likely to be ground truth label. That way, it reduces the risk of pressing down on the confidence of ground truth label for noisy data, reducing the risk of underfitting. This is further analyzed by observing the gradient of NL+ ($\nabla\mathcal{L}_{NL+(i\neq\overline{y})}$), given by Eq 5:

$$\nabla\mathcal{L}_{NL+(i\neq\overline{y})} = (1 - \boldsymbol{p}_{\overline{y}})\nabla\mathcal{L}_{NL(i\neq\overline{y})} = -\boldsymbol{p}_{\overline{y}}\boldsymbol{p}_i. \quad (6)$$

The gradient map of $\nabla\mathcal{L}_{NL+(i\neq\overline{y})}$ is shown in Figure 2 (e). Compared to Figure 2 (a), it shows gradient at upper-left region is reduced. This implies that as the training progresses with NL+, noisy data is gathered at the upper-left region. With NL+, gradient received for noisy data of high $\boldsymbol{p}_{\overline{y}}$ is reduced, allowing noisy data to maintain high $\boldsymbol{p}_{\overline{y}}$ value, where $\overline{y}$ is most likely to be ground truth label. Figure 2 (f)-(h) shows the distribution of training data mixed with diverse ratio of noise. It shows that compared to Figure 2 (b)-(d), NL+ results in more convergence. Especially in noise of high ratio (Figure 2 (d), (h)), NL+ successfully divides noisy data from training data, sending noisy data to upper-left region.

4.2. PL+

In this section, we introduce the second loss function $\mathcal{L}_{PL+}$ in JNPL. As mentioned in Section 1, when training data is verified to have clean labels, PL is a faster and more accurate method than NL. Following this fact, we apply PL+ to our method for faster convergence. But compared to NLNL, this is not applied in a sequential step but

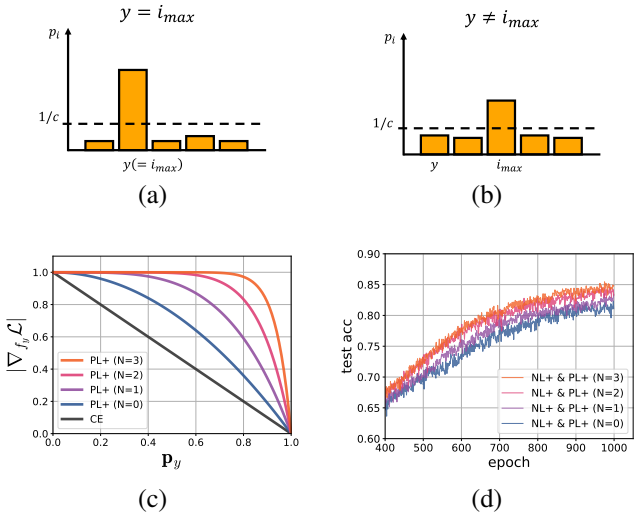


Figure 3: (a), (b): Cases for selecting data for PL+. Data is the candidate for PL+ if confidences at classes other than label of maximum probability is under uniform distribution $(1/c)$. (c): Gradient of PL+ depending on N compared to PL (cross-entropy loss). (d): Accuracy comparison between PL+ with different N. This shows that the flatter version of PL+ ($N = 3$) generates better training results.

rather as a unified step.

First of all, the criteria for selecting the training data for PL+ is required. Previously, NLNL applied PL to data over the threshold ($\boldsymbol{p}_y > 0.5$). However, the criteria for selecting data for PL should be stricter. Even if a data satisfies $\boldsymbol{p}_y > 0.5$, a probability of other class may reach as much as 0.5, resulting in the risk of selecting noisy data as clean data. Hence, PL+ considers the probabilities of classes other than the given label. When probabilities of other classes except for given label are under uniform distribution $\frac{1}{c}$, this data is a candidate for PL+ (Figure 3 (a)). Additionally, among the candidates for PL+, it is selected through Bernoulli sampling with respect to $\boldsymbol{p}_y$. The higher the $\boldsymbol{p}_y$, the more frequently the data would be trained with PL+. Furthermore, PL+ selects data not only from expected-to-be-clean data but also from noisy data. Meaning that, when the probabilities of other classes except for the label of maximum probability is under the uniform distribution, the data is also a candidate for PL+ using the maximum probability class label ($= \hat{y}$) (Figure 3 (b)). In this way, PL+ selects data for training more strictly, but also, the candidate area is increased. The pseudocode for PL+ process is shown in Algorithm 1.

PL is usually done using cross-entropy (CE) loss (Eq 1). However, while it may be tolerable when training clean data, it may not be as tolerable as when training noisy data. The reason for PL in our method is to train faster on more confident data. However, when observing the gradient of CE in Figure 3 (c), it states that a smaller gradient is provided to more confident data, while a higher gradient is provided to less confident data. Since the goal is to train faster on more confident data, not just training more on less confident data, we propose PL+ loss function to resolve this issue as follows:

Algorithm 1: PL+

Input: mini-batch $\bar{\mathcal{D}}$
Result: $\mathcal{L}_{PL+}$ over mini-batch $\bar{\mathcal{D}}_{PL+}$
for $(\boldsymbol{x}, y) \in \bar{\mathcal{D}}$ **do**
  $\boldsymbol{p} \leftarrow softmax\,(f(\boldsymbol{x}))$
  $\hat{y} \leftarrow \text{argmax}_i\, \boldsymbol{p}_i$
  if $\boldsymbol{p}_i < \frac{1}{c}$ *for* $\forall i \in \{1, ..., c\} \setminus \{\hat{y}\}$ **then**
    Append $(\boldsymbol{x}, \hat{y})$ to $\bar{\mathcal{D}}_{PL+}$ with probability $\boldsymbol{p}_{\hat{y}}$
  else
    Reject $(\boldsymbol{x}, y)$
  end
end
Calculate $\mathcal{L}_{PL+}\,(f(\boldsymbol{x}), \hat{y})$ for $\bar{\mathcal{D}}_{PL+}$ by Eq. (7)
return $\frac{1}{|\bar{\mathcal{D}}_{PL+}|} \sum_{x \in \bar{\mathcal{D}}_{PL+}} \mathcal{L}_{PL+}\,(f(\boldsymbol{x}), \hat{y})$

$$\mathcal{L}_{PL+}(f, \hat{y}) = -\prod_{n=0}^{N}(1 + \boldsymbol{p}_{\hat{y}}^{2^n}) \sum_{k=1}^{c} \boldsymbol{y}_k \log \boldsymbol{p}_k, \qquad (7)$$

and the gradient of PL+ loss is as follows:

$$\begin{aligned}\nabla\mathcal{L}_{PL+} &= \prod_{n=0}^{N}(1 + \boldsymbol{p}_{\hat{y}}^{2^n})\nabla\mathcal{L}_{PL} \\ &= -\prod_{n=0}^{N}(1 + \boldsymbol{p}_{\hat{y}}^{2^n})(1 - \boldsymbol{p}_{\hat{y}}) = -(1 - \boldsymbol{p}_{\hat{y}}^{2^{N+1}}).\end{aligned} \qquad (8)$$

Similar to NL+, $\prod_{n=0}^{N}(1+\boldsymbol{p}_{\hat{y}}^{2^n})$ acts as a constant weighting factor. By applying this weight factor, the gradient of PL+ loss function is modified as shown in Eq 8 and visualized in Figure 3 (c). It can be seen that higher gradient is being provided to data of high $\boldsymbol{p}_y$ as N increases. Figure 3 (d) proves faster convergence as N increases. We set $N = 3$ throughout the whole paper.

5. Analysis

Since our method is the advanced version of NLNL, which is targeted throughout our whole paper, this section further demonstrates the distinct advantage of our method JNPL over NLNL.

First of all, our method JNPL is a unified step pipeline for filtering noisy data, compared to 3-step pipeline of NLNL. JNPL is trained with two loss functions simultaneously, increasing the efficiency of training CNN. Figure 4 shows the performance comparison between NLNL

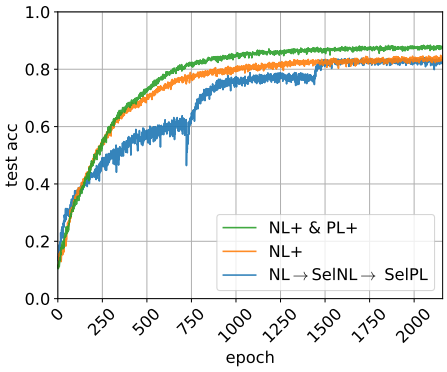


Figure 4: Accuracy graph of NLNL, NL+, and JNPL (NL+ & PL+) with CIFAR10 mixed with 60% *symm* noise.

(NL→SelNL→SelPL), NL+, and JNPL (NL+&PL+) when training with CIFAR10 mixed with 60% *symm* noise. Figure 4 clearly indicates that NL+ solely reaches the accuracy of NL→SelNL, proving better convergence of NL+ compared to NL. Furthermore, when PL+ is done simultaneously along with NL+, it results in faster training without the need for additional subsequent step. It also shows overall accuracy of NL+ and JNPL overpasses the accuracy reached by NLNL while preventing overfitting to noisy data, proving the superiority of our method over NLNL.

Secondly, NL+ is more capable of handling more diverse noise types compared to NL→SelNL owing to the nature of gradient followed by $\mathcal{L}_{NL+}$. Although NL applies SelNL to compensate for underfitting problem, we show that this is not an optimal solution for all types of noise. Consider when training data is CIFAR10 mixed with *asymm* noise, especially when class "dog" is mixed with "cat" in bidirectional manner (DOG $\leftrightarrow$ CAT). Overall probability values across all classes are shared between class "dog" and "cat," resulting in distribution of training data as shown in Figure 6 (a), (d). In this case, SelNL shows almost no effect as the noisy data is not under the uniform distribution (Figure 6 (b), (e)). Whereas for NL+, due to the fact that gradient for region ($\boldsymbol{p}_y < 0.5$ & $\boldsymbol{p}_{\overline{y}} > 0.5$) is reduced in a smooth manner compared to NL, it eventually enables both classes to be separated, showing distinct advantage of NL+ over SelNL (Figure 6 (c), (f)).

Finally, we show that our method JNPL successfully filters noisy data from training data than NLNL. Figure 5 shows overall filtering ability between NLNL and JNPL with average precision (AP). It is compared in diverse environment: CIFAR10/CIFAR100 mixed with different ratio of *symm* and *asymm* noise. It shows that our method outperforms NLNL in filtering noisy data on overall cases. Furthermore, it can be observed that gap of AP between NLNL and JNPL increases as the noise ratio increases. This implies that JNPL is more robust to the amount of noise mixed

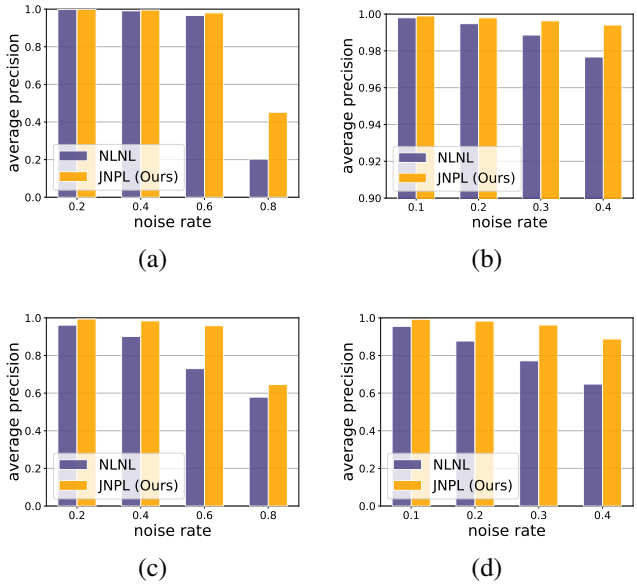


Figure 5: Average Precision (AP) for CIFAR10 / CIFAR100 on *symm* / *asymm* noises. (a), (b): AP for CIFAR10 on *symm* / *asymm* noises, respectively. (c), (d): AP for CIFAR100 on *symm* / *asymm* noises, respectively.

in training data. Also, JNPL being more robust to *asymm* noise than NLNL also proves the point made above. This phenomenon is more clearly shown in more difficult data CIFAR100. AP of NLNL drastically decreases as the noise rate gets higher. However, JNPL shows robustness in types and ratios of noise, similar to when training with CIFAR10. Figure 5 demonstrates our method JNPL is capable of being generalized to type and ratio of noise, and even number of classes in the dataset.

6. Experiments

In this section, we describe the experiments performed to evaluate our method. Pseudo-labeling is done on a training dataset filtered by JNPL for noisy data classification and resulting accuracies are compared to those of other existing methods. We verify our method by comparing with other recent baseline methods, varying experimental settings in terms of dataset and type and ratio of noise in the training data.

6.1. Experiment settings

Baseline methods We compare our method against CE, along with recent state-of-the-art approaches including Co-teaching [6], JoCoR [30], APL [18], and NLNL [12].
Dataset We conduct the experiments on CIFAR10, CIFAR100 [13] mixed with two types noises (*symm*, *asymm*), and Clothing1M [31] dataset. Clothing1M is a large-scale real-world dataset with noisy labels, containing 1 million images of clothing obtained from several online shopping

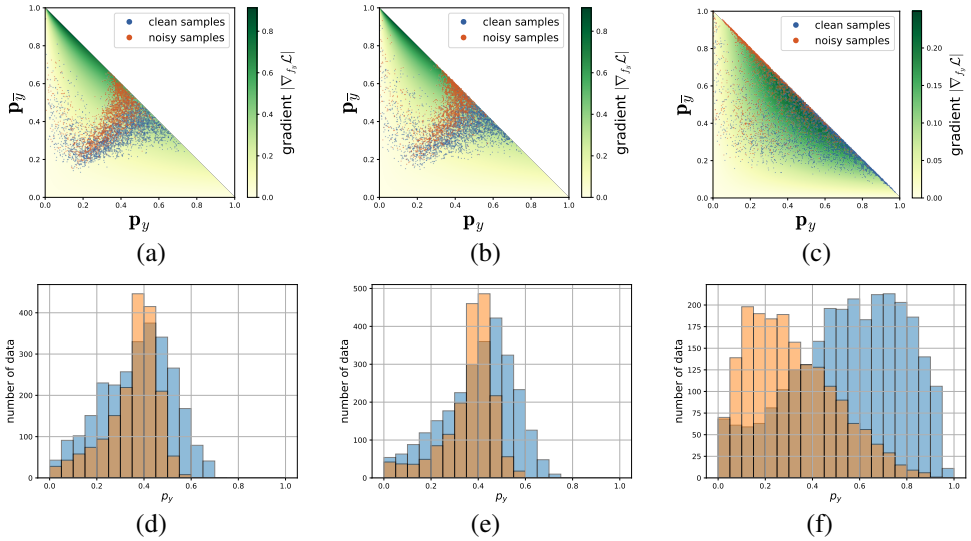


Figure 6: Comparison between NL and NL+ for *asymm* 40% noise CIFAR10 "cat" class. (a), (d): Gradient map and histogram of NL, respectively. (b), (e): Gradient map and histogram of NL→SelNL, respectively. (c), (f): Gradient map and histogram of NL+, respectively. Blue indicates clean data whereas orange indicates noisy data in histograms.

websites. It is reported that the overall accuracy of noisy labels in this dataset is 61.54%, and some pairs of classes are often confused with each other (*e.g.*, Knitwear and Sweater). For preprocessing, we performed mean subtraction, horizontal flip, and random crops for CIFAR10 and CIFAR100. For Clothing1M, we resize the image to 256×256, crop 224×224 at the center and perform mean subtraction and horizontal flip.

Label noise types We generated noisy CIFAR10 and CIFAR100 datasets according to the following procedures. In symmetric (*symm*) noise experiments, we flipped a portion of the labels by re-sampling each label uniformly from the remaining classes, excluding the ground-truth class. In asymmetric (*asymm*) noise experiments, we followed the same label transition rule used by Patrini *et al*. [22]. For CIFAR10, we mapped TRUCK → AUTOMOBILE, BIRD → PLANE, DEER → HORSE, and CAT ↔ DOG. For CIFAR100, the noise flipped each class into the next, circularly within super-classes.

For each noise type, we compared the methods under the symmetric noise rates of $\eta_{symm} \in \{0.2, 0.4, 0.6, 0.8\}$ and asymmetric noise rates of $\eta_{asymm} \in \{0.1, 0.2, 0.3, 0.4\}$.

Models For CIFAR10 and CIFAR100 experiments, we used ResNet34. For Clothing1M, we used ResNet50 [8], pretrained on ImageNet.

Hyperparameters We used stochastic gradient descent (SGD) with momentum of 0.9, weight decay of 10^{-4}. For experiments with CIFAR10 and CIFAR100, batch size is set to 128. Moreover, JNPL trains CNN for 1000 epochs with initial learning rate of 10^{-2}, and decay by a factor of 10 at 800 epochs. For pseudo labeling, initial learning rate is 0.1, decayed by a factor of 10 at 192, 288 epochs (480 epochs total). For experiments with Clothing1M, batch size is set to 64, and JNPL trains CNN for 40 epochs with initial learning rate of 10^{-3}, and decay by a factor of 10 at 30 epochs. For pseudo labeling, initial learning rate is 10^{-3}, decayed by a factor of 10 at 10 epochs (15 epochs total).

For CIFAR100, we adopt the technique NLNL proposed for generalization to the number of classes in training data: providing multi $\overline{y}$ to each data. We provide 110 $\overline{y}$ to each data in order to match the training speed to when training with CIFAR10 [12].

6.2. Results

Table 1 shows the results of our method and other baseline methods in various noise environment and two datasets. Our proposed method outperformed all other comparable baseline methods in overall noise types and ratios. The result shows other baseline methods achieve comparable results in the less-noisy environment, but the performance decreases drastically as the noise ratio increases, which is even more visible at CIFAR100, which is the harder case for noisy data classification. Our method shows a distinct improvement in this situation compared to all other methods. It was shown in Section 5 our method is robust to the amount of noise mixed in training data, regardless of the type of noises. Table 1 shows a similar result that our method achieves more distinct best accuracy as the noise rate gets higher. This phenomenon is more emphasized for CIFAR100. Our method outperforms as much as 6 to 7% at both *symm* and *asymm* noises in this dataset. It is noteworthy that our method achieved 7% higher state-of-the-art

Datasets	Model	Methods	*Symm*				*Asymm*			
			20	40	60	80	10	20	30	40
CIFAR10	ResNet34	CE	83.95	67.58	43.55	17.32	91.39	87.67	82.73	76.37
		Co-teaching [6]	91.08	88.08	80.96	21.13	94.20	93.24	90.67	70.20
		JoCoR [30]	91.84	88.15	59.20	20.72	93.13	91.19	89.01	83.61
		NFL+RCE [18]	90.50	85.16	70.77	19.67	92.35	89.66	84.92	78.30
		NCE+RCE [18]	90.36	84.57	74.09	**26.71**	91.89	90.13	85.80	78.49
		NLNL [12]	**94.23**	**92.43**	**88.32**	-	**94.57**	**93.35**	**91.80**	**89.86**
		Ours	**93.53**	**91.89**	**88.45**	**35.65**	**94.22**	**93.45**	**92.47**	**90.72**
CIFAR100	ResNet34	CE	57.32	45.64	24.30	8.06	65.12	62.12	52.77	44.55
		Co-teaching [6]	69.56	62.81	51.12	10.25	**72.52**	**67.46**	**61.50**	52.86
		JoCoR [30]	**71.75**	63.96	37.84	7.32	72.01	65.05	56.63	45.14
		NFL+RCE [18]	58.70	42.76	24.77	**10.57**	63.70	56.45	46.96	37.52
		NCE+RCE [18]	57.41	43.75	25.87	9.94	64.24	56.48	47.17	36.40
		NLNL [12]	**71.52**	**66.39**	**56.51**	-	70.35	63.12	54.87	**45.70**
		Ours	70.94	**68.11**	**61.26**	**17.55**	**72.03**	**69.95**	**68.12**	**59.51**

Table 1: Comparison with other baseline methods on CIFAR10, CIFAR100 mixed with various types and ratios of noise. Best 2 accuracies are **bold faced**.

Method	Test Accuracy
CE	69.21
Forward [21]	69.84
M-correction [1]	71.00
LIMIT [7]	71.39
Joint-Optim [26]	72.16
MetaCleaner [34]	72.5
MLNT [16]	73.47
PENCIL [32]	73.49
Ours	**74.15**

Table 2: Comparison on Clothing1M with other baseline methods.

accuracy in the most difficult setting in Table 1, which is 100 class dataset mixed with 40% *asymm* noise. It is widely known training in general is challenging as the number of classes in the dataset increases. Furthermore, compared to *symm* noise, *asymm* noise is the replica of noise that we can actually make in real-life. Achieving such a high accuracy in this setting implies that our method is more capable of generalizing to training data and various types and ratios of noise mixed within compared to other baseline methods.

It is shown that Co-teaching and JoCoR method [6, 30] exceeds the performance compared to our method for some cases. However, it should be noted that they assume prior knowledge on important statistics about the dataset such as the amount of noise. In reality, this assumption often leaves the method impractical because the ratio of noise mixed in training data is likely to be unknown. On the other hand, our method does not assume any such prior knowledge and therefore does not require extensive tuning of hyper-parameters.

To demonstrate the generalization of our method JNPL to real-world noisy data, we compose an experiment on Clothing1M dataset (Table 2). We brought recent baseline methods which conducted experiment on Clothing1M for comparison. It shows our method achieves comparable performance, outperforming other recent baseline methods. This result clearly proves that JNPL can generalize to training data mixed with various types and ratios of noise, showing the novelty of our method.

7. Conclusion

We propose Joint Negative and Positive Learning, the next version of NLNL which is the novel single-step pipeline for filtering noisy training data. Compared to 3-step pipeline of NLNL, our method trains CNN with two-loss functions ($\mathcal{L}_{NL+} + \mathcal{L}_{PL+}$) in one step. They are developed from previous NL and PL loss functions to enhance convergence and training speed, resulting in better filtering performance than NLNL. We demonstrated that JNPL is stable and robust in various types and ratios of noise mixed in training data. Our method achieves state-of-the-art performance in noisy data classification utilizing pseudo-labeling to our filtered training data, proving our method's excellent filtering ability without referring to any prior knowledge.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. *arXiv preprint arXiv:1904.11238*, 2019. 3, 8

[2] Yifan Ding, Liqiang Wang, Deliang Fan, and Boqing Gong. A semi-supervised two-stage approach to learning from noisy labels. *arXiv preprint arXiv:1802.02679*, 2018. 3

[3] Aritra Ghosh, Himanshu Kumar, and PS Sastry. Robust loss functions under label noise for deep neural networks. In *AAAI*, pages 1919–1925, 2017. 1, 2

[4] Aritra Ghosh, Naresh Manwani, and PS Sastry. Making risk minimization tolerant to label noise. *Neurocomputing*, 160:93–107, 2015. 1, 2

[5] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. 2016. 3

[6] 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. In *Advances in neural information processing systems*, pages 8527–8537, 2018. 1, 3, 6, 8

[7] Hrayr Harutyunyan, Kyle Reing, Greg Ver Steeg, and Aram Galstyan. Improving generalization by controlling label-noise information in neural network weights. *arXiv preprint arXiv:2002.07933*, 2020. 3, 8

[8] 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. 7

[9] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in neural information processing systems*, pages 10456–10465, 2018. 2

[10] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Regularizing very deep neural networks on corrupted labels. *arXiv preprint arXiv:1712.05055*, 2017. 2

[11] Ishan Jindal, Matthew Nokleby, and Xuewen Chen. Learning deep networks from noisy labels with dropout regularization. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 967–972. IEEE, 2016. 3

[12] 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. 1, 3, 6, 7, 8

[13] Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009. 2, 6

[14] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013. 3

[15] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. *arXiv preprint arXiv:1711.07131*, 2017. 2

[16] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5051–5059, 2019. 3, 8

[17] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *ICCV*, pages 1928–1936, 2017. 2

[18] 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. 1, 2, 6, 8

[19] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah M Erfani, Shu-Tao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. *arXiv preprint arXiv:1806.02612*, 2018. 3

[20] Curtis G Northcutt, Tailin Wu, and Isaac L Chuang. Learning with confident examples: Rank pruning for robust classification with noisy labels. *arXiv preprint arXiv:1705.01936*, 2017. 3

[21] Giorgio Patrini, Alessandro Rozza, Aditya Menon, Richard Nock, and Lizhen Qu. Making neural networks robust to label noise: a loss correction approach. *arXiv preprint arXiv:1609.03683*, 2016. 2, 8

[22] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR)*, pages 2233–2241, 2017. 1, 7

[23] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014. 3

[24] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*, 2018. 2

[25] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*, 2014. 3

[26] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560, 2018. 3, 8

[27] Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In *Advances in Neural Information Processing Systems*, pages 5596–5605, 2017. 2

[28] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge J Belongie. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, pages 6575–6583, 2017. 2

[29] 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. 1, 2

[30] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with

co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13726–13735, 2020. 1, 3, 6, 8

[31] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015. 1, 2, 6

[32] Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7017–7025, 2019. 3, 8

[33] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? *arXiv preprint arXiv:1901.04215*, 2019. 1, 3

[34] Weihe Zhang, Yali Wang, and Yu Qiao. Metacleaner: Learning to hallucinate clean representations for noisy-labeled visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7373–7382, 2019. 3, 8

[35] Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *arXiv preprint arXiv:1805.07836*, 2018. 2