

Meta Label Correction for Noisy Label Learning

Guoqing Zheng, Ahmed Hassan Awadallah, Susan Dumais

Microsoft Research

{zheng, hassanam, sdumais}@microsoft.com

Abstract

Leveraging weak or noisy supervision for building effective machine learning models has long been an important research problem. Its importance has further increased recently due to the growing need for large-scale datasets to train deep learning models. Weak or noisy supervision could originate from multiple sources including non-expert annotators or automatic labeling based on heuristics or user interaction signals. There is an extensive amount of previous work focusing on leveraging noisy labels. Most notably, recent work has shown impressive gains by using a meta-learned instance re-weighting approach where a meta-learning framework is used to assign instance weights to noisy labels. In this paper, we extend this approach via posing the problem as a label correction problem within a meta-learning framework. We view the label correction procedure as a meta-process and propose a new meta-learning based framework termed MLC (Meta Label Correction) for learning with noisy labels. Specifically, a label correction network is adopted as a meta-model to produce corrected labels for noisy labels while the main model is trained to leverage the corrected labels. Both models are jointly trained by solving a bi-level optimization problem. We run extensive experiments with different label noise levels and types on both image recognition and text classification tasks. We compare the re-weighting and correction approaches showing that the correction framing addresses some of the limitation of re-weighting. We also show that the proposed MLC approach outperforms previous methods in both image and language tasks.

Introduction

Recent advances in deep learning have enabled impressive performance on various tasks, including image recognition (He et al. 2016) and natural language processing (Devlin et al. 2018). At the core of this success lies the availability of large amounts of annotated data. However, such datasets are not readily available at scale for many tasks. Learning with weak supervision aims to address this challenge by leveraging weak evidences of supervision. Weak supervision can come in several forms including: incomplete supervision; where only a small subset of the training data has labels, inexact supervision; where only coarse-grained annotations are available, and inaccurate supervision; where noisy labels

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

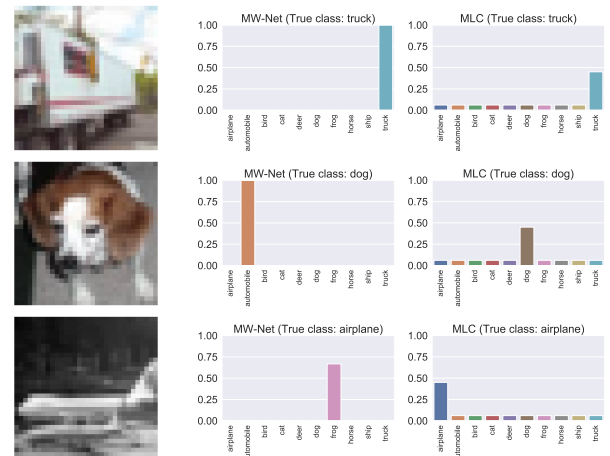


Figure 1: Illustration of label reweighting vs label correction. The first image is a truck and the given noisy label is truck, where MW-Net (A label reweighting method) and MLC successfully recover it. However, for the second image whose true label is dog and the given noisy label is automobile, MW-Net is unable to down-weight the incorrect label while MLC correctly adjusts its label by predicting the maximum weight on the class dog than all other classes. So is the case for the airplane with noisy label frog.

are given (Zhou 2017). In this work, we focus on using inaccurate (noisy) labels as a form of weak supervision. Noisy labels may originate from multiple sources including: corrupted labels, non-expert annotators, automatic labels based on heuristics or user interaction signals, etc.

Training deep networks with noisy labels is challenging since they are prone to fitting and memorizing the noise (Zhang et al. 2016) given their high model capacity. As such, multiple lines of work have been proposed recently to effectively combine clean (or gold) labeled data with noisy (or weak) supervision data for more effective learning. One line of work focused on selecting samples from the noisy data that are likely to be correct using co-teaching or curriculum learning (Jiang et al. 2017; Han et al. 2018). Another line of work tries to re-weight the weak instances for selective training (Ren et al. 2018; Shu et al. 2019), instead of either including or excluding them. Some of these approaches use a meta-learning framework to assign importance scores to each

sample in the noisy training set such that the ones with higher weights can contribute more to the main model training (Ren et al. 2018; Shu et al. 2019).

One of the limitations of label re-weighting is that it is limited to up or down weighting the contribution of an instance in the learning process. An alternative approach relies on the idea of label correction. It aims to correct the noisy labels based on certain assumptions about the weak label generation process. In a sense, label correction aims to go beyond selecting or assigning high weights to useful examples to also altering the assigned labels of incorrectly labeled examples. However, previous methods of label correction rely on assumptions about the weak label generation process and thus often involves two independent steps: (1) estimating a label corruption matrix (Hendrycks et al. 2018), (2) training a model on the noisy data leveraging the corruption matrix. Estimating the corruption matrix often involves assumptions about the noise generation process, such as assuming that the noisy label is only dependent on the true label and is independent of the data itself (Hendrycks et al. 2018).

In this paper, we adopt label correction to address the problem of learning with noisy labels, but from a meta-learning perspective. We term our method meta label correction (MLC). Specifically, we view the label correction procedure as a meta-process, meaning that its objective is to provide corrected labels for the examples with weak labels. Meanwhile, the main predictive model of interest is then trained with such corrected labels generated by the meta-model. Both the meta-model and the main model are learned by a bi-level optimization procedure concurrently. This allows the model to maximize the performance on the clean data set (i.e., the clean labels serve as a validation set w.r.t. the noisy set) by updating the label correction process in a differentiable manner. MLC extends work on re-weighting and correction leveraging the advantages of both approaches. In contrast to meta-learning based instance re-weighting, which only considers up or down weighting the given noisy label,¹ MLC provides a more refined way of leveraging noisy labels by exploring all possible classes in the label space. In contrast to previous label correction methods, MLC doesn't make assumptions about the underlying label noises and concurrently learns a correction model with the main model. Figure 1 shows examples where label re-weighting could at best down-weight noisy samples, reducing their impact on the learning process. On the other hand, MLC can successfully correct the noisy labels to the true ones.

Meta learning has been successfully used for many applications including hyper-parameter tuning (Maclaurin, Duvenaud, and Adams 2015), optimizer learning (Ravi and Larochelle), model selection (Pedregosa 2016), adaptation to new tasks (Finn, Abbeel, and Levine 2017) and neural architecture search (Liu, Simonyan, and Yang 2018). This work leverages meta-learning for label correction to learn from noisy data and makes the following contributions:

- We pose the problem of learning from weak (noisy) supervision as a meta label correction where a correction

¹For classification, instance re-weighting is effectively equivalent to label re-weighting.

network is trained as a meta process to provide reliable labels for the main models to learn;

- We compare and contrast re-weighting and correction as two strategies for handling noisy labeled data;
- We conduct experiments on a combination of 3 image recognition and 4 large-scale text classification tasks with varying noise levels and types, including real-world noisy labels. We show that the proposed method outperform previous best methods on label correction and re-weighting, demonstrating the power of the proposed method.

Related Work

Labeled data largely determines whether a machine learning system can perform well on a task or not, as noisy label or corrupted labels could cause dramatic performance drop (Nettleton, Orriols-Puig, and Fornells 2010). The problem gets even worse when an adversarial rival intentionally injects noises into the labels (Reed et al. 2014). Thus, understanding, modeling, correcting, and learning with noisy labels has been of interest at large in the research communities (Natarajan et al. 2013; Frénay and Verleysen 2013). Several works (Mnih and Hinton 2012; Patrini et al. 2017; Sukhbaatar et al. 2014; Larsen et al. 1998) have attempted to address the weak labels by modifying the model's architecture or by implementing a loss correction. (Sukhbaatar et al. 2014) introduced a stochastic variant to estimate label corruption, however the methods have to have access to the true labels, rendering it inapplicable when no true labels are present. A forward loss correction adds a linear layer to the end of the model and the loss is adjusted accordingly to incorporate learning about the label noise. (Patrini et al. 2017) also make use of the forward loss correction mechanism, and propose an estimate of the label corruption estimation matrix which relies on strong assumptions, and does not make use of clean labels that might be available for a portion of the data set. Similar idea is also explored in (Goldberger and Ben-Reuven 2016).

In this paper, we limit our attention to the setting where in addition to a large amount of weakly labeled data, there is also a small set of clean data available. Under this setup, two major lines of work have been proposed to solve learning problem with noisy labels and we briefly review them here.

Learning with label correction

The first line of work aims to correct the weak labels as much as possible by imposing assumptions of how the noisy labels are generated from its underlying true labels. Consider the problem of classifying the data into k categories, label correction involves estimating a label corruption matrix $C_{k \times k}$ whose entry $C_{i,j}$ denotes the probability of observing weak label for class i while the underlying true class label is actually j . Gold loss correction (Hendrycks et al. 2018) falls into this category; a key drawback of this line of work is that the label perturbing matrix is estimated in an adhoc way and also that the estimation process is separate from the main model process, hence allowing no feedback from the main model to the estimation process.

Learning to re-weight training instances

Knowing that not all training examples are equally important and useful for building a main model given the noise, another line of work for learning with weak supervision focuses on selecting a subset of samples from the noisy data that are likely to be correct (Jiang et al. 2017; Han et al. 2018). Instead of discarding examples, an extension of this idea focused on assigning learn-able weights to each example in the training noisy set. The goal is to assign a weight for each training example, indicating how useful the example is, such that the main model could use these weights to improve performance on a separate validation set (the clean set) (Ren et al. 2018). The example weights are essentially hyper-parameters for the main model and can be learned by formulating a bi-level optimization problem. This framework allows the example weights learning and the main model to communicate with each other and a better model could be learned.

Our work follows the *learning to correct* framework by learning to model and correct the label noise in the noisy examples. Instead of separately handling the label correction and model learning steps, we propose a meta-learning approach to co-optimize for the two steps. We show that our model can outperform state-of-the-art methods for both learning to correct and learning to re-weight.

Meta Label Correction

Following (Veit et al. 2017; Li et al. 2017; Xiao et al. 2015; Ren et al. 2018), we assume that the setup of learning with weak supervision involves two sets of data: a small set of data with clean/trusted labels and a large set of data with noisy/weak labels. Typically the clean set is much smaller compared to the noisy set, due to scarcity of expert labels and high labeling costs. Training directly on the small clean set often tends to be sub-optimal, as too little data can easily cause over-fitting. Training directly on the noisy set (or a combination of the noisy and clean sets) also tends to be sub-optimal, as large high-capacity models can fit and memorize the noise (Zhang et al. 2016). Note that unlike some of the work in this area, e.g., (Veit et al. 2017), we do not require having trusted and noisy labels for the same instances.

One advantage of the label correction approach is that it allows us to combine clean labels and *corrected* noisy labels in the learning process. Our proposed approach adopts the label correction methodology while also co-optimizing the label correction process together with the main model process through a unified meta-learning framework. We achieve that by training a meta learner (meta model) that tries to correct the noisy labels and a main model that tries to build the best predictive model with corrected labels coming from the meta model, allowing the meta model and main model to reinforce each other.

A meta-learning based formulation for label correction

We describe the framework in detail as follows. Given a set of clean data examples $D = \{\mathbf{x}, \mathbf{y}\}^m$ and a set of weak (noisy) data examples $D' = \{\mathbf{x}, \mathbf{y}'\}^M$ with m much smaller than M . To best exploit the information carried by the weak labels,

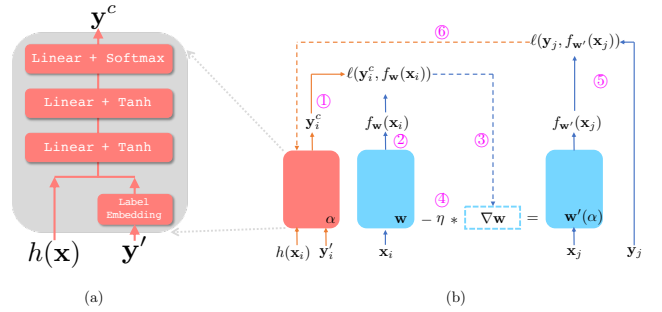


Figure 2: MLC Architecture and computation flow. $(\mathbf{x}, \mathbf{y}')$ denotes a pair of sample with weak label and (\mathbf{x}, \mathbf{y}) is a pair of sample with clean label. (a) Architecture of the label correction network, where $h(\mathbf{x})$ is a feature representation of input \mathbf{x} ; (b) The computation flow of updating the LCN. In order, operations are: ① Feed the weak instance to the LCN and get its corrected label. ② Feed the data instance to the current classifier and compute the logits for prediction, ③ Compute the loss with the logits and corrected label, and compute the gradient of the loss with respect to the parameter of the classifier. Note that the gradient will be a function of the parameters of the LCN. ④ Update the classifier parameter while keeping the computation graph for the gradient, ⑤ feed a pair of clean instance to the new model and compute its loss, ⑥ Compute the gradient of the loss w.r.t the parameter to LCN and update the LCN.

we propose to construct a label correction network (LCN), serving as a *meta model*, which takes a pair of noisy data example and its weak label as input and attempts to produce a corrected label for this data example. The LCN is parameterized as a function with parameters α , $y_c = g_\alpha(h(\mathbf{x}), y')$ to correct the weak label y' of example feature $h(\mathbf{x})$ to a more accurate one. (Note that y_c is a soft label, i.e., a multinomial distribution for all possible classes and the subscription in y_c emphasizes that it's generating a corrected label). Meanwhile, the *main model* f , that we aim to train and use for prediction after training, is instantiated as another function with parameters \mathbf{w} , $y = f_{\mathbf{w}}(\mathbf{x})$.

Without linking the two models, there's no way to enforce that: 1) the corrected label from LCN for an example from the meta model g is indeed a meaningful one, let alone a corrected one, since directly training the LCN is not possible without clean labels for the noisy examples; 2) The main model f ends up fitting onto the correct true labels, if the labels provided by the LCN do not align with the unknown true labels. Fortunately, the two models can be linked together via a bi-level optimization framework, motivated by the intuition that *if the labels generated by the LCN are of high quality, then a classifier trained with such corrected labels as supervision should achieve low loss on a separate set of clean examples*. Formally, this can be formulated as the following bi-level optimization problem:

$$\begin{aligned} \min_{\alpha} \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \in D} \ell(y, f_{\mathbf{w}_\alpha^*}(\mathbf{x})) \\ \text{s.t. } \mathbf{w}_\alpha^* = \arg \min_{\mathbf{w}} \mathbb{E}_{(\mathbf{x}, \mathbf{y}') \in D'} \ell(g_\alpha(h(\mathbf{x}), y'), f_{\mathbf{w}}(\mathbf{x})) \end{aligned} \quad (1)$$

where $\ell(\cdot)$ is the loss function for classification, i.e., cross-

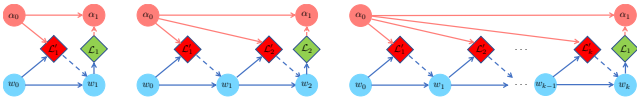


Figure 3: Different bi-level optimization learning strategies. (a) For any step, the optimal solution \mathbf{w}_α^* is approximated with one-step look-ahead optimizer update, and used as the trained model to compute the evaluation loss \mathcal{L}_1 , which is then used to update the meta parameters α . (b) The meta parameters α are updated once every two updates from the main model parameters \mathbf{w} , i.e., a two-step look-ahead update is used to approximate the optimal solution \mathbf{w}_α^* . (c) Extending to k -step look-ahead approximation of the optimal solution \mathbf{w}_α^* for updating the current α . With a larger k , the approximation of \mathbf{w}_α^* will be more accurate with the cost of fewer updates for the meta parameters α (assuming a fixed number of updates for the main model parameters). We find using a k in the range of $1 \sim 10$ works well for our experiments.

entropy². We term this framework as Meta Label Correction (MLC); Figure 2 provides an overview of the framework. Note that to facilitate a light-weight design of the LCN, we take $h(\mathbf{x})$ to be the feature representations from the main classifier, e.g., representations from the last layer, **with stop-gradient operators before feeding to LCN to prevent gradient flowing the LCN back to the main model.**

In this bi-level optimization, the LCN parameters α are the upper parameters (or meta parameters) while the main model parameters \mathbf{w} are the lower parameters (or main parameters). Like many other work involving bi-level optimizations, exact solutions to Problem (1) requires solving for the optimal \mathbf{w}^* whenever α gets updated. This is both analytically infeasible and computationally expensive, particularly when the main model f is complex, such as ResNet (He et al. 2016) for image recognition and BERT (Devlin et al. 2018) for text classification.

Gradient-based optimization for bi-level optimization.

Outside of label correction research, various other studies, including differentiable architecture search (Liu, Simonyan, and Yang 2018), few-shot meta learning (Finn, Abbeel, and Levine 2017; Nichol, Achiam, and Schulman 2018), have used similar bi-level formulation as Problem (1). Instead of solving for the optimal for \mathbf{w}^* for each α , one step of SGD update for \mathbf{w} to approximate the optimal main model for a given α has been employed³

$$\mathbf{w}_\alpha^* \approx \mathbf{w}'(\alpha) = \mathbf{w} - \eta \nabla_{\mathbf{w}} \mathcal{L}_{D'}(\alpha, \mathbf{w}) \quad (2)$$

where $\mathcal{L}_{D'}(\alpha, \mathbf{w}) \triangleq \mathbb{E}_{(\mathbf{x}, y') \in D'} \ell(g_\alpha(\mathbf{x}, y'), f_{\mathbf{w}}(\mathbf{x}))$ is a shorthand for the lower-level objective function and η is the learning rate for the main model f . Denoting the upper-level objective function (meta loss) as $\mathcal{L}_D(\mathbf{w}) \triangleq \mathbb{E}_{(\mathbf{x}, y) \in D} \ell(y, f_{\mathbf{w}}(\mathbf{x}))$, the proxy optimization problem with

²Note that cross-entropy loss also works with soft labels in the lower-level optimization of Problem (1)

³For clarity, we derive this with plain SGD, however this also holds for most variants of SGD, including SGD with momentum, Adam (Kingma and Ba 2014) (See the Appendix for details).

while not converged do

 Update meta parameters α by descending Eq. (4)

 Update model parameters \mathbf{w} by descending $\nabla_{\mathbf{w}} \mathcal{L}_{D'}(\alpha, \mathbf{w})$

end

Algorithm 1: MLC - Meta Label Correction

one-step look ahead SGD now becomes

$$\min_{\alpha} \mathcal{L}_D(\mathbf{w}'(\alpha)) = \mathcal{L}_D(\mathbf{w} - \eta \nabla_{\mathbf{w}} \mathcal{L}_{D'}(\alpha, \mathbf{w})) \quad (3)$$

Efficient meta-gradient with k -step SGD from main-parameters

Different from DARTS (Liu, Simonyan, and Yang 2018), the meta loss $\mathcal{L}_D(\mathbf{w}_\alpha^*)$ depends only implicitly on the meta parameters α via the trained model \mathbf{w}_α^* , hence a more accurate estimate of the optimal solution \mathbf{w}_α^* for the current LCN parameters α is desired. To this end, we propose to employ a k -step ahead SGD update as the proxy estimate for the optimal solution. Figure 3 demonstrates the parameter updating schemes for different k . **A larger k principally provides less noisy estimate for the optimal solution \mathbf{w}_α^* , however it also results in longer dependencies over the past k iterations, which requires caching k copies of the model parameters \mathbf{w} .** To address this, we further propose to approximate the current meta-parameter gradient with information from the previous k step as follows (See Appendix for a derivation sketch)

$$\frac{\partial \mathcal{L}_D(\mathbf{w}')}{\partial \alpha} \approx g_{\mathbf{w}'} (I - \Lambda) \frac{g_{\mathbf{w}'}^\top}{\|g_{\mathbf{w}'}\|^2} \frac{\partial \mathcal{L}_D(\mathbf{w})}{\partial \alpha} - g_{\mathbf{w}'} \Lambda H_{\alpha, \mathbf{w}} \quad (4)$$

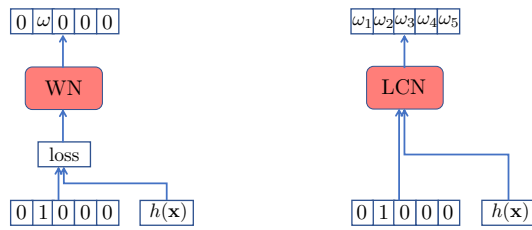
where \mathbf{w}' is the model parameter for next step, $g_{\mathbf{w}}$ is a short hand for the gradient of the training loss w.r.t \mathbf{w} , Λ is a diagonal matrix representing the current learning rates for all parameters in \mathbf{w} , and $H_{\alpha, \mathbf{w}}$ is a short hand for $\frac{\partial^2}{\partial \alpha \partial \mathbf{w}} \mathcal{L}_{D'}(\alpha, \mathbf{w})$. The second term can be efficiently computed as

$$\begin{aligned} g_{\mathbf{w}'} \Lambda H_{\alpha, \mathbf{w}} &= \nabla_{\alpha, \mathbf{w}}^2 \mathcal{L}_{D'}(\alpha, \mathbf{w}) \Lambda \nabla_{\mathbf{w}'} \mathcal{L}_D(\mathbf{w}') \\ &= \nabla_{\alpha} \left(\nabla_{\mathbf{w}}^\top \mathcal{L}_{D'}(\alpha, \mathbf{w}) \Lambda \nabla_{\mathbf{w}'} \mathcal{L}_D(\mathbf{w}') \right) \end{aligned} \quad (5)$$

Algorithm 1 outlines an iterative procedure to solve the above proxy problem with k -step look ahead SGD for the main model.

Training with soft labels from LCN

Not only does the LCN explicitly model the dependency of the corrected label on both the data example and its noisy label, but also it ensures that the output from the LCN is a valid categorical distribution over all possible classes. **Soft labels are crucial in MLC as they make gradient propagation back to the meta model from the main model possible. However, when the main model takes these examples with soft corrected labels, it brings difficulty to training due to the additional uncertainty in the corrected labels.** This can be alleviated by the following strategy. In training for each batch



(a) Weighting-Network (WN) in Meta-WN (Shu et al. 2019) (b) Label Correction Network (LCN) in MLC (Ours)

Figure 4: Comparisons between Meta-WN vs MLC on different treatments to the noisy labels. Total number of classes assumed to be 5 for illustration purpose. (a) Meta-WN; (b) MLC.

of clean data, we split it into two parts, with one serving as the clean evaluation set and add the other to the training process for f , as a small portion of the clean set will provide clean guidance for training, to ease model training. This has been shown to be effective in similar settings (Ranzato et al. 2015).

Remark: Label correction vs label reweighting

To address noisy labels, Meta-WN (Shu et al. 2019) leverages the Weight-Network (WN) as the meta-module to **reweight** the given noisy label, while MLC aims to provide a more refined treatment, i.e., to **correct** the noisy label. More explicitly, Figure 4 demonstrates the difference of the underlying operations between Meta-WN and MLC. To highlight

- For an input, Meta-WN tries to learn a weight for the given noisy class only, while ignoring all other possible classes (demonstrated by the single non-negative weight ω for Class 2 while 0 for all other classes in Figure 4(a)), while since MLC tries to correct the given weak label, essentially it considers all possible classes (demonstrated by the full non-negative vector resulting from the final softmax layer of the LCN, i.e., $(\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ for all classes with $\omega_1 + \omega_2 + \omega_3 + \omega_4 + \omega_5 = 1$, essentially weighting all possible classes)
- Another key difference between MLC and Meta-WN relies on the information bottleneck to their corresponding meta-modules. For Meta-WN doesn't directly take an pair of data and noisy label as input, but rather relies on the *scalar* loss of the classifier that particular input incurs as input. In other words, the meta-module lacks the ability to differentiate the different input pairs if the loss values for them are similar, effectively limiting the modeling capacity of the meta-module. While the LCN in MLC directly takes the data feature $h(x)$ and its weak label as input, allowing a more flexible treatment and enabling the LCN to identify different information brought by different input pairs.

Experiments

To test the performance of MLC, we conduct experiments on a combination of three image recognition and four text classification tasks, and compare with previous state-of-the-art approaches for learning with weak/noisy labels, under

different types of noisy labels.

Datasets and Setup

Datasets. We evaluate our method on 3 image recognition datasets, CIFAR-10, CIFAR-100 (Krizhevsky et al. 2009) and Clothing1M (Xiao et al. 2015) and 4 large-scale multi-class text classification benchmark datasets, that are widely used by text classification research (Zhang, Zhao, and LeCun 2015; Xie et al. 2019; Dai et al. 2019; Yang et al. 2016; Conneau et al. 2016), AG news, Amazon reviews, Yelp reviews and Yahoo answers. Information about all datasets is summarized in Table 1.

Noisy label sources Following related work (Hendrycks et al. 2018; Ren et al. 2018; Shu et al. 2019), for each dataset, we sample a portion of the entire training set as the clean set (except for Clothing1M). To ensure a fair and consistent evaluation, we use only 1000 images as clean set for both CIFAR-10 and CIFAR-100, and only 100 instances per class for the four large scale text classification data sets. The noisy sets are generated by corrupting the labels of all the remaining data points based on the following two setting:

Uniform label noise (UNIF). For a dataset with C classes, a clean example with true label y is randomly corrupted to all possible classes y' with probability $\frac{\rho}{C}$ and stays in its original label with probability $1 - \rho$. (Note the corrupted label might also happen to be the original label, hence the label has probability of $1 - \rho + \frac{\rho}{C}$ to stay uncorrupted.)

Flipped label noise (FLIP). For a dataset with C classes, a clean example with true label y is randomly flipped to *one of the rest $C - 1$ classes* with probability ρ and stays in its original label with probability $1 - \rho$.

We vary ρ in the range of $[0, 1]$ to simulate different noise levels for both types. We emphasize that both simulated noise types makes the simplifying assumption that given the true label the noisy label doesn't depend on the data itself. Hence we also evaluate all the methods on another source of noisy labels:

Real-world noisy labels. Clothing1M (Xiao et al. 2015) is a dataset where noisy labels for images are devised by leveraging user tags as proxy annotations. As Clothing1M is the only dataset that comes with real-world noisy labels, we use its original split of clean and noisy sets.

Finally, we also note that regardless of noise types, none of the methods tested in this paper is aware of the label corruption probability ρ nor do they have knowledge about which data sample in the noisy set is actually corrupted.

Baseline methods and model architectures

We focus our evaluation of MLC against state-of-the-art methods for learning with weak supervision from two different themes, i.e., (Hendrycks et al. 2018) for label correction (denoted by GLC hereafter) and instance re-weighting with meta learning (Shu et al. 2019) (denoted by MW-Net hereafter). Note that GLC and MW-Net were shown to consistently outperform other methods such as training on clean data only, cleaning on weak data only, combining clean and weak data, as well as more sophisticated models for combining the clean and weak labels such as distillation (Li et al. 2017) and forward loss correction (Sukhbaatar et al. 2014). Additionally,

Table 1: Dataset statistics and classifier architectures used. Note the clean set is significantly smaller than the noisy label set.

Dataset	CIFAR-10	CIFAR-100	Clothing1M	AG	Amazon-5	Yelp-5	Yahoo
# classes	10	100	14	4	5	5	10
Train	50K	50K	1.05M	120K	3M	650K	1.4M
Test	10K	10K	10K	7.6K	650K	50K	60K
Clean	1000	1000	50K	400	500	500	1000
Noisy	49K	49K	1M	119.6K	~ 3M	649.5K	~ 1.4M
Classifier	ResNet 32		ResNet 50	Pre-trained BERT-base			

Table 2: Mean accuracies on all data sets. Each cell represents the average runs over two noise types and 10 noise levels. A $k = 5$ (5-step ahead SGD) is used for all experiment. (Each configuration is run for 5 times and the mean is reported)

Datasets (# clean labels)	CIFAR-10 (10 × 100)	CIFAR-100 (100 × 10)	AG (4 × 100)	Yelp-5 (5 × 100)	Amazon-5 (5 × 100)	Yahoo (10 × 100)
MW-Net (Shu et al. 2019)	65.12	39.96	75.91	51.27	49.49	60.18
GLC (Hendrycks et al. 2018)	86.62	50.50	83.88	60.12	60.31	68.03
MLC (Ours)	86.81	53.68	92.22	62.61	61.21	73.72

Table 3: Test set accuracies on Clothing1M with real-world noisy labels ($k = 5$)

Method	Forward (Patrini et al. 2017)	Joint Learning (Tanaka et al. 2018)	MLNT (Li et al. 2019)	MW-Net (Shu et al. 2019)	GLC (Hendrycks et al. 2018)	MLC (Ours)
Accuracy	69.84	72.23	73.47	73.72	73.69	75.78

MW-Net was shown to outperform a slightly different variant for instance re-weighting with meta-parameters (Ren et al. 2018). As such, we do not show results from these methods here.

For fair and consistent comparisons, we use the same classifier architectures for all methods, i.e., ResNet 32 for CIFAR-10 and CIFAR-100, ResNet 50 pretrained from ImageNet for Clothing1M, and pre-trained BERT-base for the four large-scale text data sets. We implement all models and experiments in PyTorch. All models are trained with the same number of training epochs for the same dataset.⁴ Code for reproducing the results in this paper will be made publicly available.

LCN architecture. We use the same LCN architecture for MLC across all settings. The architecture for the LCN used through all experiments is as follows (Figure 2(a)):

An embedding layer of size $(C, 128)$ to embed the input noisy labels, followed by a three-layer feed-forward network with dimensions of $(128 \times \text{xdim}, \text{hdim})$, $(\text{hdim}, \text{hdim})$, (hdim, C) respectively. \tanh is used as the nonlinear activation function in-between them and lastly a Softmax layer to output a categorical distribution as the corrected labels

where C is the number of classes, xdim is the feature dimension of input \mathbf{x} from the last layer from the main classifier, i.e., 64 from ResNet32 for CIFAR-10 and CIFAR-100, 2048 from ResNet 50 for Clothing1M and 768 from BERT-base for text datasets and hdim is hidden dimension for the LCN (64 for CIFAR-10, CIFAR-100 and Clothing1M, and 768 for all the text datasets.)

⁴Complete training details can be found in the Appendix.

Main results

MLC on image recognition. We start by comparing all methods on the standard image recognition datasets. Table 2 presents the averaged accuracies across multiple configurations (two noise types, 10 noise levels) with $k = 5$. The table shows that MLC consistently outperforms other methods over all datasets. In addition, on Clothing1M with real noisy labels, MLC outperforms all baseline methods and improves over GLC and MW-Net by over 2 points in accuracy, suggesting its ability to capture better data-dependent label corruptions via the meta-learning framework.

MLC on text classification. Table 2 also presents the mean accuracies of MLC on 4 large text data sets with pre-trained BERT-base as its main classifier. Overall, label reweighting (MW-Net) seems insufficient to fully address the text classification problem; label correction approaches demonstrate much higher performances, while MLC achieves the best thanks to its nature of combining of both label correction and the data-driven meta-learning framework.

Analysis and ablation studies

Effects of noise levels ρ and k for MLC. Figure 5 presents the results of all methods under UNIF with noise levels from 0 to 1.0 with a step size of 0.1. It’s clear that, since Meta-WN only attempts to re-weight the observed weak label, its performance decreases significantly when the noise level goes up, as the given label turns more likely to be the wrong label thus re-weighting for this case is insufficient; while label correction based methods (GLC and MLC) show to be robust against severe label noises. This is consistent with results reported in (Hendrycks et al. 2018) where label correction was shown to perform well even in extreme noise level situations. Moreover, we observe that MLC is more effective in

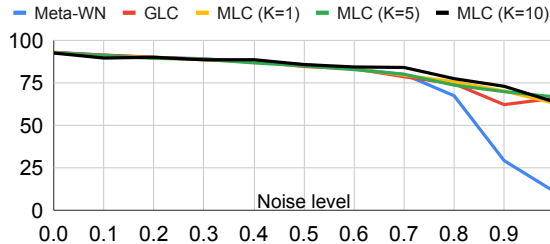


Figure 5: Test accuracy w.r.t noise levels

doing this than previous label correction methods for severe noise. In terms of the number of look-ahead steps, k , used to compute the meta-gradient, the value of k does not seem to have an impact on MLC’s performance when the noise level is low; however when the noise level is high (more than 0.6), a larger k leads to higher test accuracy, validating the strategy of using multiple steps to compute the meta-gradients. Similar trends can be observed on FLIP (See Appendix).

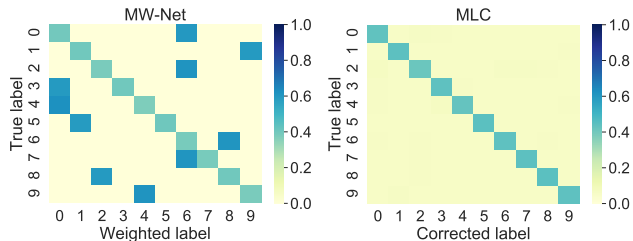


Figure 6: (a) Heatmap of learned weights of MW-Net w.r.t to the true labels; note that MW-Net does not alter the noisy labels but only assigns them weight. (b) Heatmap of probability distribution of the corrected labels of MLC w.r.t to the true labels; note that MLC can alter the input label (For all examples in the test set with noise level $\rho = 0.6$.)

Meta net evaluation We perform additional experiments to understand what the meta model, i.e., the LCN, actually learns after model convergence. Additionally, we seek to quantify the benefit of correcting noisy labels, v.s. re-weighting instances. We use the FLIP setting to generate corrupted labels for instances in the test set in CIFAR-10 and feed them to the meta nets of both MLC and Meta-WN. MLC will produce a probability distribution over all possible classes where Meta-WN will assign a scalar weight to each instance. Note that, for CIFAR-10, we know which of the noisy label is actually correct and which is not but neither of the models have access to this information. Ideally, Meta-WN will assign higher weights to the correct instances and lower weights to the incorrect ones. Similarly, MLC should keep the label of correct instances as is and alter the labels of the incorrect ones. We see from Figure 7 that this is actually the case. **On average, both model seem to be able to distinguish between the correct and incorrect labels. However, for incorrect labels, Meta-WN can only down-weight the sample reducing the dependence of the training process on it. MLC goes beyond this by also trying to change the label to assign the sample to the correct one, allowing the main model to fully leverage it.** We can see from Figure 6 that it does that

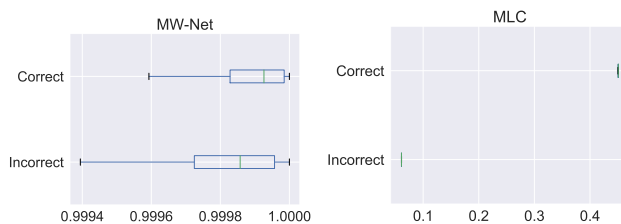


Figure 7: Effect of reweighting vs correction for FLIP with $\rho = 0.6$. (Left) Weights assigned by MW-Net w.r.t to the input noisy label; (Right) Correction probabilities learned by LCN to the input noisy label.

successfully. On other other hand, MW-Net can only assign a weight to the noisy label. More information about the correction accuracy of the meta model in MLC can be found in the Appendix.

MLC training dynamics Figure 8(a,b,c) shows the training progress for one run on the CIFAR-10 with UNIF under different noise levels. We monitor a set of different metrics during training, including the loss function on the noisy data with corrected labels, loss function on clean data, and the test set accuracy as training progresses. The figure shows that both losses decrease and test accuracy increases as the training process progresses. Note that with larger noise levels (hence more difficult cases), training with MLC gets harder (as seen by the slightly higher loss on clean data and loss on noisy data). However, MLC still converges and achieves good results on test set as shown in Figure 8(c).

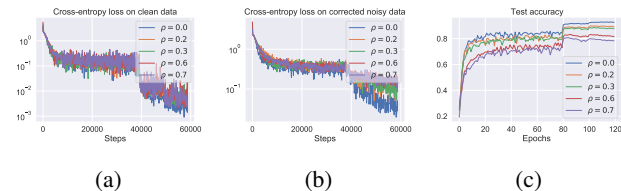


Figure 8: (a,b,c) Loss and test set accuracy dynamics (losses are in log-scale). The two jump-points in the curves are due to the decay of learning rates at 60th and 80th Epoch, or $\sim 40K$ and $50K$ steps equivalently.

Conclusions

In this paper, we address the problem of learning with noisy labels from a meta-learning perspective. Specifically, we propose to use a meta network to correct the noisy labels from the data set, and a main classifier network is trained to fit the example to a provided label, i.e., corrected labels for the noisy examples and true labels for the clean ones. The meta network and main network are jointly optimized in a bi-level optimization fashion; to address the computation challenge, we employ a k -step ahead SGD update to compute the meta-gradient. Empirical experiments on three image recognition and four text classification tasks with various label noise types show the benefits of label correction over instance re-weighting and demonstrate the strong performance of MLC over previous methods leveraging noisy labels.

References

- Conneau, A.; Schwenk, H.; Barrault, L.; and Lecun, Y. 2016. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781* .
- Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q. V.; and Salakhutdinov, R. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860* .
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* .
- 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.
- Fréney, B.; and Verleysen, M. 2013. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems* 25(5): 845–869.
- Goldberger, J.; and Ben-Reuven, E. 2016. Training deep neural-networks using a noise adaptation layer .
- 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*, 10477–10486.
- Jiang, L.; Zhou, Z.; Leung, T.; Li, L.-J.; and Fei-Fei, L. 2017. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. *arXiv preprint arXiv:1712.05055* .
- Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .
- Krizhevsky, A.; et al. 2009. Learning multiple layers of features from tiny images .
- Larsen, J.; Nonboe, L.; Hintz-Madsen, M.; and Hansen, L. K. 1998. Design of robust neural network classifiers. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, volume 2, 1205–1208. IEEE.
- Li, J.; Wong, Y.; Zhao, Q.; and Kankanhalli, M. S. 2019. Learning to learn from noisy labeled data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5051–5059.
- Li, Y.; Yang, J.; Song, Y.; Cao, L.; Luo, J.; and Li, L.-J. 2017. Learning From Noisy Labels With Distillation. In *The IEEE International Conference on Computer Vision (ICCV)*.
- Liu, H.; Simonyan, K.; and Yang, Y. 2018. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055* .
- Maclaurin, D.; Duvenaud, D.; and Adams, R. 2015. Gradient-based hyperparameter optimization through reversible learning. In *International Conference on Machine Learning*, 2113–2122.
- Mnih, V.; and Hinton, G. E. 2012. Learning to label aerial images from noisy data. In *Proceedings of the 29th International conference on machine learning (ICML-12)*, 567–574.
- Natarajan, N.; Dhillon, I. S.; Ravikumar, P. K.; and Tewari, A. 2013. Learning with noisy labels. In *Advances in neural information processing systems*, 1196–1204.
- Nettleton, D. F.; Orriols-Puig, A.; and Fornells, A. 2010. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial intelligence review* 33(4): 275–306.
- Nichol, A.; Achiam, J.; and Schulman, J. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999* .
- 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.
- Pedregosa, F. 2016. Hyperparameter optimization with approximate gradient. *arXiv preprint arXiv:1602.02355* .
- Ranzato, M.; Chopra, S.; Auli, M.; and Zaremba, W. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732* .
- Ravi, S.; and Larochelle, H. Optimization as a Model for Few-Shot Learning. In *5th International Conference on Learning Representations, ICLR 2017*.
- Reed, S.; Lee, H.; Anguelov, D.; Szegedy, C.; Erhan, D.; and Rabinovich, A. 2014. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596* .
- 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.
- 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.
- Veit, A.; Alldrin, N.; Chechik, G.; Krasin, I.; Gupta, A.; and Belongie, S. 2017. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 839–847.

- Xiao, T.; Xia, T.; Yang, Y.; Huang, C.; and Wang, X. 2015. Learning from Massive Noisy Labeled Data for Image Classification. In *CVPR*.
- Xie, Q.; Dai, Z.; Hovy, E.; Luong, M.-T.; and Le, Q. V. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.
- Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 1480–1489.
- Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2016. Understanding deep learning requires rethinking generalization.
- Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, 649–657.
- Zhou, Z.-H. 2017. A brief introduction to weakly supervised learning. *National Science Review* 5(1): 44–53.