Understanding and Utilizing Deep Neural Networks Trained with Noisy Labels

Pengfei Chen [1 2] **Benben Liao** [2] **Guangyong Chen** [2] **Shengyu Zhang** [1 2]

Abstract

Noisy labels are ubiquitous in real-world datasets, which poses a challenge for robustly training deep neural networks (DNNs) as DNNs usually have the high capacity to memorize the noisy labels. In this paper, we find that the test accuracy can be quantitatively characterized in terms of the noise ratio in datasets. In particular, the test accuracy is a quadratic function of the noise ratio in the case of symmetric noise, which explains the experimental findings previously published. Based on our analysis, we apply cross-validation to randomly split noisy datasets, which identifies most samples that have correct labels. Then we adopt the Co-teaching strategy which takes full advantage of the identified samples to train DNNs robustly against noisy labels. Compared with extensive state-of-the-art methods, our strategy consistently improves the generalization performance of DNNs under both synthetic and real-world training noise.

1. Introduction

The remarkable success of DNNs on supervised learning tasks heavily relies on a large number of training samples with accurate labels. Correctly labeling extensive data is too costly while alternating methods such as crowdsourcing (Yan et al., 2014; Chen et al., 2017) and online queries (Schroff et al., 2011; Divvala et al., 2014) inexpensively obtain data, but unavoidably yield noisy labels. Training with too many noisy labels reduces generalization performance of DNNs since the networks can easily overfit on corrupted labels (Zhang et al., 2017; Arpit et al., 2017). To utilize extensive noisy data, understanding how noisy labels affect training and generalization of DNNs is the very first step, based on which we can design specific methods to train DNNs robustly in practical applications.

Numerous methods have been proposed to deal with noisy labels. Several methods focus on estimating the noise transition matrix and correcting the objective function accordingly, e.g., forward or backward correction (Patrini et al., 2017), S-model (Goldberger & Ben-Reuven, 2017). However, it is a challenge to estimate the noise transition matrix accurately. An alternative approach is training on selected or weighted samples, e.g., Decoupling (Malach & Shalev-Shwartz, 2017), MentorNet (Jiang et al., 2018), gradient-based reweighting (Ren et al., 2018) and Co-teaching (Han et al., 2018). A remaining issue is to design a reliable and convincing criteria of selecting or weighting samples. Another approach proposes to correct labels using the predictions of DNNs, e.g., Bootstrap (Reed et al., 2015), Joint Optimization (Tanaka et al., 2018) and D2L (Ma et al., 2018), all of which are vulnerable to overfitting. To improve the robustness, Joint Optimization introduces regularization terms requiring a prior knowledge of how actual classes distribute among all training samples. However, the prior knowledge is usually unavailable in practice.

How noisy labels affect training and generalization of DNNs is not well understood, which deserves more attention since it may promote fundamental approaches of robustly training DNNs against noise. Without label corruption, the generalization error can be bounded by complexity measures such as VC dimension (Vapnik, 1998), Rademacher complexity (Bartlett & Mendelson, 2002) and uniform stability (Mukherjee et al., 2002; Bousquet & Elisseeff, 2002; Poggio et al., 2004). But the bounds become trivial in the presence of noisy labels. Zhang et al. (2017) demonstrated that DNNs have the high capacity to fit even random labels, but obtain a large generalization error. Zhang et al. (2017) also showed a positive correlation between generalization error and noise ratio, which implies DNNs do capture some useful information out of the noisy data. Arpit et al. (2017) showed that during training, DNNs tend to learn simple patterns first, then gradually memorize all samples, which justifies the widely used *small-loss criteria*: treating samples with small training loss as clean ones (Han et al., 2018; Jiang et al., 2018). Ma et al. (2018) qualitatively attributed the

[1]Department of Computer Science and Engineering, The Chinese University of Hong Kong [2]Tencent Technology. Correspondence to: Guangyong Chen <gycchen@tencent.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

poor generalization performance of DNNs to the increased dimensionality of the latent feature subspace. Through extensive experiments, these works gained empirical insight into the interesting behavior of DNNs trained with noisy labels, while a theoretical and quantitative explanation is yet to emerge.

In this paper, we can quantitatively clarify the generalization performance of DNNs normally trained with noisy labels. To verify our theoretical analysis, we apply cross-validation to randomly split a set of collected samples, whose labels may be polluted by some noise. DNNs can be trained on a subset, then evaluated on the remaining dataset to compare the theoretically and empirical results on the generalization performance. We find that DNNs can fit noisy training sets exactly and generalize in distribution (see Claim 1 for more details). Hence, we can quantitatively characterize the test accuracy in terms of noise ratio in datasets. In particular, the test accuracy is a quadratic function of the noise ratio in the case of symmetric noise. In Zhang et al. (2017), it has been empirically found that the generalization performance of DNNs is highly dependent on the noise ratio. One of our contributions is to provide a thorough explanation for their empirical findings.

Based on our analysis, we further develop a specific method to train DNNs against noisy labels. Our method is developed on top of the Co-teaching strategy, which is first presented in Blum & Mitchell (1998) and then modified to deal with noisy labels with impressive performance in (Han et al., 2018). In the Co-teaching strategy, one trains two networks simultaneously: mini-batches are drawn from the whole noisy training set, then each network selects a certain number of small-loss samples and feeds them to its peer network. However, the performance of the Co-teaching decays seriously when the noise ratio of the training set increases. Moreover, the number of small-loss samples selected in each mini-batch is set according to the noise ratio of the training set, which is unavailable in practice. Fortunately, we can address these issues based on our theoretical analysis on the generalization performance of DNNs. Specially, we present the Iterative Noisy Cross-Validation (INCV) method to select a subset of samples, which has much smaller noise ratio than the original dataset, resulting in a more stable training process of DNNs. Moreover, we can automatically estimate the noise ratio of the selected set, which makes our method more practical for industrial applications. Briefly speaking, our main contributions are

- theoretically relating the generalization performance of DNNs to the label noise,
- practical algorithms of selecting clean labels and training noise-robust DNNs.

Experiments on both synthetic and real-world noisy labels show that compared with state-of-the-art methods (Patrini et al., 2017; Malach & Shalev-Shwartz, 2017; Han et al., 2018; Jiang et al., 2018; Ma et al., 2018), DNNs trained using our strategy achieve the best test accuracy on the clean test set. In particular, our method is verified on (i) the CIFAR-10 dataset (Krizhevsky & Hinton, 2009) with synthetic noisy labels generated by randomly flipping the original ones, and (ii) the WebVision dataset (Li et al., 2017), which is a large benchmark consisting of 2.4 million images crawled from websites, containing real-world noisy labels.

2. Preliminaries

For a c-class classification, we collect a dataset $\mathcal{D} = \{x_t, y_t\}_{t=1}^n$, where x_t is the t-th sample with its observed label as $y_t \in [c] := \{1, \dots, c\}$. As discussed previously, the observed label y may be corrupted since the example x are often labeled by online queries or in crowdsourcing system. Let $\hat{y}$ denote the true label, we can describe the corruption process of the set $\mathcal{D}$ by introducing a noise transition matrix $T \in \mathbb{R}^{c\times c}$, where $T_{ij} = P(y = j|\hat{y} = i)$ denotes the probability of labeling an i-th class example as j. In the cross-validation, we randomly split the collected samples $\mathcal{D}$ into two halves $\mathcal{D}_1$ and $\mathcal{D}_2$. In this way, $\mathcal{D}_2$ shares the same noise transition matrix T with $\mathcal{D}_1$. Let $f(x;\omega)$ denote a neural network parameterized by ω, and $y^f \in [c]$ denote the predicted label of x given by the network $f(x;\omega)$.

3. Understanding DNNs trained with noisy labels

Extensive experiments in (Zhang et al., 2017) have shown that DNNs can fit the noisy, even random, labels contained in the training set, but the generalization error is large even on a test set with the same noise. In this section, we use the previously introduced noise transition matrix T to theoretically quantify the generalization performance of DNNs normally trained with noisy labels, which perfectly explains the empirical findings reported in (Zhang et al., 2017).

In the classical *Probably Approximately Correct* framework (Valiant, 1984), good generalization performance means that prediction y^f and observed test label y are approximately identical as random variables, namely they should be equal for each testing sample x. Without label corruption, the generalization error can be bounded by VC dimension (Vapnik, 1998), Rademacher complexity (Bartlett & Mendelson, 2002), etc. However, in dealing with DNNs trained with noisy labels, $y^f = y$ possibly does not hold when evaluated at each testing example x, resulting in a large generalization error (Zhang et al., 2017). Fortunately, we find that the generalization still occurs in the sense of distribution, namely *generalization in distribution*, as shown in the following Claim 1. Recall that in cross-validation, we randomly di-

vide a noisy dataset $\mathcal{D}$ into two halves $\mathcal{D}_1$ and $\mathcal{D}_2$.

Claim 1. *(Generalization in distribution). Let $f(x;\omega)$ be the network trained on $\mathcal{D}_1$ and tested on $\mathcal{D}_2$. If we assume (i) the observed input examples x are i.i.d. in the set $\mathcal{D}$, (ii) f has a sufficiently high capacity, then on $\mathcal{D}_2$, the probability of predicting an truly i-th class test sample as j is*

$$P(y^f = j|\hat{y} = i) = T_{ij}, \tag{1}$$

where $T_{ij} := P(y = j|\hat{y} = i)$ denotes the noise transition matrix shared by $\mathcal{D}_1$ and $\mathcal{D}_2$.

Claim 1 reveals the fact that the prediction y^f and the test label y have the same distribution. Actually, if the model trained on $\mathcal{D}_1$ is tested on another clean test set with true labels, Eq. (1) still holds, while in this case it implies that the probability of predicting an i-th class test sample as j equals to the T_{ij} of the training set $\mathcal{D}_1$. We will justify the Claim 1 through experiments in Sec. 5.1.

The **Test Accuracy** is a widely used metric, which is defined as the proportion of testing examples for which the *prediction* y^f equals to the *observed label* y. In the following Prop. 1, we formulate the test accuracy on the test set $\mathcal{D}_2$.

Proposition 1. *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two datasets with the same noise transition matrix T, $f(x;\omega)$ be a network trained on $\mathcal{D}_1$ and tested on $\mathcal{D}_2$. Following the assumptions in Claim 1, the test accuracy for any class $i \in [c]$ is*

$$P(y^f = y|\hat{y} = i) = \sum_{j=1}^{c} T_{ij}^2. \tag{2}$$

Proof. Based on Claim 1, y^f and y have the same distribution characterized by T. Assume the label corruption process is independent, then on the test set, we have

$$\begin{aligned} &P(y^f = j, y = k|\hat{y} = i) \\ =&P(y^f = j|\hat{y} = i)P(y = k|\hat{y} = i) = T_{ij}T_{ik}. \end{aligned} \tag{3}$$

Hence, Eq. (2) follows from $P(y^f = y|\hat{y} = i) = \sum_{j=1}^{c} P(y^f = j, y = j|\hat{y} = i)$. □

3.1. Symmetric and Asymmetric Noise

Following previous literatures (Ren et al., 2018; Han et al., 2018; Jiang et al., 2018; Ma et al., 2018), in this subsection we focus on investigating two representative types of noise, symmetric and asymmetric noise, which can be defined as follows (see Fig. 1 for examples),

Definition 1. *In the case of **symmetric noise** of ratio ε, $\forall i \in [c]$, we define $T_{ii} = 1-\varepsilon$, and $T_{ij} = \varepsilon/(c-1), \forall j \neq i$. In the case of **asymmetric noise** of ratio ε, $\forall i \in [c]$, we define $T_{ii} = 1-\varepsilon$, $T_{ij} = \varepsilon$ for some $j \neq i$, and $T_{ij} = 0$ otherwise.*

Algorithm 1 Noisy Cross-Validation (NCV): selecting clean samples out of the noisy ones

INPUT: the noisy set $\mathcal{D}$, epoch E

1: $\mathcal{S} = \emptyset$, initialize a network $f(x;\omega)$
2: Randomly divide $\mathcal{D}$ into two halves $\mathcal{D}_1$ and $\mathcal{D}_2$
3: Train $f(x;\omega)$ on $\mathcal{D}_1$ for E epochs
4: Select samples, $\mathcal{S}_1 = \{(x,y) \in \mathcal{D}_2 : y^f = y\}$
5: Reinitialize the network $f(x;\omega)$
6: Train $f(x;\omega)$ on $\mathcal{D}_2$ for E epochs
7: Select samples, $\mathcal{S}_2 = \{(x,y) \in \mathcal{D}_1 : y^f = y\}$
8: $\mathcal{S} = \mathcal{S}_1 \cup \mathcal{S}_2$

OUTPUT: the selected set $\mathcal{S}$

In the cases of symmetric and asymmetric noise, we can use the noise ratio ε to quantify the test accuracy of DNNs, which are trained and tested on previously mentioned noisy datasets $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively.

Corollary 1.1. *For symmetric noise of ratio ε, the test accuracy is*

$$P(y^f = y) = (1-\varepsilon)^2 + \frac{\varepsilon^2}{c-1}. \tag{4}$$

For asymmetric noise of ratio ε, the test accuracy is

$$P(y^f = y) = (1-\varepsilon)^2 + \varepsilon^2. \tag{5}$$

Proof. Following Prop. 1, we have

$$\begin{aligned} P(y^f = y) &= \sum_{i=1}^{c} P(\hat{y} = i)P(y^f = y|\hat{y} = i) \\ &= \sum_{i=1}^{c} P(\hat{y} = i) \sum_{j=1}^{c} T_{ij}^2. \end{aligned}$$

Note that for the symmetric and asymmetric noise, $\forall i \in [c]$, $\sum_{j=1}^{c} T_{ij}^2$ is a constant given by ε. Therefore, the desired result follows by inserting ε into the equation. □

Interestingly, Eq. (4) perfectly fits the experimental results of generalization accuracy shown in Fig. 1(c) of (Zhang et al., 2017), and enables us to estimate the noise ratio of a dataset from the experimental test accuracy.

4. Training DNNs against noisy labels

In this section, we present a method on top of the Co-teaching strategy to train DNNs robustly against noisy labels. As introduced previously, the performance of the Co-teaching decays seriously and becomes unstable when the noise ratio of the training set increases, which is further

demonstrated in our experiments. To address this issue, we propose to first select a subset of samples, which has much smaller noise ratio than the original dataset.

A sample (x, y) is **clean**, if its observed label y equals to its latent true class $\hat{y}$. However, $\hat{y}$ is unavailable in practice. We propose to identify a sample (x, y) as clean if its observed label y equals to its predicted label y^f given by the network $f(x; \omega)$. If we aim to identify whether a sample (x, y) is clean or not, we should keep this sample out of the training set. An intuitive method can be found in Alg. 1, namely the Noisy Cross-Validation (NCV) method, whose validity will be justified through the following theoretical analysis and extensive experiments in the next section.

Following the standard metrics (Powers, 2011), we measure the identification performance in terms of *Label Precision* (LP) (Han et al., 2018) and *Label Recall* (LR),

$$\begin{aligned} LP &:= \frac{|\{(x,y) \in \mathcal{S} : y = \hat{y}\}|}{|\mathcal{S}|}, \\ LR &:= \frac{|\{(x,y) \in \mathcal{S} : y = \hat{y}\}|}{|\{(x,y) \in \mathcal{D} : y = \hat{y}\}|}, \end{aligned} \tag{6}$$

where $\mathcal{S} \subset \mathcal{D}$ is the selected subset as given in Alg 1, and $|\cdot|$ denotes the number of samples in a set. In this way, LP represents the fraction of clean samples in $\mathcal{S}$, and LR represents the fraction of clean samples in $\mathcal{S}$ over all clean samples in $\mathcal{D}$. Note that the noise ratio of the selected set $\mathcal{S}$ is $\varepsilon_{\mathcal{S}} = 1 - LP$ according to the above definition. We also have LP and LR for any class $i \in [c]$:

$$\begin{aligned} LP_i &:= \frac{|\{(x,y) \in \mathcal{S} : y = \hat{y} = i\}|}{|\{(x,y) \in \mathcal{S} : \hat{y} = i\}|}, \\ LR_i &:= \frac{|\{(x,y) \in \mathcal{S} : y = \hat{y} = i\}|}{|\{(x,y) \in \mathcal{D} : y = \hat{y} = i\}|}. \end{aligned} \tag{7}$$

Based on the analysis presented in Sec. 3, we quantify the performance of Alg. 1 in the following Prop. 2.

Proposition 2. *Using Alg. 1 to select clean samples, we have,* $\forall i \in [c]$

$$LP_i = \frac{T_{ii}^2}{\sum_{j=1}^{c} T_{ij}^2}, \quad LR_i = T_{ii}. \tag{8}$$

Proof. According to Alg. 1, we can reformulate Eq. (7) as

$$\begin{aligned} LP_i &= \frac{P(y^f = i, y = i | \hat{y} = i)}{P(y^f = y | \hat{y} = i)}, \\ LR_i &= \frac{P(y^f = i, y = i | \hat{y} = i)}{P(y = i | \hat{y} = i)}. \end{aligned}$$

The desired result follows by inserting Eq. (2) & (3) into the above equations. □

Algorithm 2 Iterative Noisy Cross-Validation (INCV): selecting clean samples out of the noisy ones

INPUT: the noisy set $\mathcal{D}$, number of iterations N, epoch E, remove ratio r

1: selected set $\mathcal{S} = \emptyset$, candidate set $\mathcal{C} = \mathcal{D}$
2: **for** $i = 1, \cdots, N$ **do**
3: Initialize a network $f(x; \omega)$
4: Randomly divide $\mathcal{C}$ into two halves $\mathcal{C}_1$ and $\mathcal{C}_2$
5: Train $f(x; \omega)$ on $\mathcal{S} \cup \mathcal{C}_1$ for E epochs
6: Select samples, $\mathcal{S}_1 = \{(x, y) \in \mathcal{C}_2 : y^f = y\}$
7: Identify $n = r|\mathcal{S}_1|$ samples that will be removed: $\mathcal{R}_1 = \{\#n \arg\max_{\mathcal{C}_2} \mathcal{L}(y, f(x; \omega))\}$
8: **if** $i = 1$, estimate the noise ratio ε using Eq. (4)
9: Reinitialize the network $f(x; \omega)$
10: Train $f(x; \omega)$ on $\mathcal{S} \cup \mathcal{C}_2$ for E epochs
11: Select samples, $\mathcal{S}_2 = \{(x, y) \in \mathcal{C}_1 : y^f = y\}$
12: Identify $n = r|\mathcal{S}_2|$ samples that will be removed: $\mathcal{R}_2 = \{\#n \arg\max_{\mathcal{C}_1} \mathcal{L}(y, f(x; \omega))\}$
13: $\mathcal{S} = \mathcal{S} \cup \mathcal{S}_1 \cup \mathcal{S}_2$, $\mathcal{C} = \mathcal{C} - \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{R}_1 \cup \mathcal{R}_2$
14: **end for**

OUTPUT: the selected set $\mathcal{S}$, remaining candidate set $\mathcal{C}$ and estimated noise ratio ε

4.1. Symmetric and Asymmetric Noise

Since $\forall i$, $\sum_{j=1}^{c} T_{ij} = 1$, Eq. (8) in general implies:

Corollary 2.1.

$$\frac{T_{ii}^2}{T_{ii}^2 + (1 - T_{ii})^2} \leq LP_i \leq \frac{T_{ii}^2}{T_{ii}^2 + \frac{(1-T_{ii})^2}{c-1}}. \tag{9}$$

Interestingly, we can see that the upper bound of Eq. (9) is attained for the symmetric noise, and the lower bound is attained for the asymmetric noise. In the cases of symmetric and asymmetric noise, we further have $LP = LP_1 = \cdots = LP_c$, $LR = LR_1 = \cdots = LR_c$, so that we can reformulate the LP and LR in the following Cor. 2.2.

Corollary 2.2. *For the symmetric noise of ratio ε, we have*

$$LP = \frac{(1-\varepsilon)^2}{(1-\varepsilon)^2 + \varepsilon^2/(c-1)}, \quad LR = 1 - \varepsilon. \tag{10}$$

For the asymmetric noise of ratio ε, we have

$$LP = \frac{(1-\varepsilon)^2}{(1-\varepsilon)^2 + \varepsilon^2}, \quad LR = 1 - \varepsilon. \tag{11}$$

Given the noise ratio ε of the original set $\mathcal{D}$ estimated by Eq. (4) or (5), the above Cor. 2.2 further enables us to estimate the metrics LP and LR. Recall that the noise ratio of the selected subset $\mathcal{S}$ is $\varepsilon_S = 1 - LP$ according to the definition of LP. In practical situations ($\forall i$, T_{ii} being the largest among T_{ij}, $j \in [c]$), **Alg. 1 always produces a subset with smaller noise ratio** $\varepsilon_S < \varepsilon$. See Supp. D for more details.

Algorithm 3 Training DNNs robustly against noisy labels

INPUT: the selected set $\mathcal{S}$, candidate set $\mathcal{C}$ and estimated noise ratio ε from Alg. 2, warm-up epoch E_0, total epoch E_{max}

```
Initialize two networks f_1(x;ω_1) and f_2(x;ω_2)
for e = 1, ··· , E_max do
    for batches (B_S, B_C) in (S, C) do
        if t > E_0 then B = B_S ∪ B_C, else B = B_S
        B_1 = {#n(e) arg min_B L(y, f_1(x;ω_1))}
        B_2 = {#n(e) arg min_B L(y, f_2(x;ω_2))}
        Update f_1 using B_2
        Update f_2 using B_1
    end for
end for
```

OUTPUT: $f_1(x;\omega_1)$, $f_2(x;\omega_2)$

4.2. Improving the Co-teaching with the INCV method

Although the subset selected by Alg. 1 usually has much smaller noise ratio than the original set, the robust training of DNNs may require larger number of training samples. To address this issue, we present the Iterative Noisy Cross-Validation (INCV) method to increase the number of selected samples by applying Alg. 1 iteratively. More details of the INCV can be found in Alg. 2. Apart from selecting clean samples, the INCV removes samples that have large categorical cross entropy loss at each iteration. The remove ratio r determines how many samples will be removed.

After a detailed dissection of the noisy dataset $\mathcal{D}$ by Alg. 2, we can further improve the Co-teaching to take full advantage of the selected set $\mathcal{S}$ and the candidate set $\mathcal{C}$. Specifically, we let the two networks focus on the selected set $\mathcal{S}$ at the first E_0 epochs, then incorporate the candidate set $\mathcal{C}$. Hence, both training stability and test accuracy are improved. More details of our method can be found in Alg. 3.

5. Experiments

This section consists of three parts. Firstly, we experimentally verify the theoretical results presented in Sec. 3 & 4. Then we demonstrate that the INCV method shown in Alg. 2 can identify more samples that have correct labels. Finally, we show that our proposed method outlined in Alg. 3 can train DNNs robustly against noisy labels, and outperforms state-of-the-art methods (Patrini et al., 2017; Malach & Shalev-Shwartz, 2017; Han et al., 2018; Jiang et al., 2018; Ma et al., 2018). Our code is available at `https://github.com/chenpf1025/noisy_label_understanding_utilizing`.

Experimental setup. To verify our theory and test the algorithm, we first conduct experiments on synthetic noisy labels generated by randomly corrupting the original labels in CIFAR-10 (Krizhevsky & Hinton, 2009). We focus on two representative types of noise: symmetric noise and asymmetric noise, as defined in Def. 1 and illustrated in Fig 1. To verify our method on real-world noisy labels, we use the WebVision dataset (Li et al., 2017) which contains 2.4 million images crawled from websites using the 1,000 concepts in ImageNet ILSVRC12 (Deng et al., 2009). The training set of WebVision contains many real-world noisy labels without human annotation. More implementation details are presented in Supp. A. In the following subsections, we focus on experimental results and discussions.

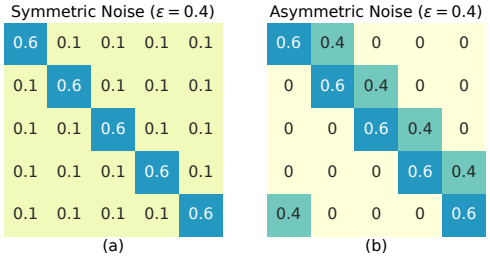

Figure 1. Examples of noise transition matrix T (taking 5 classes and noise ratio 0.4 as an example).

5.1. Behavior of DNNs trained with noisy labels

For DNNs normally trained with noisy labels, we have theoretically characterized their behavior with the following metrics (i) test accuracy given in Eq. (4) & (5), (ii) LP given in Eq. (10) & (11); (iii) LR given in Eq. (10) & (11). In this subsection, we evaluate these three metrics in extensive experiments, and show that experimental results confirm our theoretical analysis. Given a noisy dataset $\mathcal{D}$, we implement cross-validation to randomly split it into two halves $\mathcal{D}_1$, $\mathcal{D}_2$, then train the ResNet-110 (He et al., 2016b) on $\mathcal{D}_1$ and test on $\mathcal{D}_2$.

Experimental results confirm the theoretical analysis. As shown in Fig. 2, the experimental results are consistent with theoretical estimations. In particular, Fig. 2 (a) reproduces the observation shown in (Zhang et al., 2017) that the test accuracy is highly dependent of the noise ratio. (Zhang et al., 2017) did not present any theoretical explanations while we explicitly formulate in Eq. (4) that the test accuracy is a quadratic function of the noise ratio. In Fig 2 (b) and (e), the experimental LP is precisely given by our formulas. It is observed that for some data points, the experimental test accuracy and LR are slightly smaller than our theoretical values. This is reasonable since the distribution of $\mathcal{D}_2$ is not exactly the same as $\mathcal{D}_1$, and the generalization error would not become 0 even without noise.

To further investigate the prediction behavior of DNNs trained with noisy labels, we define a confusion matrix M, whose ij-th entry represents the probability of predicting an

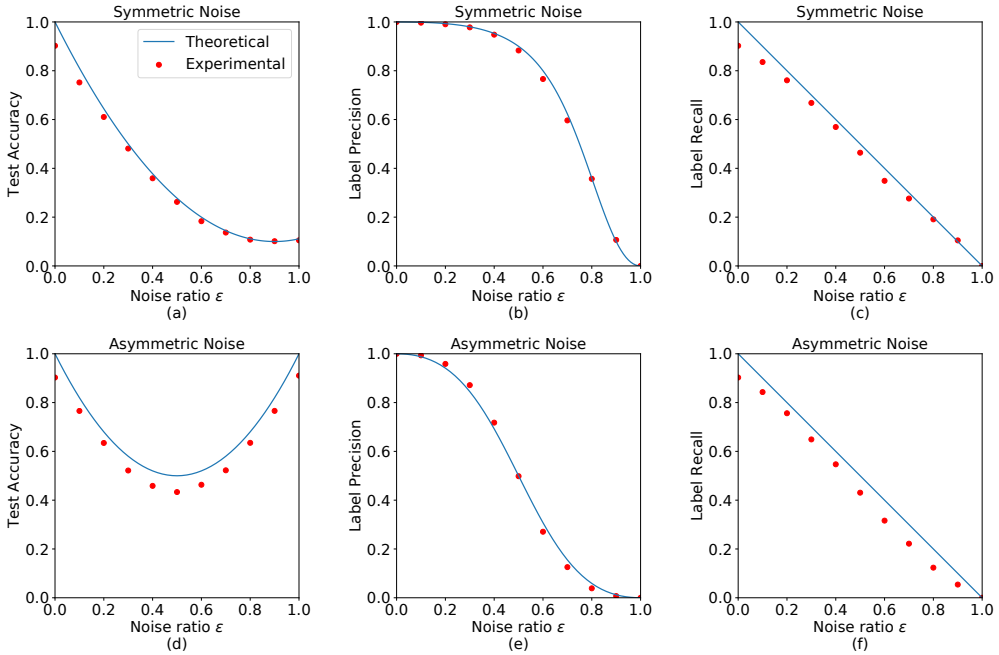

Figure 2. Test accuracy, label precision (LP) and label recall (LR) w.r.t noise ratio on manually corrupted CIFAR-10. The first row corresponds to symmetric noise and the second row asymmetric. Following cross-validation, we train the ResNet-110 on half of the noisy dataset and test on the rest half. The experimental results are consistent with the theoretical curves.

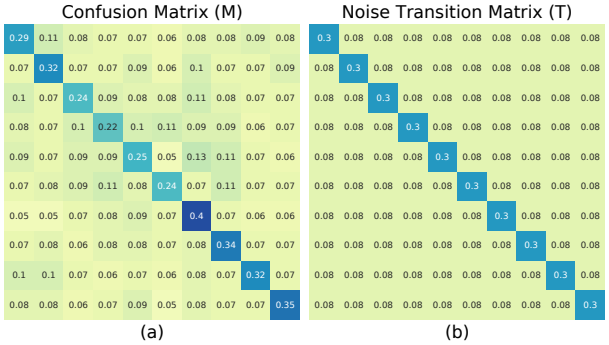

Figure 3. Confusion matrix of the RseNet-110 which is normally trained on manually corrupted CIFAR-10 with noise transition matrix T. $M \approx T$ satisfies the statement presented in Claim 1.

i-th class test sample as j, s.t.,

$$M_{ij} := P(y^f = j|\hat{y} = i).$$

Fig. 3 illustrates the confusion matrix of DNNs trained on manually corrupted CIFAR-10 with symmetric noise of ratio 0.7, and we can find that $M \approx T$, which satisfies the statement presented in Claim 1. More results can be found in Supp. B, where we show $M \approx T$ still holds.

Training accuracy converging to an extremely low value does not contradict our findings. We find that under large symmetric noise, training accuracy of the model always converges to an extremely low value. In the experiments, when trained with symmetric noise of ratio 0.7, 0.8, 0.9 and 1.0, the training accuracies are only 0.58, 0.40, 0.24 and 0.36, respectively. However, we show in Fig. 2 & 3 that our theoretical results are always consistent with the experimental ones. The phenomena further raises a fundamental question: *Is a high training accuracy a necessary condition of learning and generalization?* Without data augmentation, the theorem on finite sample expressiveness (Zhang et al., 2017) indicates that DNNs can always achieve 0 training error on the finite number of training samples. However, standard data augmentation (He et al., 2016a) is used in our implementation, which makes it difficult to achieve a high training accuracy, especially under large symmetric noise. Intuitively, due to the existence of noisy labels, nearby samples from the same class may have different labels, requiring many small regions to be classified differently. Augmentation easily generates random samples violating the classifier regions learned previously, hence increases the training error. Even in this case, our theoretical formulas presented previously still hold, as shown in Fig. 2 & 3. Here we conclude that *as long as a sufficiently rich deep neural network is trained for sufficiently many steps till convergence, the network can fit the training set and generalize in distribution, even if there are noisy labels and the training accuracy is low.* We call for more theoretical explanations on this interesting phenomena in future.

5.2. Identifying more clean samples by the INCV

Fig. 2 (b) and (e) verifies that the subset selected by Alg. 1 usually has much smaller noise ratio than the original set. Sometimes, training DNNs requires larger number of training samples. Here we demonstrate that Alg. 2 (INCV) can

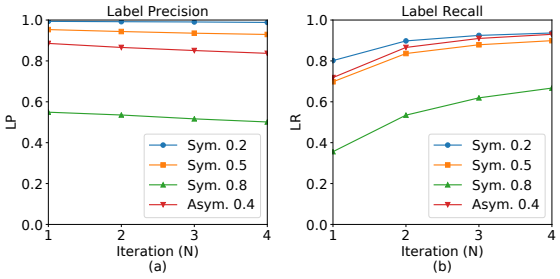

Figure 4. LP and LR of the INCV on the manually corrupted CIFAR-10. In each figure, the four curves correspond to symmetric noise of ratio 0.2, 0.5, 0.8 and asymmetric noise of ratio 0.4.

identify more clean samples through iteration. For efficiency, we use the ResNet-32 and set $N = 4$, $E = 50$ without fine tuning. ε is estimated automatically using Eq. (4) in all experiments.

The INCV identifies most clean samples accurately. Fig. 4 illustrates the average LP and LR values of the Alg. 2, computed by repeating all experiments 5 times. As show in the figure, the LP and LR are better than the theoretical lower bound even after a single iteration. Compared with ResNet-110 used in Sec. 5.1, in this subsection we train the ResNet-32 for only 50 epochs at each iteration. A much simpler model naturally releases the overfitting problem, yielding better LP and LR. Besides, Fig. 4 also demonstrates that the LR increases much with iteration, while the LP slightly decreases. After four iterations, the INCV accurately identifies most clean samples. For example, under symmetric noise of ratio 0.5, it selects about 90% ($= LR$) of the clean samples, and the noise ratio of the selected set is reduced to around 10% ($= 1 - LP$).

Noisy labels exist even in the original CIFAR-10. We also run the INCV on the original CIFAR-10 for just 1 iteration and examine samples that are identified as corrupted ones. Interestingly, there are several confusing samples, as shown in Fig. 5. This indicates that noisy labels exist even in the original CIFAR-10. Although corrupted samples contained in CIFAR-10 are so rare, which have negligible influence on training, being capable of identifying them implies that the INCV is a powerful algorithm for cleaning noisy labels.

5.3. Training DNNs robustly against noisy labels

As outlined in Alg. 3, we reformulate the Co-teaching to take full advantage of our INCV method. The followings clarify some questions that are useful for practical implementations of Alg. 3.

- Q: *How to set the size of mini-batches $\mathcal{B}_\mathcal{C}$ and $\mathcal{B}_\mathcal{S}$ drawn from $\mathcal{C}$ and $\mathcal{S}$?*
  A: In general, it is reasonable to draw mini-batches such that $|\mathcal{B}_\mathcal{C}|/|\mathcal{B}_\mathcal{S}| = |\mathcal{C}|/|\mathcal{S}|$. However, when $\mathcal{C}$ is large, it results in drawing too many samples from $\mathcal{C}$, which harms the training process since $\mathcal{C}$ usually contains many corrupted samples. Therefore, we adjust the strategy slightly by setting $|\mathcal{B}_\mathcal{C}|/|\mathcal{B}_\mathcal{S}| = \min(0.5, |\mathcal{C}|/|\mathcal{S}|)$. In the experiments, we set the batch size $|\mathcal{B}_\mathcal{S}|$ to 128, then compute $|\mathcal{B}_\mathcal{C}|$ accordingly.

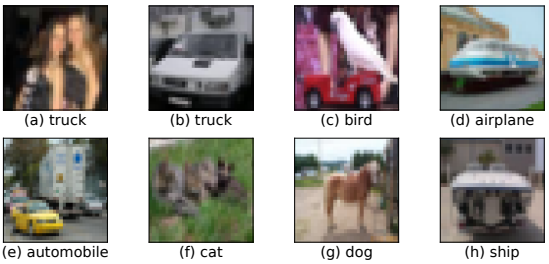

Figure 5. Noisy labels contained in the CIFAR-10 and identified by the INCV. Original Labels are annotated under images. (a) Human labeled as truck. (b) Labeled as truck, actually an automobile? (c) A bird on a toy car. (d) Labeled as airplane. (e) An automobile beside a truck. (f) Labeled as cat. (g) Labeled as dog, actually a horse? (h) Labeled as ship.

- Q: *How many samples should we keep in each mini-batch?*
  A: In each mini-batch, we update the network using $\#n(e)$ samples that have small training loss, where e is the current epoch. Following Co-teaching (Han et al., 2018), we set $n(e) = |\mathcal{B}_\mathcal{S}|(1-\varepsilon_\mathcal{S}\min(e/10, 1))$, which means we decrease $n(e)$ from $|\mathcal{B}_\mathcal{S}|$ to $|\mathcal{B}_\mathcal{S}|(1-\varepsilon_\mathcal{S})$ linearly at the first 10 epochs and fix it after that. Recall that $\varepsilon_\mathcal{S} = 1 - LP$ denotes the noise ratio of $\mathcal{S}$.

Comparable methods. We compare Alg. 3 with the following baselines (1) *F-correction* (Patrini et al., 2017). It first trains a network to estimate T, then corrects the loss function accordingly. (2) *Decoupling* (Malach & Shalev-Shwartz, 2017). It trains two networks on samples for which the predictions from the two networks are different. (3) *Co-teaching* (Han et al., 2018). It maintains two networks. Each network selects samples of small training loss from the mini-batches and feeds them to the other network. (4) *MentorNet* (Jiang et al., 2018). A teacher network is pre-trained, which provides a sample weighting scheme to train the student network. (5) *D2L* (Ma et al., 2018). For each sample, it linearly combines the original label and the prediction of network as the new label. The combining weight depends on the dimensionality of the latent feature subspace (Amsaleg et al., 2017).

Experiments on manually corrupted CIFAR-10. We first evaluate all methods on the CIFAR-10 by manually corrupting the labels with different types of noise. For symmetric noise, we test noise ratio 0.2, 0.5 and 0.8. For asymmetric noise, we choose a non-trivial and challenging noise ratio 0.4, since asymmetric noise larger than 0.5 is trivial. Still, we use the ResNet-32 and repeat all experiments five times.

Table 1. Average test accuracy (%, 5 runs) with standard deviation under different noise types and noise ratios. We train the RseNet-32 on manually corrupted CIFAR-10 and test on the clean test set. The best result is marked in bold face.

Method	Sym.			Asym.
	0.2	0.5	0.8	0.4
F-correction	85.08 ±0.43	76.02 ±0.19	34.76 ±4.53	83.55 ±2.15
Decoupling	86.72 ±0.32	79.31 ±0.62	36.90 ±4.61	75.27 ±0.83
Co-teaching	89.05 ±0.32	82.12 ±0.59	16.21 ±3.02	84.55 ±2.81
MentorNet	88.36 ±0.46	77.10 ±0.44	28.89 ±2.29	77.33 ±0.79
D2L	86.12 ±0.43	67.39 ±13.62	10.02 ±0.04	85.57 ±1.21
Ours	**89.71** ±0.18	**84.78** ±0.33	**52.27** ±3.50	**86.04** ±0.54

Table 2. Validation accuracy (%) on the WebVision validation set and ImageNet ILSVRC12 validation set. The number outside (inside) the parentheses denotes Top-1 (Top-5) classification accuracy. We train the inception-resnet v2 on the first 50 classes of the WebVision training set, which contains real-world noisy labels. The best result is marked in bold face.

Method	WebVision Val.	ILSVRC2012 Val.
F-correction	61.12 (82.68)	57.36 (82.36)
Decoupling	62.54 (84.74)	58.26 (82.26)
Co-teaching	63.58 (85.20)	61.48 (84.70)
MentorNet	63.00 (81.40)	57.80 (79.92)
D2L	62.68 (84.00)	57.80 (81.36)
Ours	**65.24 (85.34)**	**61.60 (84.98)**

As shown in Table 1, our method always achieves the best test accuracy (marked in boldface) under all cases. Even for symmetric noise of ratio 0.8 which is challenging for most methods, we achieve a good test accuracy. Fig. 6 illustrates the test accuracy of all methods on the clean test set after every training epoch. It can be found that our method impressively achieves the best test accuracy in all settings, while some baseline methods suffer from overfitting at the later stage of training, such as F-correction, Decoupling and MentorNet shown in Fig 6 (b) & (d), and D2L shown in all four sub-figures. In particular, compared with the Co-teaching (Han et al., 2018), our method further enjoys a more stable training process and obtains better test accuracy by training on a clean subset firstly.

Experiments on real-world noisy labels. To verify the practical usage of our method on real-world noisy labels, we use the WebVision dataset 1.0 (Li et al., 2017), whose training set contains many real-world noisy labels. Since the dataset is quite large, for quick experiments, we compare all

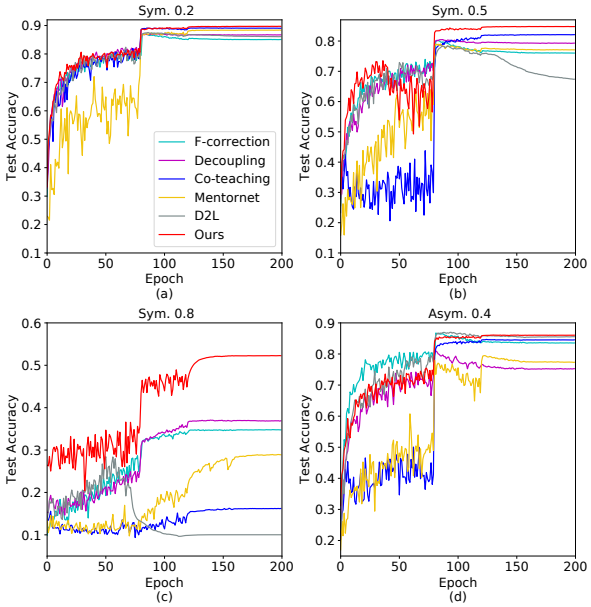

Figure 6. Average test accuracy (5 runs) during training under different noise types and noise ratios. We train the RseNet-32 on manually corrupted CIFAR-10 and test on the clean test set. The sharp change of accuracy results from the learning rate change.

methods on the first 50 classes of the Google image subset using the inception-resnet v2 (Szegedy et al., 2017). We test the trained model on the human-annotated WebVision validation set and the ILSVRC12 validation set. As shown in table 2, our method consistently outperforms other state-of-the-art ones in terms of test accuracy. Moreover, Supp. C, contains some noisy examples identified automatically from the WebVision dataset by our INCV method (Alg. 2), which implies the INCV is reliable on datasets containing real-world noisy labels.

6. Conclusion

In this work, we initiate a formal study of noisy labels. We first formulate several findings towards the generalization of DNNs trained with noisy labels. Theoretical analysis and extensive experiments are presented to justify our statements. Based on our findings, we then propose the INCV method, which randomly divides noisy datasets, then utilizes cross-validation to identify clean samples. We provide theoretical guarantees for the INCV, and then demonstrate through experiments that it is capable of identifying most clean samples accurately. Finally, we adopt the Co-teaching strategy which takes full advantage of the identified samples to train DNNs robustly against noisy labels. By comparing with extensive baselines, we show that our method achieves state-of-the-art test accuracy on the clean test set. In future, our formulations on the generalization performance of DNNs trained with noisy labels may promote more fundamental approaches of dealing with label corruption.

References

Amsaleg, L., Bailey, J., Barbe, D., Erfani, S., Houle, M. E., Nguyen, V., and Radovanović, M. The vulnerability of learning to adversarial perturbation increases with intrinsic dimensionality. *WIFS*, pp. 1–6, 2017.

Arpit, D., Jastrzębski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. A closer look at memorization in deep networks. *ICML*, 2017.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Blum, A. and Mitchell, T. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pp. 92–100. ACM, 1998.

Bousquet, O. and Elisseeff, A. Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526, 2002.

Chen, G., Zhang, S., Lin, D., Huang, H., and Heng, P. A. Learning to aggregate ordinal labels by maximizing separating width. *ICML*, 2017.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. 2009.

Divvala, S. K., Farhadi, A., and Guestrin, C. Learning everything about anything: Webly-supervised visual concept learning. *CVPR*, 2014.

Goldberger, J. and Ben-Reuven, E. Training deep neural-networks using a noise adaptation layer. *ICLR*, 2017.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: robust training deep neural networks with extremely noisy labels. *NeurIPS*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *CVPR*, 2016a.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. pp. 630–645, 2016b.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. *ICML*, 2018.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Li, W., Wang, L., Li, W., Agustsson, E., and Van Gool, L. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

Ma, X., Wang, Y., Houle, M. E., Zhou, S., Erfani, S. M., Xia, S.-T., Wijewickrema, S., and Bailey, J. Dimensionality-driven learning with noisy labels. *ICML*, 2018.

Malach, E. and Shalev-Shwartz, S. Decoupling" when to update" from" how to update". *NeurIPS*, 2017.

Mukherjee, S., Niyogi, P., Poggio, T., and Rifkin, R. Statistical learning: Stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. 2002.

Patrini, G., Rozza, A., Menon, A. K., Nock, R., and Qu, L. Making deep neural networks robust to label noise: A loss correction approach. *CVPR*, 2017.

Poggio, T., Rifkin, R., Mukherjee, S., and Niyogi, P. General conditions for predictivity in learning theory. *Nature*, 428 (6981):419, 2004.

Powers, D. M. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.

Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. *ICLR*, 2015.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. *ICML*, 2018.

Schroff, F., Criminisi, A., and Zisserman, A. Harvesting image databases from the web. *TPAMI*, 33(4):754–766, 2011.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. Inception-v4, inception-resnet and the impact of residual connections on learning. 4:12, 2017.

Tanaka, D., Ikami, D., Yamasaki, T., and Aizawa, K. Joint optimization framework for learning with noisy labels. *CVPR*, 2018.

Valiant, L. G. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Vapnik, V. N. Adaptive and learning systems for signal processing communications, and control. *Statistical learning theory*, 1998.

Yan, Y., Rosales, R., Fung, G., Subramanian, R., and Dy, J. Learning from multiple annotators with varying expertise. *Machine learning*, 95(3):291–327, 2014.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *ICLR*, 2017.