Learning to Purify Noisy Labels via Meta Soft Label Corrector

Yichen Wu[1], Jun Shu[1], Qi Xie[1], Qian Zhao[1], Deyu Meng[1,2,3]*
[1]Xi'an Jiaotong University, Shaanxi, China
[2]Pazhou Lab, Guangzhou, 510330, China
[3]Macau University of Science and Technology, Macau, China
wuyichen.am97@gmail.com, xjtushujun@gmail.com, xq.liwu@stu.xjtu.edu.cn
{timmy.zhaoqian,dymeng}@mail.xjtu.edu.cn

Abstract

Recent deep neural networks (DNNs) can easily overfit to biased training data with noisy labels. Label correction strategy is commonly used to alleviate this issue by identifying suspected noisy labels and then correcting them. Current approaches to correcting corrupted labels usually need manually pre-defined label correction rules, which makes it hard to apply in practice due to the large variations of such manual strategies with respect to different problems. To address this issue, we propose a meta-learning model, aiming at attaining an automatic scheme which can estimate soft labels through meta-gradient descent step under the guidance of a small amount of noise-free meta data. By viewing the label correction procedure as a meta-process and using a meta-learner to automatically correct labels, our method can adaptively obtain rectified soft labels gradually in iteration according to current training problems. Besides, our method is model-agnostic and can be combined with any other existing classification models with ease to make it available to noisy label cases. Comprehensive experiments substantiate the superiority of our method in both synthetic and real-world problems with noisy labels compared with current state-of-the-art label correction strategies.

Introduction

The remarkable success of deep neural networks (DNNs) on various tasks heavily relies on pre-collected large-scale dataset with high-quality annotations (He et al. 2016; Krizhevsky, Sutskever, and Hinton 2012). However, practical annotated training dataset always contains certain amount of noisy (incorrect) labels, easily conducting overfitting issue and leading to the poor performance of the trained DNNs in generalization (Zhang et al. 2016; Arpit et al. 2017). In fact, such biased training data are commonly encountered in practice, due to the coarse annotation sources for collecting them, like web searches (Liu et al. 2011) and crowd-sourcing (Welinder et al. 2010). Therefore, how to train DNNs robustly with such biased training data is a critical issue in current machine learning field.

To address this problem, various methods have been proposed (Arazo et al. 2019; Shu et al. 2019; Jiang et al. 2018; Huang et al. 2019; Yi and Wu 2019; Sukhbaatar et al. 2014; Vahdat 2017; Han et al. 2018), which can be coarsely categorized as sample selection and label correction approaches. The sample selection approach tackles this challenge mainly via adopting sample re-weighting schemes by imposing importance weights sample-wisely according to their loss values, which typically include boosting and self-paced learning methods (Kumar, Packer, and Koller 2010; Meng, Zhao, and Jiang 2017; Jiang et al. 2014). Recently, some pioneering works (Ren et al. 2018; Shu et al. 2019) further make such weighting schemes more adaptive through employing a small set of validation data to guide the network training process. All these weighting methods aim to throw off the suspected noisy samples in the training process. However, these discarded samples, even most are noisy ones, usually contain beneficial information that could improve the accuracy and robustness of the network, especially in large noise-ratio scenarios (Chang, Learned-Miller, and McCallum 2017).

The label correction approach alleviates this issue through attempting to find and correct noisy labels to their underlying true ones. Some works (Hendrycks et al. 2018; Patrini et al. 2017; Shu et al. 2020; Xia et al. 2019) tried to estimate the noise transition matrix, i.e., the probabilistic mapping from true labels to noisy ones. Then, the estimated matrix is used to correct the corrupted samples. However, matrix size increases at a geometric rate with an increasing number of classes, which makes it intractable to correct noisy labels for large scale datasets. Besides, these methods assume that the noise is class-dependent, which is not a valid assumption for more complex noises such as feature-related noise.

Some other works attempt to rectify the noisy labels by exploiting the prediction of network. E.g., (Reed et al. 2015) adopted the bootstrapping loss, which assigns a weight to the current network prediction in the learning objective, to compensate for the wrong guiding of corrupted samples. Similarly, SELFIE (Song, Kim, and Lee 2019) used the co-teaching strategy to select clean samples and progressively refurbish noisy labels that most frequently predicted by previous learned models. Another typical work, Joint Optimization (Tanaka et al. 2018) used two progressive steps to update the labels and classifier weights separately. Besides, U-correction (Arazo et al. 2019) built a two-component Beta Mixture Model (BMM) to estimate the probability of a sample being mislabeled and correct noisy labels by bootstrap-

*Corresponding author

ping loss. Similar to some semi-supervised learning methods (Tarvainen and Valpola 2017; Laine and Aila 2016; Lee 2013; Li, Socher, and Hoi 2020), this line of works can be viewed as different means of generating soft labels to replace the original targets. Albeit capable of correcting noisy labels to a certain extent, the performance of these methods heavily rely on the reliability of the generated soft labels, which depend on the accuracy of the classifier trained on the noisy dataset. However, the predictions of the base model have a huge fluctuation during training especially to the samples with corrupted labels. Some false predictions supplied by the base model will further degrade the quality of the obtained classifier.

To alleviate the above issues, in this study we design a meta soft label corrector (MSLC) to automatically and gradually purify the corrupted labels iteratively, from the perspective of meta-learning. Specifically, we treat the label correction procedure as a mutually ameliorated two-stage optimization process. One stage is to generate soft labels through MSLC by utilizing the original targets and dynamic information of predictions delivered from algorithm iterations from the base model. Then the MSLC is updated by gradient descent step in order to minimize the loss of clean meta data. The other stage is to train the base network to fit the pseudo-soft-labels generated by MSLC. Such an iteratively two-stage optimization strategy is expected to automatically obtain a faithful soft label corrector through sufficiently making use of the noise-free meta data. The contributions of this paper can be summarized as follows:

- Our method can obtain a meta soft label corrector which is able to map input labels (i.e. original target and some side information) to its corrected soft ones automatically without using conventional pre-defined generating rules, and thus makes the label correction process more flexible and easily adapting to complicated real dataset with different types and levels of noise.
- With the dynamic prediction in the iterative process to gradually ameliorate meta soft label corrector, our method tends to get more accurate classifiers through alleviating false information accumulation brought by noisy labels.
- Our approach is model agnostic and can be readily equipped on any existing classification models. Comprehensive experiments validate the superiority of the proposed method on robust deep learning with noisy labels. This can be interpreted by its obviously better noisy-clean label distinguishing capability and more accurate pseudo-labels generated by MSLC.

Typical Label Correction Methods

For a classification task, let $\mathcal{X} \subset \mathbb{R}^d$ be the feature space, $\mathcal{Y} \subset \{0,1\}^C$ be the label space. Given training data $D = \{(x_i, y_i)\}_{i=1}^N \subset (\mathcal{X} \times \mathcal{Y})^N$, where x_i denotes the i-th sample, and y_i is the corresponding one-hot encoding label vector. Denoting the network as $f(x; w)$, w represents the network parameters. Under the classical setting of supervised learning with a noisy dataset D (i.e. the label y_i can be wrongly annotated), the parameters w are learned by optimizing a chosen loss function:

$$\mathcal{L}_D(w) = \frac{1}{N} \sum_{i=1}^{N} l\left(f\left(x_i; w\right), y_i\right). \tag{1}$$

Since the given dataset D involves wrong label annotations which could misguide the training process and degrade the performance of the classifier through optimizing the objective function in Eq. (1). The existing label correction methods, therefore, mainly focus on how to generate more accurate soft pseudo-labels (represent as $\tilde{y}$) that replace the original noisy ones (i.e. y) to increase the performance of the classifier $f(x; w)$, i.e.,

$$\mathcal{L}_D(w) = \frac{1}{N} \sum_{i=1}^{N} l\left(f\left(x_i; w\right), \tilde{y}_i\right). \tag{2}$$

Typically, Reed et al. (2015) proposed a static hard bootstrapping loss to deal with label noise, which set $\tilde{y_i}^{(t)} = \lambda_i y_i + (1-\lambda_i)\hat{y}_i^{(t)}$ to replace the original label y_i in the training objective of the $(t+1)^{th}$ step:

$$\mathcal{L}_D(w) = \frac{1}{N} \sum_{i=1}^{N} l\left(f\left(x_i; w\right), \lambda_i y_i + (1-\lambda_i)\hat{y}_i^{(t)}\right), \tag{3}$$

where $\hat{y}_i^{(t)}$ denotes the predicted soft label by the classifier in the t^{th} step, λ_i is the preset parameter and $l(\cdot)$ is a chosen loss function.

In the similar formulation as Eq. (3), some other methods design its own strategy to generate pseudo-labels. For example, SELFIE (Song, Kim, and Lee 2019) set a threshold to separate the low-loss instances as clean samples and decide which samples are corrupted according to the volatility of the predictions of samples, and then correct them by the most frequently predicted label in previous q iterations.

Furthermore, Arazo et al. (2019) learned the λ_i dynamically for every sample by using a Beta-Mixture model, which is an unsupervised method to group the loss values of samples into two categories, and choose the prediction of the t^{th} step as $\hat{y}^{(t)}$ similar to Eq. (3).

Different from the form of Eq. (3), Joint Optimization (Tanaka et al. 2018) trained their model on the original targets in a large learning rate for several epochs, and then tried to use the predictions of the model to generate pseudo-labels without using the original labels. Their objective function is,

$$\mathcal{L}_D(w) = \frac{1}{N} \sum_{i=1}^{N} l\left(f\left(x_i; w\right), \frac{1}{q}\sum_{j=0}^{q-1} \hat{y}_i^{(t-j)}\right), \tag{4}$$

where the pseudo-label is valued by $\tilde{y_i}^{(t)} = \frac{1}{q}\sum_{j=0}^{q-1} \hat{y}_i^{(t-j)}$, i.e., the average of the predictions calculated from the past q epochs. With a finely set hyper-parameters q, it could achieve robust performance.

The aforementioned strategies represent the current two characteristics of existing label correction methods. The

first is that the current methods are required to specifically set the label correction rules with manually defined hyper-parameters. However, the optimal hyper-parameters vary across different problems, and thus it is difficult to construct a unique label correction methodology finely adaptable to different tasks.

The second is that these methods may cause evident error accumulation issue by substituting its generated soft pseudo-label with relatively low quality for the original labels. Bootstrap (Reed et al. 2015) and U-correction (Arazo et al. 2019) combined the observed label y with the current prediction $\hat{y}^{(t)}$ to generate new soft labels. However, the predictions $\hat{y}^{(t)}$ usually have significant variation during the training process especially to the samples are corrupted. Comparatively, Joint Optimization (Tanaka et al. 2018) alleviates this issue by integrating the predictions of the network at different iterations, its strategy, however, uses the new soft labels to replace all the observed targets no matter whether it's clean or not. This tends to introduce additional error information since some original clean labels might possibly be wrongly corrected.

The Proposed MSLC Method

To alleviate the aforementioned issues, we propose the MSLC learning framework. Different from the existing label correction methods, we view the label correction procedure as a meta-process and use a meta-learner to automatically correct labels. In this section we first introduce the MSLC framework as well as presenting an efficient algorithm for solving it, and then provide some theoretical evidences to support its underlying effectiveness insights.

Framework Formulation

Following the research line of label correction methods, we construct the label corrector with the expressoin as:

$$\tilde{y} = g\left(y, I; \theta\right), \tag{5}$$

where $\tilde{y}$ is the soft pseudo-label generated by our proposed MSLC, y denotes the original label, I represents the side information that are helpful to generate fine pseudo-label output, and θ denotes meta-learner parameters used for predicting pseudo-labels. With meta soft label corrector Eq. (5), the final training objective for $(t+1)^{th}$ step can be written as:

$$\mathcal{L}_D(\theta, w) = \frac{1}{N}\sum_{i=1}^{N} l\left(f\left(x_i; w\right), \tilde{y}_i^{(t)}\right). \tag{6}$$

Synthesize the helpful experience that we analyzed in the previous section, we choose $\hat{y}^{(t)}$ and $\tilde{y}^{(t-1)}$ as the side information for helping correct the input label y[1], which could alleviate the negative impact of the predictions' significant fluctuation during the training process. i.e.,

$$\tilde{y}^{(t)} = g\left(y, I^{(t)}; \theta\right) = g\left(y, \hat{y}^{(t)}, \tilde{y}^{(t-1)}; \theta\right), \tag{7}$$

[1]Note that more earlier generated pseudo-labels $\tilde{y}^{(t-j)}$ for $j > 1$ could be easily adopted in our method. Our experiments show that one projection $\tilde{y}^{(t-1)}$ can already guarantee a good performance.

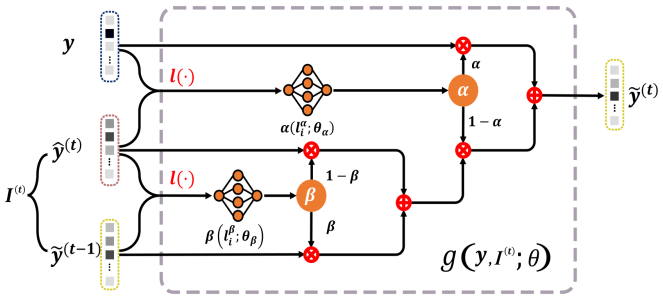


Figure 1: The Structure of MSLC

where $I^{(t)}$ denotes the side information $\hat{y}^{(t)}, \tilde{y}^{(t-1)}$ used in the current iteration step.

Inspired by (Reed et al. 2015) and (Tanaka et al. 2018), we set the corrected label as the form of soft label, which is the convex combination of $y, \hat{y}^{(t)}, \tilde{y}^{(t-1)}$. That is:

$$\begin{aligned} g\left(y, \hat{y}^{(t)}, \tilde{y}^{(t-1)}; \theta\right) = &\alpha(l^{\alpha}; \theta_{\alpha})y + (1-\alpha(l^{\alpha}; \theta_{\alpha})) \\ &\times \left(\beta(l^{\beta}; \theta_{\beta})\tilde{y}^{(t-1)} + (1-\beta(l^{\beta}; \theta_{\beta}))\hat{y}^{(t)}\right), \end{aligned} \tag{8}$$

where $\alpha(\cdot)$ and $\beta(\cdot)$ are two networks, whose outputs represent coefficients of this convex combination, with their parameters denoted as θ_{α} and θ_{β}, respectively, and thus $\theta = [\theta_{\alpha}, \theta_{\beta}]$. The two coefficient networks, with $l^{\alpha} = l(\hat{y}^{(t)}, y)$ and $l^{\beta} = l(\hat{y}^{(t)}, \tilde{y}^{(t-1)})$, constitute the main parts of our proposed meta soft label corrector, which is intuitively shown in Fig. 1. Through the two networks, the input target information, i.e. $y, \hat{y}^{(t)}, \tilde{y}^{(t-1)}$, could be combined in a convex combination to form a new soft target $\tilde{y}^{(t)}$, which will replace the original label y in the training process. The symbol α and β denote the output value of $\alpha(\cdot)$ and $\beta(\cdot)$ respectively.

In the framework of MSLC, α reflects the confidence of the original label in the given corrupted dataset. Larger α means that the corresponding sample tends to more reserve the original label, conversely, smaller α indicates the MSLC will more integrate the predictions of the classifier to replace the initial target to compensate its erroneous guidance. Similarly, the β could adaptively determine the proportion between the current and earlier prediction sample-wisely.

Training with Meta Dataset

We then introduce how to learn meta-learner parameter θ (Eq. (8)). We readily employ a meta-data driven learning regime as used in (Shu et al. 2019) under the guidance of a small amount of noise-free meta data. The meta dataset contains the meta-knowledge of underlying label distribution of clean samples, and it is thus rationally to be exploited as a sound guidance to help estimate θ. In this work, we denoted meta dataset as

$$\mathcal{D}_m = \{(x_i^{meta}, y_i^{meta})\}_{i=1}^{M}, \tag{9}$$

where $M (M \ll N)$ is the number of data samples in meta dataset. By utilizing the meta dataset, we can then design the entire training framework for the noise label correction model (Eq. (6)).

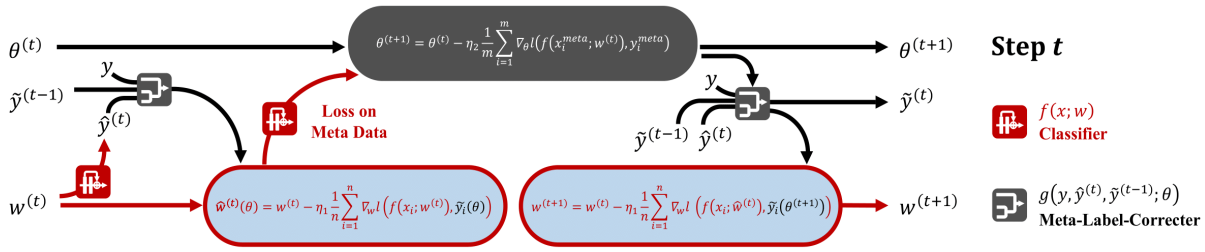


Figure 2: Main flowchart of the proposed MSLC

Algorithm 1 The Learning Algorithm of Meta Soft Label Corrector(MSLC)

Input: Training data D, meta data D_m, batch size n, m, MaxEpoch T.

Output: Classifier network parameter $w^{(T-1)}$

1: Initialize classifier parameter $w^{(0)}$ and meta-learner parameter $\theta^{(0)}$.
2: **for** $t = 0$ **to** $T-1$ **do**
3: $\{x, y\} \leftarrow$ SampleMiniBatch(D, n).
4: $\{x^{(meta)}, y^{(meta)}\} \leftarrow$ SampleMiniBatch(D_m, m).
5: Update $\theta^{(t+1)}$ by Eq. (12).
6: Update $w^{(t+1)}$ by Eq. (13).
7: Update $\hat{y}_i^{(t+1)}, \tilde{y}_i^{(t+1)}$ with parameters $w^{(t+1)}$ and $\theta^{(t+1)}$.
8: **end for**

Specifically, we formulate the following bi-level minimization problem:

$$\begin{aligned} w^*(\theta) &= \arg\min_w \mathcal{L}_D(w;\theta) \\ \theta^* &= \arg\min_\theta \mathcal{L}_{D_m}(w^*(\theta)), \end{aligned} \tag{10}$$

where $\mathcal{L}_{D_m}(w) = \frac{1}{M}\sum_{i=1}^{M} l\left(f\left(x_i^{meta};w\right), y_i^{meta}\right)$ is the meta loss on meta dataset. After achieving θ^*, we can then get the soft label corrector, which incline to ameliorate noisy labels to be correct ones, and further improve the quality of the trained classifier.

Optimizing the classifier parameters w and meta-learner parameters θ requires two nested loop of optimization (Eq. (10)), which tends to be computationally inefficient (Franceschi et al. 2018). We thus exploit SGD technique to speedup the algorithm by approximately solving the problem in a mini-batch updating manner (Shu et al. 2019; Finn, Abbeel, and Levine 2017) to jointly ameliorating θ and w. The algorithm flowchart is shown in Fig. 2.

The algorithm includes mainly the following steps. Firstly, denote the mini-batch training samples as $\{(x_i, y_i)\}_{i=1}^n$, and then the training loss becomes $\frac{1}{n}\sum_{i=1}^{n} l\left(f\left(x_i;w\right), g\left(y_i, \hat{y}_i^{(t)}, \tilde{y}_i^{(t-1)};\theta\right)\right)$. We can then deduce the formulate of one-step w updating equation with respect to θ as

$$\begin{aligned} \hat{w}(\theta) &= w^{(t)} - \eta_1 \nabla_w \mathcal{L}_D(\theta, w)) \\ &= w^{(t)} - \eta_1 \frac{1}{n}\sum_{i=1}^{n} \nabla_w l\left(f(x_i;w), g\left(y_i, \hat{y}_i^{(t)}, \tilde{y}_i^{(t-1)};\theta\right)\right)\Big|_{w^{(t)}}, \end{aligned} \tag{11}$$

where η_1 is the learning rate. Then, with current mini-batch meta data samples $\{(x_i^{meta}, y_i^{meta})\}_{i=1}^m$, we can perform one step updating for solving $\min_\theta \frac{1}{m}\sum_{i=1}^{m} l\left(f\left(x_i^{meta};w\right), y_i^{meta}\right)$, that is

$$\begin{aligned} \theta^{(t+1)} &= \theta^{(t)} - \eta_2 \nabla_\theta \mathcal{L}_{D_m}(\hat{w}(\theta)) \\ &= \theta^{(t)} - \eta_2 \frac{1}{m}\sum_{i=1}^{m} \nabla_\theta l\left(f(x_i^{meta};\hat{w}(\theta)), y_i^{meta}\right)\Big|_{w^{(t)}}. \end{aligned} \tag{12}$$

After achieving $\theta^{(t+1)}$, we can calculate the pseudo label by Eq. (5) and update w, that is

$$\begin{aligned} w^{(t+1)} &= w^{(t)} - \eta_1 \frac{1}{n} \times \\ &\sum_{i=1}^{n} \nabla_w l\left(f(x_i;w), g\left(y_i, \hat{y}_i^{(t)}, \tilde{y}_i^{(t-1)};\theta^{(t+1)}\right)\right)\Big|_{w^{(t)}}. \end{aligned} \tag{13}$$

The predicted pseudo-labels $\tilde{y}_i^{(t+1)}$ can then be updated with parameters $w^{(t+1)}$ and $\theta^{(t+1)}$. The entire algorithm is then summarized in Algorithm 1.

Theoretical Analysis On the Algorithm

Weighting Scheme: The computation of Eq. (12) can be rewritten as:

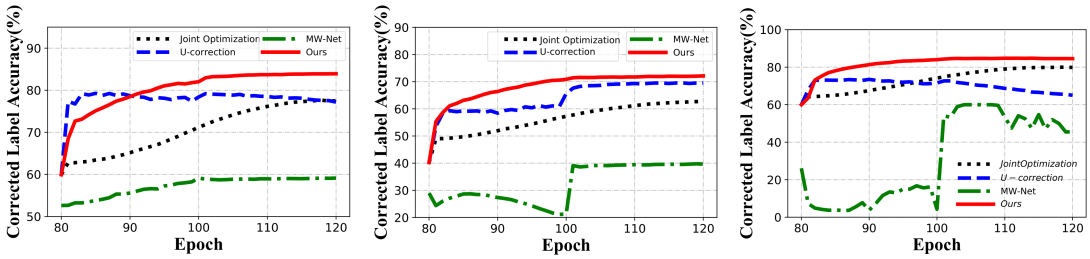
$$\begin{aligned} \theta^{(t+1)} &= \theta^{(t)} - \eta_2 \nabla_\theta \mathcal{L}_{D_m}(w^{(t)} - \eta_1 \nabla_w \mathcal{L}_D(\theta, w)) \\ &= \theta^{(t)} + \eta_2 \eta_1 \nabla^2_{\theta, w^{(t)}} \mathcal{L}_D(\theta, w^{(t)}) \nabla_{\hat{w}} \mathcal{L}_{D_m}(\hat{w}) \\ &= \theta^{(t)} + \eta_2 \eta_1 \nabla_\theta (\nabla^T_{w^{(t)}} \mathcal{L}_D(\theta, w^{(t)}) \nabla_{\hat{w}} \mathcal{L}_{D_m}(\hat{w})). \end{aligned} \tag{14}$$

It can be seen that $\nabla^T_{w^{(t)}} \mathcal{L}_D(\theta, w^{(t)}) \nabla_{\hat{w}} \mathcal{L}_{D_m}(\hat{w})$ represents the similarity between the gradient of the training sample computed on training loss and the average gradient of the mini-batch meta data calculated on meta loss. It means that if a pair of training and meta samples are very similar, then this training sample is considered as helpful for getting right results and should be up-weighted. Conversely, this training sample is harmful and should be suppressed. This understanding is consistent with (Finn, Abbeel, and Levine 2017; Ren et al. 2018; Liu, Simonyan, and Yang 2018).

Generalization Bound: Following on work by (Zhao et al. 2019), to better comprehend the effect of the parameter dimension and the size of meta data to θ, we provide a theoretical analysis of an instance of MSLC where θ is searched over a fixed candidate set $\mathcal{A}$ that covers the unit ball.

Noise-type		Symmetric Noise								Asymmetric Noise			
Dataset		CIFAR-10				CIFAR-100				CIFAR-10		CIFAR-100	
Method \ Noise ratio γ		0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8	0.2	0.4	0.2	0.4
Cross-Entropy	Best	90.22	87.33	83.2	54.79	68.03	61.18	46.43	17.91	92.85	90.22	69.05	65.14
	Last	86.33	79.61	72.99	54.26	63.67	46.92	30.96	8.29	91.29	87.23	63.68	50.10
Fine-tuning	Best	91.17	87.34	83.75	56.28	67.81	62.55	50.82	19.05	93.11	91.04	69.55	65.75
	Last	88.27	82.16	79.36	54.82	63.97	51.14	38.22	18.86	92.35	89.49	66.43	55.08
GCE(Zhang and Sabuncu 2018)	Best	90.27	88.50	83.70	57.27	71.36	63.39	58.06	16.51	90.11	85.24	69.56	57.50
	Last	90.15	88.01	82.87	57.22	71.02	52.15	45.31	15.71	89.33	82.04	66.36	56.81
GLC(Hendrycks et al. 2018)	Best	91.43	88.52	84.08	64.21	69.30	63.24	56.12	18.59	92.46	91.74	71.40	67.73
	Last	90.13	87.04	82.63	62.19	66.62	59.03	51.96	8.08	92.41	91.02	70.01	66.68
MW-Net(Shu et al. 2019)	Best	91.48	87.34	81.98	65.88	69.79	65.44	55.42	19.62	93.44	91.64	67.54	60.24
	Last	90.11	86.42	81.62	64.78	68.37	64.81	55.04	19.20	91.95	90.88	66.71	59.53
Bootstrap(Reed et al. 2015)	Best	91.46	88.75	84.03	63.80	69.79	63.73	57.20	17.63	93.08	91.18	70.93	67.82
	Last	88.00	83.57	78.69	63.41	63.00	47.08	35.86	17.04	91.02	85.59	63.46	49.18
Joint Optimization(Tanaka et al. 2018)	Best	90.85	90.27	86.49	66.39	63.84	59.82	49.13	18.95	93.39	91.43	66.90	64.82
	Last	89.77	88.58	85.57	65.92	60.10	56.85	47.68	17.38	92.12	90.20	66.69	59.31
U-correction(Arazo et al. 2019)	Best	92.05	89.07	85.64	68.23	68.37	62.37	55.19	17.10	91.85	90.34	67.71	66.75
	Last	90.21	85.45	83.15	64.78	67.42	55.40	55.04	9.33	90.92	84.31	63.82	60.64
Ours	Best	**93.46**	**91.42**	**87.39**	**69.87**	**72.51**	**68.98**	**60.81**	**24.32**	**94.39**	**92.81**	**72.66**	**70.51**
	Last	**93.38**	**91.21**	**87.25**	**68.88**	**72.02**	**68.70**	**60.25**	**20.53**	**94.11**	**92.48**	**70.20**	**69.24**

Table 1: Test accuracy (%) of all competing methods on CIFAR-10 and CIFAR-100 under Symmetric noise and Asymmetric noise with different noise levels. The best results are highlighted in bold.



(a) Cifar100-Sym-Noise40% (b) Cifar100-Sym-Noise60% (c) Cifar100-Asy-Noise40%

Figure 3: The corrected label accuracy on different noise types and noise ratios. (a)(b)shows the accuracy of 40% and 60% symmetric noise on Cifar100 respectively, (c) shows the accuracy of 40% asymmetric noise on Cifar100.

Theorem 1 *Let $\theta \in \mathbb{B}^d$ be the parameter of MSLC in a d-dimensional unit ball. Let $\mathcal{F}$ be the underlying ground truth distribution which doesn't have noisy labels. Let m be the meta data size. Define the generalization risk as:*

$$R(w) = \mathbb{E}_{(X,Y)\sim\mathcal{F}}[l(f(X;w),Y)]$$
$$\hat{R}(w) = \frac{1}{m}\sum_{i=1}^{m}[l(f(x_i;w),y_i)]$$

Let $\mathcal{A}$ be an ϵ-cover of $\mathbb{B}^d$ (i.e. $\forall\theta\in\mathbb{B}^d, \exists\theta^{'} \in \mathcal{A} : ||\theta - \theta^{'}|| \leq \epsilon$,). Let $\theta^* = argmax_{\theta\in\mathbb{B}^d} R(w^*(\theta))$ be the optimal parameter in the unit ball, and $\hat{\theta} = argmax_{\theta\in\mathcal{A}}\hat{R}(w^*(\theta))$ be the empirically optima among a candidate set $\mathcal{A}$. Assume:

- *The loss function is sub-Gaussian with parameter σ.*
- *The loss function is λ-Lipschitz continuous w.r.t θ*

For $m > 9\sigma^2$, with probability at least $1-\delta$ we have,

$$R(w^*(\theta^*)) \leq \hat{R}(w^*(\hat{\theta})) + \frac{3\sigma\lambda}{\sqrt{m}} + \sqrt{\frac{dln(m)}{m} + 2\frac{1}{m}ln(2/\delta)}.$$

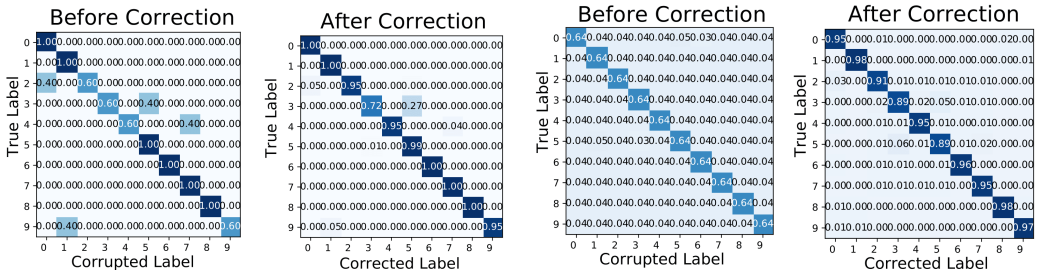
Theorem 1 establishes that our method approaches the optimal weight at a rate $O(\sqrt{dln(m)/m})$. Moreover, it shows that the impact of meta data size (i.e. m) and parameter dimension (i.e. d) to the generalization error when we use $\hat{R}(w^*(\hat{\theta}))$ to estimate $R(w^*(\theta^*))$.

Experimental Results

To evaluate the capability of the proposed method, we implement experiments on CIFAR-10, CIFAR-100 (Krizhevsky, Hinton et al. 2009) under different types and levels of noise, as well as a real-word large-scale noisy dataset Clothing1M (Xiao et al. 2015). For CIFAR-10/100, we use two types of label noise: symmetric and asymmetric. **Symmetric:**

#	Method	Accuracy	#	Method	Accuracy
1	Cross Entropy	68.94	4	Joint Optimization(Tanaka et al. 2018)	72.23
2	Bootstrapping(Reed et al. 2015)	69.12	5	MW-Net(Shu et al. 2019)	73.72
3	U-correction(Arazo et al. 2019)	71.00	6	Ours	**74.02**

Table 2: Test accuracy (%) of different models on real-world noisy dataset Clothing1M.



(a) Asym-Noise40% (b) Sym-Noise40%

Figure 4: The comparison of confusion matrices between before and after correction on CIFAR-10 with (a) asymmetric noise 40% and (b) symmetric noise 40%.

We follow (Zhang et al. 2016; Tanaka et al. 2018) for label noise addition, which generates label corruptions by flipping labels of a given proportion of training samples to one of the other class labels uniformly (the true label could be randomly maintained). **Asymmetric:** We use the setting in (Yao et al. 2019), which designs to mimic the structure of real-world label noise. Concretely, we set a probability r to disturb the label to its similar class, e.g., truck $\rightarrow$ automobile, bird $\rightarrow$ airplane, deer $\rightarrow$ horse, cat $\rightarrow$ dog. For CIFAR-100, a similar r is set but the label flip only happens in each super-class as described in (Hendrycks et al. 2018).

Baselines. The compared methods include: **Fine-tuning**, which finetunes the result of Cross-Entropy on the meta-data to further enhance its performance. **GCE** (Zhang and Sabuncu 2018), which employs a robust loss combining the benefits of both CE loss and mean absolute error loss against label noise. **GLC** (Hendrycks et al. 2018), which estimates the noise transition matrix by using a small clean label dataset. **MW-Net** (Shu et al. 2019), which uses a MLP net to learn the weighting function. **Bootstrap** (Reed et al. 2015), which deals with label noise by adding a perceptual term to the standard CE loss. **Joint Optimization** (Tanaka et al. 2018), which updates the label and model at the same time by using the pseudo-labels it generated. **U-correction** (Arazo et al. 2019), which models sample loss with BMM and applied MixUp. For fair comparison, we only compare its proposed method without mixup augmentation.

Experiment Details. We use ResNet-34 as classifier network for all baseline experiments in Table 1. We use two multi-layer perception (MLP) with one hidden layer (100 nodes) as the network structure of $\alpha(\cdot)$ and $\beta(\cdot)$ respectively. We chose cross-entropy as loss function and we began to correct labels at 80th epoch (i.e. there is an initial warm-up). Followed by (Shu et al. 2019), in synthetic datasets(noisy CIFAR-10/100), we randomly selected 1000 images with clean labels from the training dataset as the meta-data set. In the Clothing1M dataset, we use 7000 clean data as the metadata. More detail settings on synthetic dataset and real-world dataset are shown in Appendix.

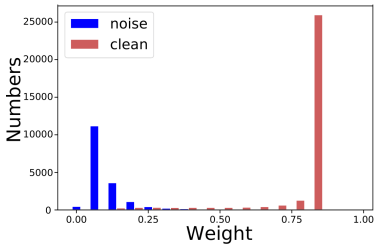
Comparison with State-of-the-Art Methods

Table 1 shows the results of all competing methods on CIFAR-10/100 under symmetric and asymmetric noise as aforementioned. To compare different methods in more detail, we report both the best and the averaged test accuracy over the last 5 epochs. It can be observed that our method gets the best performance across both datasets and all noise rates. Specifically, even under relatively high noise ratios (E.g. $\gamma = 0.8$ on CIFAR-10 with sym-noise), our algorithm has competitive classification accuracy (69.87%). It worths noted that U-correction achieved best accuracy of 68.23% that is comparable with, while its accuracy decreases in the later training as 64.78% probably due to its error accumulation issue. This indicates that our proposed meta soft label corrector has better convergence under the guidance of meta data in the training process. It also can be seen that MW-Net has relatively poor performance in asymmetric condition, which might because all classes share one weighting function in the method, which is unreasonable when noise is asymmetric. Comparatively, our proposed MSLC has a higher degree of freedom and thus performs better.

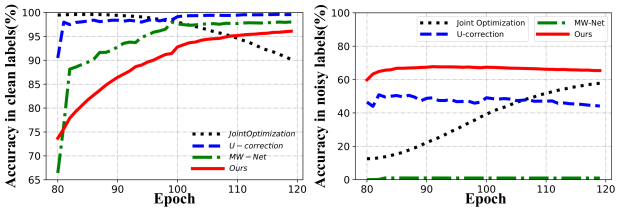
Fig.3 plots the corrected label accuracy, which used the hard form of pseudo-labels (Eq. (5)) compared with the ground truth. As can be seen in Fig. 3, the corrected labels generated by our method attain the best accuracy. The accuracy of MW-Net is always below the value of the proportion of clean samples, since it intrinsically tries to select the clean samples while ignores the corrupted ones by its weighting mechanism. From Fig. 3 (a)(c), one can see that the corrected label accuracy of the U-correction are slightly decrease, it

Dataset		CIFAR-10						CIFAR-100					
β		0	0.2	0.4	0.6	0.8	Ours	0	0.2	0.4	0.6	0.8	Ours
Accuracy	Best	89.84	90.49	91.04	90.34	89.46	**91.27**	67.42	68.52	68.25	67.13	67.08	**68.84**
	Last	89.46	90.19	90.91	89.64	89.20	**91.11**	66.93	68.06	67.83	66.61	66.24	**68.35**
Corrected Label Accuracy		92.23	93.36	94.24	93.44	91.94	**94.52**	81.47	83.29	83.04	81.28	81.24	**83.98**

Table 3: The test accuracy(%) of ablation study under 40% of sym-noise. Mean accuracy over 3 repetitions are reported.



(a) The output weight of $\alpha(\cdot)$



(b) Accuracy in clean/noisy samples corresponding to Fig.3 (a)

Figure 5: The Analysis of the proposed method on CIFAR-100 with Symmetric 40% noise. (a) denotes the output weight of $\alpha(\cdot)$ on clean/noise samples, (b) shows the corrected label accuracy on clean/noisy data which split by the whole dataset according to the ground-truth.

might be caused by its false correction[2]. Moreover, although the accuracy of Joint Optimization increases all the time, its performance is limited by the strategy that only uses the pseudo-labels to replace all the targets, which has the risk of corrupting the original clean labels.[2].

Table 2 depicts the results on real noisy dataset Clothing1M, which consists of 1 million clothing images belonging to 14 classes. These images are obtained from online websites with clean labels for validation(14K) and testing(10K). Since the labels are generated by using surrounding texts of the images provided by the sellers, they thus contain many error labels. From Table 2, we can observe that the proposed method achieves the best performance, which shows the effectiveness of our MSLC in real scenarios.

More Property Evaluations

Fig.4 shows the confusion matrices of our method obtained under symmetric and asymmetric noise on CIFAR-10. The left column of Fig. 4 (a) and (b) is the noise transition matrix, which is the guideline for generating the synthesized noisy datasets. And the right column is the matrix after corrected by our proposed method, which x-axis denotes the hard form corrected labels. By comparing the left and right columns of Fig. 4 (a) and (b), we can see that the probability of most diagonal terms exceeds 0.95 after correction. That indicates the high correction accuracy of our proposed MSLC.

Fig.5 demonstrates the output weights of $\alpha(\cdot)$ and the corrected labels accuracy on clean and noisy samples. From Fig.5 (a), we can see that the weights of clean and noisy samples are evidently different, implying that our proposed MSLC inclines to preserve the original clean labels and use other target information when the original labels are noisy. Fig.5 (b) explains that our method can finely correct the noise samples while retain the original clean samples. It is worth noting that U-correction retains more than 99% of clean samples. However, through experiments, it can be seen that it inclines to treat many noisy samples as clean ones during training, which limits its ability to correct noisy samples. As for JointOptimization, one can see that its training process corrupted the original clean labels, since it used prediction targets to replace all original labels without considering whether they are clean or not.

For further analysis the effectiveness of the network $\beta(\cdot)$, we compared its learned meta-learner parameters (β) with a set of different manually set values on CIFAR-10 and CIFAR-100. It can be observed from Table 3 that the performance is worst when the β is set to 0, which means that directly choosing the predictions of current model could not accurately correct the original labels. Furthermore, we can find that the best manually set β changes when the dataset is different. Specifically, for CIFAR-10, the best test accuracy is 91.04% corresponding to $\beta = 0.4$ case, while for CIFAR-100, the best is 68.52% corresponding to $\beta = 0.2$. Compared with setting the β value manually, our algorithm can learn it more flexibly and achieving good performance in both test accuracy and the corrected label accuracy.

Conclusion

In this paper, we provide a solution to an important problem in weakly supervised learning. Combining with the meta-learning method, we proposed a label correction method that can adaptively ameliorate corrupted labels for robust deep learning with noisy labels. Compared with current methods that use a pre-fixed generation mechanism, our method is able to do this task in a flexible automatic data-driven manner. Moreover, our methods can generate more accurate pseudo-labels to compensate the misguiding of the erroneous samples. The proposal is flexible to be used in various deep classification algorithms and different noisy datasets. Experimental results show consistent superiority of our method in datasets with different types and levels of noise.

[2]This will be further analyzed in Fig. 5

Acknowledgements

This research was supported by the National Key R&D Program of China (2020YFA0713900) and the China NSFC projects under contracts 11690011,61721002, U1811461, 62076196.

References

Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N. E.; and McGuinness, K. 2019. Unsupervised label noise modeling and loss correction. *arXiv preprint arXiv:1904.11238* .

Arpit, D.; Jastrzebski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M. S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; et al. 2017. A Closer Look at Memorization in Deep Networks. In *ICML*.

Chang, H.-S.; Learned-Miller, E.; and McCallum, A. 2017. Active bias: Training more accurate neural networks by emphasizing high variance samples. In *Advances in Neural Information Processing Systems*, 1002–1012.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1126–1135. JMLR. org.

Franceschi, L.; Frasconi, P.; Salzo, S.; Grazzi, R.; and Pontil, M. 2018. Bilevel programming for hyperparameter optimization and meta-learning. *arXiv preprint arXiv:1806.04910* .

Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.; and Sugiyama, M. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, 8527–8537.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hendrycks, D.; Mazeika, M.; Wilson, D.; and Gimpel, K. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in neural information processing systems*, 10456–10465.

Huang, J.; Qu, L.; Jia, R.; and Zhao, B. 2019. O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, 3326–3334.

Jiang, L.; Meng, D.; Mitamura, T.; and Hauptmann, A. G. 2014. Easy samples first: Self-paced reranking for zero-example multimedia search. In *Proceedings of the 22nd ACM international conference on Multimedia*, 547–556.

Jiang, L.; Zhou, Z.; Leung, T.; Li, L.-J.; and Fei-Fei, L. 2018. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. In *International Conference on Machine Learning*, 2304–2313.

Krizhevsky, A.; Hinton, G.; et al. 2009. *Learning multiple layers of features from tiny images*. Master's thesis, Department of Computer Science, University of Toronto.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NeurIPS*.

Kumar, M. P.; Packer, B.; and Koller, D. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, 1189–1197.

Laine, S.; and Aila, T. 2016. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242* .

Lee, D.-H. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 2.

Li, J.; Socher, R.; and Hoi, S. C. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394* .

Liu, H.; Simonyan, K.; and Yang, Y. 2018. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055* .

Liu, W.; Jiang, Y.-G.; Luo, J.; and Chang, S.-F. 2011. Noise resistant graph ranking for improved web image search. In *CVPR 2011*, 849–856. IEEE.

Meng, D.; Zhao, Q.; and Jiang, L. 2017. A theoretical understanding of self-paced learning. *Information Sciences* 414: 319–328.

Patrini, G.; Rozza, A.; Krishna Menon, A.; Nock, R.; and Qu, L. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1944–1952.

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

Ren, M.; Zeng, W.; Yang, B.; and Urtasun, R. 2018. Learning to Reweight Examples for Robust Deep Learning. In *International Conference on Machine Learning*, 4334–4343.

Shu, J.; Xie, Q.; Yi, L.; Zhao, Q.; Zhou, S.; Xu, Z.; and Meng, D. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, 1917–1928.

Shu, J.; Zhao, Q.; Xu, Z.; and Meng, D. 2020. Meta Transition Adaptation for Robust Deep Learning with Noisy Labels. *arXiv preprint arXiv:2006.05697* .

Song, H.; Kim, M.; and Lee, J.-G. 2019. SELFIE: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning*, 5907–5915.

Sukhbaatar, S.; Bruna, J.; Paluri, M.; Bourdev, L.; and Fergus, R. 2014. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080* .

Tanaka, D.; Ikami, D.; Yamasaki, T.; and Aizawa, K. 2018. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5552–5560.

Tarvainen, A.; and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, 1195–1204.

Vahdat, A. 2017. Toward robustness against label noise in training deep discriminative neural networks. In *Advances in Neural Information Processing Systems*, 5596–5605.

Welinder, P.; Branson, S.; Perona, P.; and Belongie, S. J. 2010. The multidimensional wisdom of crowds. In *Advances in neural information processing systems*, 2424–2432.

Xia, X.; Liu, T.; Wang, N.; Han, B.; Gong, C.; Niu, G.; and Sugiyama, M. 2019. Are Anchor Points Really Indispensable in Label-Noise Learning? In *Advances in Neural Information Processing Systems*, 6835–6846.

Xiao, T.; Xia, T.; Yang, Y.; Huang, C.; and Wang, X. 2015. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2691–2699.

Yao, J.; Wu, H.; Zhang, Y.; Tsang, I. W.; and Sun, J. 2019. Safeguarded dynamic label regression for noisy supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 9103–9110.

Yi, K.; and Wu, J. 2019. Probabilistic end-to-end noise correction for learning with noisy labels. In *CVPR*.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530* .

Zhang, Z.; and Sabuncu, M. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*.

Zhao, S.; Fard, M. M.; Narasimhan, H.; and Gupta, M. 2019. Metric-optimized example weights. In *International Conference on Machine Learning*, 7533–7542.