2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) | 978-1-6654-9346-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/WACV56688.2023.00392

Adaptive Sample Selection for Robust Learning under Label Noise

Deep Patel and P S Sastry
Indian Institute of Science,
Bangalore, India - 560012
deeppatel, sastry@iisc.ac.in

Abstract

Deep Neural Networks (DNNs) have been shown to be susceptible to memorization or overfitting in the presence of noisily-labelled data. For the problem of robust learning under such noisy data, several algorithms have been proposed. A prominent class of algorithms rely on sample selection strategies wherein, essentially, a fraction of samples with loss values below a certain threshold are selected for training. These algorithms are sensitive to such thresholds, and it is difficult to fix or learn these thresholds. Often, these algorithms also require information such as label noise rates which are typically unavailable in practice. In this paper, we propose an adaptive sample selection strategy that relies only on batch statistics of a given mini-batch to provide robustness against label noise. The algorithm does not have any additional hyperparameters for sample selection, does not need any information on noise rates and does not need access to separate data with clean labels. We empirically demonstrate the effectiveness of our algorithm on benchmark datasets.[1]

1. Introduction

The deep learning models, which are highly effective in many applications, need vast amounts of training data. Such large-scale labelled data is often generated through crowd-sourcing or automated labeling, which naturally cause random labelling errors. In addition, subjective biases in human annotators too can cause such errors. The training of deep networks is adversely affected by label noise and hence robust learning under label noise is an important problem of current interest.

In recent years many different approaches for robust learning of classifiers have been proposed, such as, robust loss functions [9, 6, 53, 42], loss correction [35], meta-learning [25, 43], sample reweighting [38, 39, 41, 16, 11], etc. In this paper we present a novel algorithm that adaptively selects samples based on the statistics of observed loss values in a minibatch and achieves good robustness to label noise. Our algorithm does not use any additional system for learning weights for examples, does not need extra data with clean labels and does not assume any knowledge of noise rates. The algorithm is motivated by curriculum learning and can be thought of as a way to design an adaptive curriculum.

The curriculum learning [4, 21] is a general strategy of sequencing of examples so that the networks learn from the 'easy' examples well before learning from the 'hard' ones. This is often brought about by giving different weights to different examples in the training set. Many of the recent algorithms for robust learning based on sample reweighting can be seen as motivated by a similar idea. A good justification for this approach comes from some recent studies [50] that have shown that deep networks can learn to achieve zero training error on completely randomly labelled data, a phenomenon termed as 'memorization'. Further studies such as [3, 29] have shown that the networks, when trained on noisily-labelled data, learn simpler patterns first before overfitting to the noisily-labelled data.

In the last few years, several strategies have been proposed that aim to select (or give more weightage to) 'clean' samples for obtaining a degree of robustness against label noise (e.g., [16, 11, 49, 47, 27, 38, 13]). All such methods essentially employ the heuristic of 'small loss' for sample selection wherein (a fraction of) small-loss valued samples are preferentially used for learning the network. Many of these methods use an auxiliary network to assess the loss of an example or to learn to map loss values to sample weights. Such methods need additional computing resources to learn multiple networks and may also need separate clean data (without label noise) and the methods involve careful choice of additional hyperparameters. In general, it is difficult to directly relate the loss value of an example with the reliability of its label. Loss value of any specific example is itself a function of the current state of learning and it evolves with epochs. Loss values of even

[1] Codes for reproducibility will be made available here: https://github.com/dbp1994/masters_thesis_codes/tree/main/BARE

2642-9381/23/$31.00 ©2023 IEEE
DOI 10.1109/WACV56688.2023.00392

clean samples may change over a significant range during the course of learning. Further, the loss values achievable by a network even on clean samples may be different for examples of different classes.

Motivated by these considerations, we propose a simple, adaptive selection strategy called *BAtch REweighting* (**BARE**). Our algorithm utilizes the statistics of loss values in a mini-batch to compute the threshold for sample selection in that mini-batch. Since, it is possible that this automatically calculated threshold is different for different mini-batches even within the same epoch, our method amounts to using a dynamic threshold which naturally evolves as learning proceeds. In addition, while calculating the batch statistics we take into consideration the class labels also and hence the dynamic thresholds are also dependent on the given labels of the examples.

The main contribution of this paper is an adaptive sample selection strategy for robust learning that is simple to implement, does not need any clean validation data, needs no knowledge at all of the noise rates and also does not have any hyperparameters in the sample selection strategy. It does not need any auxiliary network for sample selection. We empirically demonstrate the effectiveness of our algorithm on benchmark datasets: MNIST [22], CIFAR-10 [19], and Clothing-1M [46] and show that our algorithm is much more efficient in terms of time and has as good or better robustness compared to other algorithms for different types of label noise and noise rates.

The rest of the paper is organized as follows: Section 2 discusses related work, Section 3 discusses our proposed algorithm. Section 4 discusses our empirical results and concluding remarks are provided in Section 5.

2. Related Work

Curriculum learning (CL) as proposed in [4] is the designing of an (optimal) manner of sequencing of training samples (based on a notion of *easiness* of an example) to improve the performance of the learning system. A curriculum called Self-Paced Learning (SPL) is proposed in [21] wherein easiness is decided upon based on how small the loss values are. A framework to unify CL and SPL is proposed in [15]. SPL with diversity [14] proposes a sample selection scheme to encourage selection of a diverse set of *easy* examples. This is further improved in [56] by encouraging more exploration during early phases of learning. More recently, [18] propose a curriculum which computes exponential moving averages of loss values as difficulty scores for training samples.

Motivated by similar ideas, many sample reweighting algorithms are proposed for tackling label noise in neural networks. Many different ways of fixing/learning such weights have been proposed (e.g., [16, 11, 49, 47, 27, 38, 13, 39]) with the general heuristic being that low loss values indicate reliable labels. Algorithms such as Co-Teaching [11] and Co-Teaching+ [49] use two networks and select samples with loss value below a threshold in one network to train the other. In Co-Teaching, the threshold is chosen based on the knowledge of noise rates. The same threshold is used in Co-Teaching+ but the sample selection is based on disagreement between the two networks. [27] also relies on 'small loss' heuristic but the threshold for sample selection is adapted based on the knowledge of label noise rates. MentorNet [16], another recent algorithm based on curriculum learning, uses an auxiliary neural network trained to serve as a sample selection function. Another sample selection algorithm is proposed in [31] where the idea is to train two networks and update the network parameters only in case of a disagreement between the two networks. These sample selection functions are mostly hand-crafted and, hence, they can be sub-optimal. Another strategy is to solve a bilevel optimization problem to find the optimal sample weights (e.g., [13]). The sample selection function used in [11, 49] is sub-optimally chosen for which [47] proposes an AutoML-based approach to find a better function, by fine-tuning on separate data with clean labels. Sample reweighting algorithms proposed in [38] and [39] use online meta-learning and need some extra data with clean labels.

Apart from the sample selection/reweighting approaches described above, there are other approaches to tackling label noise. *Label cleaning* algorithms [41, 48, 40] attempt at identifying and correcting the potentially incorrect labels through joint optimization of sample weights and network weights. *Loss correction* methods [35, 43] suitably modify loss function (or posterior probabilities) to correct for the effects of label noise on risk minimization; however, they need to know (or estimate) the noise rates. There are also theoretical results that investigate robustness of risk minimization [9, 53, 20, 42, 28, 32]. *Regularization* methods, of which sample reweighting approaches are a part, employ explicit or implicit regularization to reduce overfitting to noisy data [1, 24, 30, 51, 37, 33]. More recently, some works have used *self-supervised learning* methods to obtain better initializations for robustness [10, 54], second-order statistics for label cleaning [57] and cluster-based consensus methods [58] to improve noise transition matrix estimations thereby improving loss-correction methods. In this paper, our interest is in the approach of sample selection for achieving robustness to label noise.

The proposed algorithm, BARE, is a simple, adaptive method to select samples which relies only on statistics of loss values (or, equivalently, statistics of class posterior probabilities because we use CCE loss) in a given mini-batch. We do not need any extra data with clean labels or any knowledge about label noise rates. Since it uses batch statistics, the selection thresholds are naturally tied

to the evolving state of learning of the network without needing any tunable hyperparameters. Unlike in many of the aforementioned algorithms, we do not need any auxiliary networks for learning sample selection function, or cross-training, or noise rate estimation and, thus, our algorithm is computationally more efficient.

3. Batch Reweighting Algorithm

In this section we describe the proposed sample reweighting algorithm that relies on mini-batch statistics.

3.1. Problem Formualtion and Notation

Under label noise, the labels provided in the training set may be 'wrong' and we want a classifier whose test error with respect to 'correct' labels is good.

Consider a K-class problem with $\mathcal{X}$ as the feature/pattern space and $\mathcal{Y} = \{0,1\}^K$ as the label space. We assume all labels are one-hot vectors and denote by e_k the one-hot vector corresponding to class k. Let $S^c = \{(x_i, y_i^c),\ i = 1, 2, \cdots, m\}$ be iid samples drawn according to a distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$. We are interested in learning a classifier that does well on a test set drawn according to $\mathcal{D}$. We can do so if we are given S^c as training set. However, what we have is a training set $S = \{(x_i, y_i),\ i = 1, 2, \cdots, m\}$ drawn according to a distribution $\mathcal{D}_\eta$. The y_i here are the 'corrupted' labels and they are related to y_i^c, the 'correct' labels through

$$P[y_i = e_{k'} \mid y_i^c = e_k] = \eta_{kk'} \tag{1}$$

The $\eta_{kk'}$ are called noise rates. (In general the above probability can also depend on the feature vector, x_i, though we do not consider that possibility in this paper). We call this general model as class conditional noise because here the probability of label corruption depends on the original label. A special case of this is the so called symmetric noise where we assume $\eta_{kk} = (1-\eta)$ and $\eta_{kk'} = \frac{\eta}{K-1}, \forall k' \neq k$. Here, η represents the probability of a 'wrong' label. With symmetric noise, the corrupted label is equally likely to be any other label.

We can represent $\eta_{kk'}$ as a matrix and we assume it is diagonally dominant (that is, $\eta_{kk} > \eta_{kk'}, \forall k' \neq k$). (Note that this is true for symmetric noise if $\eta < \frac{K-1}{K}$). Under this condition, if we take all patterns labelled by a specific class in the label-corrupted training set, then patterns that truly belong to that specific class are still in majority in that set. Now the problem of robust learning under label noise can be stated as follows: We want to learn a classifier for the distribution $\mathcal{D}$ but given training data drawn from $\mathcal{D}_\eta$.

We denote by $f(\cdot;\theta)$ a classifier function parameterized by θ. We assume that the neural network classifiers that we use have softmax output layer. Hence, while the training set labels, y_i, are one-hot vectors, we will have $f(x;\theta) \in \Delta^{K-1}$, where $\Delta^{K-1} \subset [0,1]^K$ is the probability simplex. We denote by $\mathcal{L}(f(x;\theta), y)$ the loss function used for the classifier training which in our case is the CCE loss.

3.2. Adaptive Curriculum through Batch Statistics

General curriculum learning can be viewed as minimization of a weighted loss [21, 16]

$$\min_{\theta, \mathbf{w}\in[0,1]^m} \mathcal{L}_{\text{wtd}}(\theta, \mathbf{w}) = \sum_{i=1}^{m} w_i \mathcal{L}(f(x_i;\theta), y_i) + G(\mathbf{w}) + \beta||\theta||^2 \tag{2}$$

where $G(\mathbf{w})$ represents the curriculum. Since one normally employs SGD for learning, we will take m here to be the size of a mini-batch. One simple choice for the curriculum is [21]: $G(\mathbf{w}) = -\lambda||\mathbf{w}||_1,\ \lambda > 0$. Putting this in the above, omitting the regularization term and taking $l_i = \mathcal{L}(f(x_i;\theta), y_i)$, the optimization problem becomes

$$\min_{\theta, \mathbf{w}\in[0,1]^m} \mathcal{L}_{\text{wtd}}(\theta, \mathbf{w}) = \sum_{i=1}^{m} (w_i l_i - \lambda w_i) \tag{3}$$

$$= \sum_{i=1}^{m} (w_i l_i + (1-w_i)\lambda) - m\lambda \tag{4}$$

Under the usual assumption that loss function is non-negative, for the above problem, the optimal $\mathbf{w}$ for any fixed θ is: $w_i = 1$ if $l_i < \lambda$ and $w_i = 0$ otherwise. We first consider a modification where we make λ depend on the class label. The optimization problem becomes

$$\min_{\theta, \mathbf{w}\in[0,1]^m} \mathcal{L}_{\text{wtd}}(\theta, \mathbf{w}) = \sum_{i=1}^{m} (w_i l_i - \lambda(y_i) w_i) \tag{5}$$

$$= \sum_{j=1}^{K} \sum_{\substack{i=1 \\ i:y_i=e_j}}^{m} (w_i l_i + (1-w_i)\lambda_j) - \sum_{j=1}^{K} \sum_{\substack{i=1 \\ i:y_i=e_j}}^{m} \lambda_j \tag{6}$$

where $\lambda_j = \lambda(e_j)$. As is easy to see, the optimal w_i (for any fixed θ) are still given by the same relation: for an i with $y_i = e_j$, $w_i = 1$ when $l_i < \lambda_j$. Note that this relation for optimal w_i is true even if we make λ_j a function of θ and of all x_i with $y_i = e_j$. Thus we can have a truly dynamically adaptive curriculum by making these λ_j depend on all x_i of that class in the mini-batch and the current θ.

The above is an interesting insight: in the Self-Paced Learning formulation [21], the nature of the final solution is same even if we make the λ parameter a function of the class-labels and also other feature vectors corresponding to that class. This gives rise to class-label-dependent thresholds on loss values. To the best of our knowledge, this direction of curriculum learning has not been explored. The next question is how should we decide or evolve these λ_j. As we mentioned earlier, we want these to be determined by the statistics of loss values in the mini-batch.

Consider those i for which $y_i = e_j$. We would be setting $w_i = 1$ and hence use this example to update θ in this minibatch if this $l_i < \lambda_j$. We want λ_j to be fixed based on the observed loss values of this mini-batch. Since there is sufficient empirical evidence that we tend to learn from the clean samples before overfitting to the noisy ones, some quantile or similar statistic of the set of observed loss values in the mini-batch (among patterns labelled with a specific class) would be a good choice for λ_j.

Since we are using CCE loss, we have $l_i = -\ln(f_j(x_i;\theta))$ and as the network has softmax output layer, $f_j(x_i;\theta)$ is the posterior probability of class-j under current θ for x_i. Since the loss and this posterior probability are inversely related, our criterion for selection of an example could be that the assigned posterior probability is above a threshold which is some statistic of the observed posterior probabilities in the mini-batch. In this paper we take the statistic to be mean plus one standard deviation.

In other words, in any mini-batch, we set the weights for samples as

$$w_i = \begin{cases} 1 & \text{if } f_{y_i}(\mathbf{x}_i;\theta) \geq \lambda_{y_i} = \mu_{y_i} + \kappa * \sigma_{y_i} \\ 0 & \text{else} \end{cases} \tag{7}$$

where $\mu_{y_i} = \frac{1}{|\mathcal{S}_{y_i}|}\sum_{s\in\mathcal{S}_{y_i}} f_{y_i}(\mathbf{x}_s;\theta)$ and $\sigma^2_{y_i} = \frac{1}{|\mathcal{S}_{y_i}|}\sum_{s\in\mathcal{S}_{y_i}}(f_{y_i}(\mathbf{x}_s;\theta) - \mu_{y_i})^2$ indicate the sample mean and sample variance of the class posterior probabilities for samples having class label y_i. **[Note:** $\mathcal{S}_{y_i} = \{k \in [m] \mid y_k = y_i\}$ where m is the size of mini-batch**]**. We use $\kappa = 1$ in this paper but we empirically observe that as long as samples from the 'top quantiles' are chosen (i.e. $\kappa > 0$), we get good and similar robustness against label noise across different κ. See Table 19 in Supplementary for an ablation study.

Figures 9–12 (in Supplementary) show that the threshold value (RHS of Equation 7 with $\kappa = 1$) varies across different mini-batches for a given class or epoch. This varying nature of statistics of the loss values in a mini-batch further justifies the rationale for our method of choosing an adaptive threshold.

Algorithm Implementation

Algorithm 1 outlines the proposed method. Keeping in mind that neural networks are trained in a mini-batch manner, Algorithm 1 consists of three parts: i.) computing sample selection thresholds, $\lambda_{y_\mathbf{x}}$, for a given mini-batch of data (Step 8-13), ii.) sample selection based on these thresholds (Steps 15-19) as per Equation 7, and iii.) network parameter updation using these selected samples (Step 20).

Algorithm 1 BAtch REweighting (BARE) Algorithm

1: **Input:** noisy dataset $\mathcal{D}_\eta$, # of classes K, # of epochs T_{max}, learning rate α, mini-batch size $|\mathcal{M}|$
2: **Initialize:** Network parameters, θ_0, for classifier $f(\cdot;\theta)$
3: **for** $t = 0$ **to** $T_{max} - 1$ **do**
4: Shuffle the training dataset $\mathcal{D}_\eta$
5: **for** $i = 1$ to $|\mathcal{D}_\eta|/|\mathcal{M}|$ **do**
6: Draw a mini-batch $\mathcal{M}$ from $\mathcal{D}_\eta$
7: $m = |\mathcal{M}|$ // mini-batch size
8: **for** $p = 1$ to K **do**
9: $\mathcal{S}_p = \{k \in [m] \mid y_k = e_p\}$ // collect indices of samples with class-p
10: $\mu_p = \frac{1}{|\mathcal{S}_p|}\sum_{s\in\mathcal{S}_p} f_p(\mathbf{x}_s;\theta_t)$ // mean posterior prob. for samples with class-p
11: $\sigma_p^2 = \frac{1}{|\mathcal{S}_p|}\sum_{s\in\mathcal{S}_p}(f_p(\mathbf{x}_s;\theta_t) - \mu_p)^2$ // variance in posterior prob. for samples with class-p
12: $\lambda_p \leftarrow \mu_p + \sigma_p$ // sample selection threshold for class-p as per Equation 7
13: **end for**
14: $\mathcal{R} \leftarrow \phi$ // selected samples in $\mathcal{M}$
15: **for each** $\mathbf{x} \in \mathcal{M}$ **do**
16: **if** $f_{y_\mathbf{x}}(\mathbf{x};\theta_t) \geq \lambda_{y_\mathbf{x}}$ **then**
17: $\mathcal{R} \leftarrow \mathcal{R} \cup (\mathbf{x}, y_\mathbf{x})$ // Select sample as per Equation 7
18: **end if**
19: **end for**
20: $\theta_{t+1} = \theta_t - \alpha\nabla\Big(\frac{1}{|\mathcal{R}|}\sum_{(\mathbf{x},y_\mathbf{x})\in\mathcal{R}} \mathcal{L}(\mathbf{x}, y_\mathbf{x};\theta_t)\Big)$ // parameter updates
21: **end for**
22: **end for**
23: **Output:** θ_t

4. Experiments on Noisy Dataset

Dataset: We demonstrate the effectiveness of the proposed algorithm on two benchmark image datasets: MNIST and CIFAR10. These data sets are used to benchmark almost all algorithms for robust learning under label noise and we briefly describe the data sets. MNIST contains 60,000 training images and 10,000 test images (of size 28×28) with 10 classes. CIFAR-10 contains 50,000 training images and 10,000 test images (of size 32×32) with 10 classes. We test the algorithms on two types of label noise: symmetric and class-conditional label noise. In symmetric label noise, each label is randomly flipped to any of the remaining classes with equal probability, whereas for class-conditional noise, label flipping is done in a set of similar classes. For the simulations here, for MNIST, the following flipping is done: $1 \leftarrow 7$, $2 \rightarrow 7$, $3 \rightarrow 8$, and $5 \leftrightarrow 6$. Similarly, for CIFAR10, the following flipping is done: TRUCK $\rightarrow$ AUTOMOBILE, BIRD $\rightarrow$ AIRPLANE, DEER $\rightarrow$ HORSE, CAT $\leftrightarrow$ DOG. We use this type of noise because it is arguably a more realistic scenario and also because it is the type of noise, in addition to symmetric noise, that other algorithms for learning under label noise have used. We also provide results with an arbitrary noise rate matrix (see

Supplementary). For all the datasets, 80% of the training set is used for training and, from the remaining 20% data, we sample 1000 images that constitute the validation set.

We also experiment with the Clothing-1M dataset [46] which is a large-scale dataset obtained by scrapping off the web for different images related to clothing. It contains noise that can be characterized as somewhat close to feature-dependent noise, the most generic kind of label noise. An estimated 40% images have noisy labels. The training dataset contains 1 million images and the number of classes are 14. There are additional training, validation, and test sets of 50k, 14k, and 10k images respectively with clean labels. Since there's a class imbalance, following similar procedure as in existing baselines, we use 260k images from the original noisy training set for training while ensuring equal number of images per class in the set and test set of 10k images for performance evaluation.

Data Augmentations: No data augmentation is used for MNIST. Random croppings with padding of 4, and random horizontal flips are used for CIFAR-10. For Clothing-1M. we do random cropping while ensuring image size is fixed.

Baselines: We compare the proposed algorithm with the following algorithms from literature: 1.) **Co-Teaching (CoT)** [11] which involves cross-training of two similar networks by selecting a fraction (dependent on noise rates) of low loss valued samples; 2.) **Co-Teaching+ (CoT+)** [49] which improves upon CoT with the difference being sample selection only from the subset upon which the two networks' predictions disagree; 3.) **Meta-Ren (MR)** [38], which involves meta-learning of sample weights on-the-fly by comparing gradients for clean and noisy data; 4.) **Meta-Net (MN) [39]**, which improves upon MR by explicitly learning sample weights via a separate neural network; **Curriculum Loss (CL)** [27], which involves a curriculum for sample selection based on (estimated) noise rates; and 6.) **Standard (CCE)** which is the usual training through empirical risk minimization with cross-entropy loss (using the data with noisy labels).

Among these baselines, CoT, CoT+, and CL are sample selection algorithms that require knowledge of noise rates. The algorithms CoT+ and CL need a few initial iterations without any sample selection as a warm-up period; we used 5 epochs and 10 epochs as warm up period during training for MNIST and CIFAR-10 respectively. MR and MN assume access to a small set of clean validation data. Because of this, and for a fair comparison among all the baselines, a clean validation set of 1000 samples is used in case of MR and MN, and the same set of samples but with the noisy labels is used for the rest of the algorithms including the proposed one.

Network architectures & Optimizers: While most algorithms for learning under label noise use MNIST and CIFAR10 data, different algorithms use different network architectures. Hence, for a fairer comparison, we have decided to use small networks that give state of art performance on clean data and investigate the robustness we get by using our algorithm on these networks. Please refer the supplementary material for details about the network architectures and optimization routines.

Performance Metrics: For all algorithms we compare **test accuracies** on a separate test set with clean labels. The main idea in all sample selection schemes is to identify noisy labels. Hence, in addition to test accuracies, we also compare **precision** ($\#$ *clean labels selected* / $\#$ *of selected labels*) and **recall** ($\#$ *clean labels selected* / $\#$ *of clean labels in the data*) in identifying noisy labels.

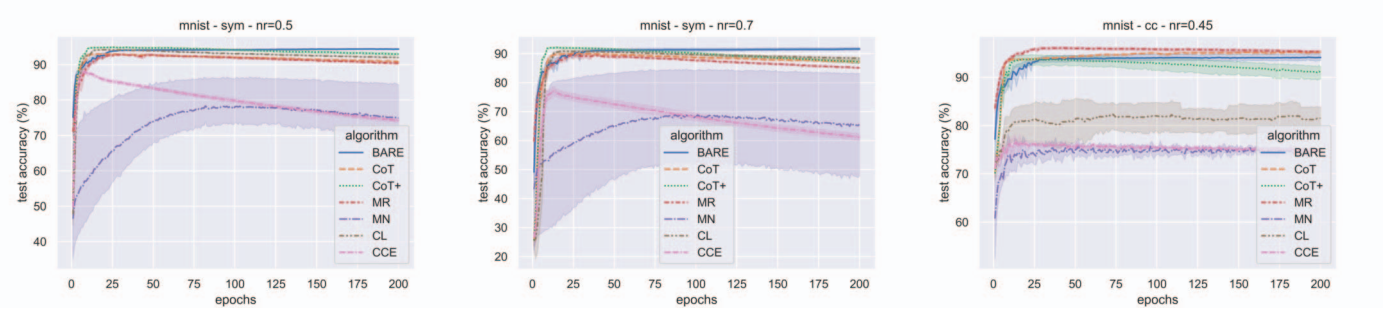
4.1. Discussion of Results

Performance on MNIST. Figure 1 shows the evolution of test accuracy (with training epochs) under symmetric ($\eta \in \{0.5, 0.7\}$) and class conditional ($\eta = 0.45$) label noise for different algorithms. We can see from the figure that the proposed algorithm outperforms the baselines for symmetric noise. For the case of class-conditional noise, the test accuracy of the proposed algorithm is marginally less than the best of the baselines, namely CoT and MR.

Performance on CIFAR-10. Figure 2 shows the test accuracies of the various algorithms as the training progresses for both symmetric ($\eta \in \{0.3, 0.7\}$) and class-conditional ($\eta = 0.4$) label noise. We can see from the figure that the proposed algorithm outperforms the baseline schemes and its test accuracies are uniformly good for all types of label noise. It is to be noted that while test accuracies for our algorithm stay saturated after attaining maximum performance, the other algorithms' performance seems to deteriorate as can be seen in the form of accuracy dips towards the end of training. This suggests that our proposed algorithm doesn't let the network overfit even after long durations of training unlike the case with other algorithms.

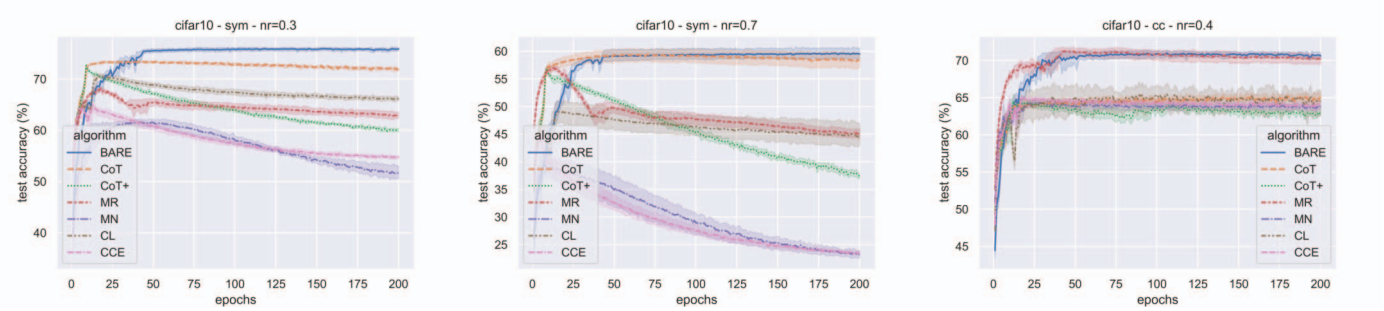
All the algorithms, except the proposed one, have hyperparameters (in the sample selection/weighting method) and the accuracies reported here are for the best possible hyperparameter values obtained through tuning. The MR and MN algorithms are particularly sensitive to hyperparameter values in the meta learning algorithm. In contrast, BARE has no hyperparameters for the sample selection and hence no such tuning is involved. It may be noted for the test accuracies on MNIST and CIFAR-10 that sometimes the standard deviation in the accuracy for MN is high. As we mentioned earlier, we noticed that MN is very sensitive to the tuning of hyper parameters. While we tried our best to tune all the hyper parameters, may be the final ones we found for these cases are still not the best and that is why the standard deviation is high.

Performance on Clothing1M. On this dataset, BARE achieved a test accuracy of 72.28% against the accuracy



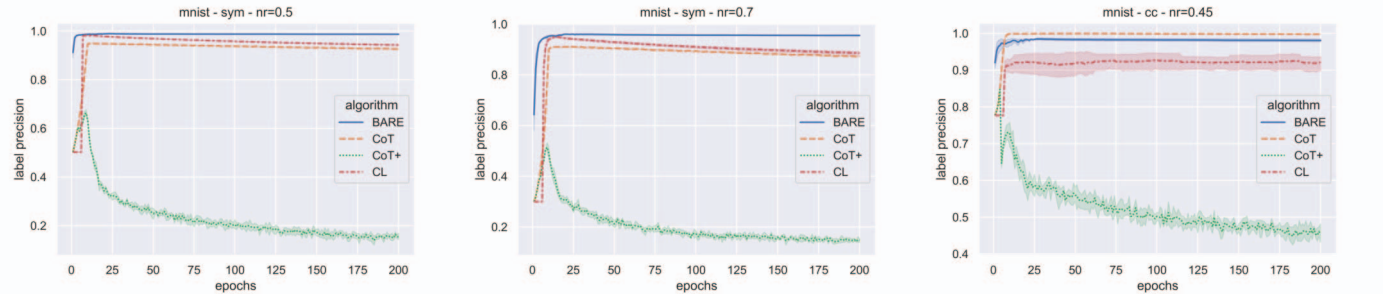
(a) (b) (c)

Figure 1: Test Accuracies - MNIST - Symmetric (a & b) & Class-conditional (c) Label Noise



(a) (b) (c)

Figure 2: Test Accuracies - CIFAR10 - Symmetric (a & b) & Class-conditional (c) Label Noise

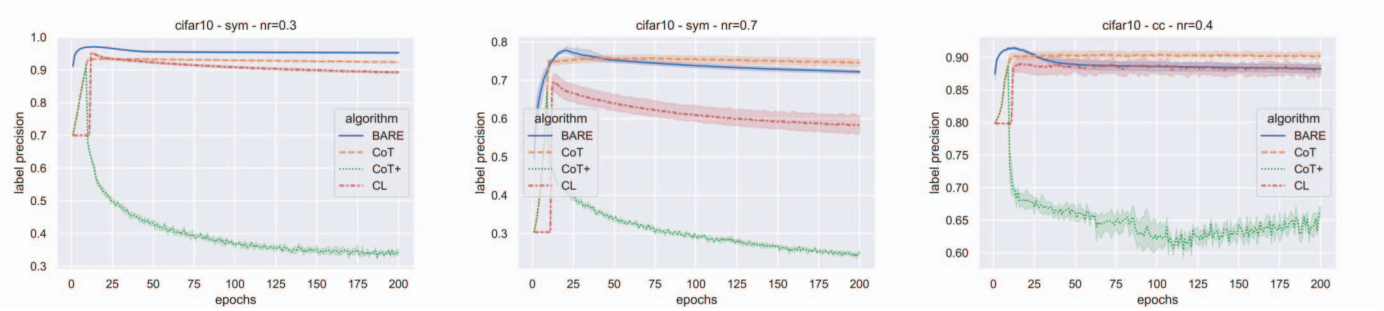


(a) (b) (c)

Figure 3: Label Precision - MNIST - Symmetric (a & b) & Class-conditional (c) Label Noise

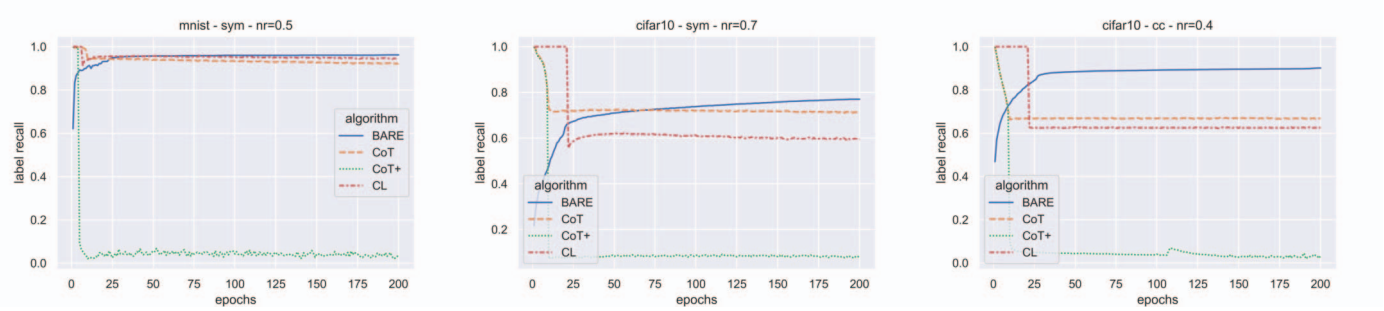
of 68.8% achieved by CCE. The accuracy achieved by BARE is better than that reported in the corresponding papers for all other baselines except for C2D [54] & DivideMix [24] which reported accuracy of 74.58% & 74.76% resp. (The results are summarized in Table 3 in the Supplementary). These results show that even for datasets used in practice which have feature-dependent label noise, BARE performs better than all but two baselines. We note that the best performing baseline, DivideMix, requires about 2.4 times the computation time required for BARE. In addition to this, DivideMix requires tuning of 5 hyperparameters whereas no such tuning is required for BARE. The second best performing baseline, C2D, is also computationally expensive than BARE as it relies on self-supervised learning.

Efficacy of detecting clean samples. Figure 3 and Figure 4 show the label precision (across epochs) of the various algorithms on MNIST and CIFAR-10 respectively. One can see from these figures that BARE has comparable or better precision. Thus, compared to other sample selection algorithms, a somewhat higher fraction of examples selected for training by BARE have clean labels.



(a) (b) (c)

Figure 4: Label Precision - CIFAR10 - Symmetric (a & b) & Class-conditional (c) Label Noise



(a) (b) (c)

Figure 5: Label Recall - Symmetric (a & b) & Class-conditional (c) Label Noise

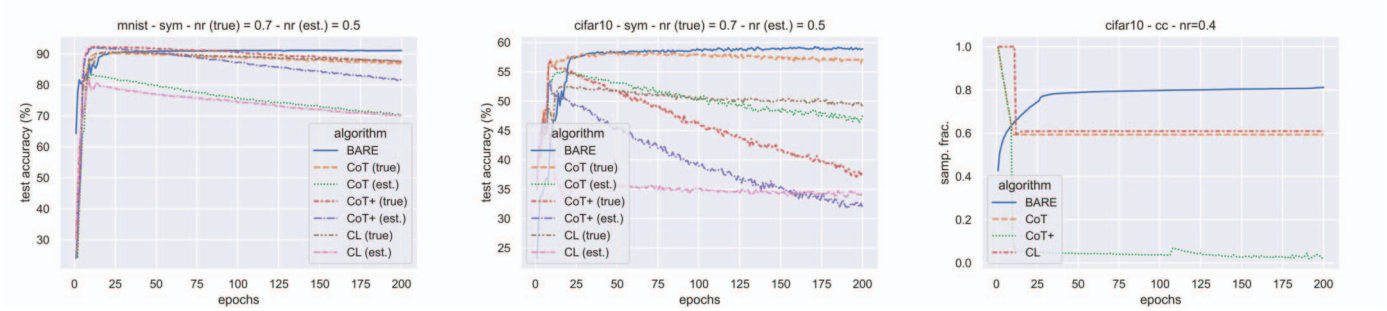
While *test accuracies* and *label precision* values do demonstrate the effectiveness of algorithms, it's also instructive to look at the label recall values. Label recall tells us how a sample selection algorithm performs when it comes to selecting reliable, clean samples. Figure 5 shows the label recall values for CoT, CoT+, CL, and BARE for MNIST (5a) and CIFAR-10 (5b & 5c). It can be noted that BARE consistently achieves better recall values compared to the baselines. Higher recall values indicate that the algorithm is able to identify clean samples more reliably. This is useful, for example, to employ a label cleaning algorithm on the samples flagged as noisy (i.e., not selected) by BARE. CoT+ selects a fraction of samples where two networks disagree and, hence, after the first few epochs, it selects very few samples ($\sim$ 3000) in each epoch. Since these are samples in which the networks disagree, a good fraction of them may have noisy labels. This may be the reason for the poor precision and recall values of CoT+ as seen in these figures.

This can be seen from Figure 6c as well which shows the fraction of samples chosen by the sample selection algorithms as epochs go by for $\eta = 0.4$ (class-conditional noise) on CIFAR-10 dataset. It can be noted that, as noise rate is to be supplied to CoT and CL, they select $1-\eta = 0.6$ fraction of data with every epoch. Whereas, in case of CoT+, the samples where the networks disagree is small because of the training dynamics and as a result, after a few epochs, it consistently selects very few samples. Since the noise is class-conditional, even though $\eta = 0.4$, the actual amount of label flipping is $\sim 20\%$. And this is why it's interesting to note that BARE leads to an approximate sample selection ratio of 80%. (We provide similar plots for different noise rates and datasets in the supplementary.)

Table 1: Algorithm run times for training (in seconds)

ALGORITHM	MNIST	CIFAR10
BARE	**310.64**	**930.78**
CoT	504.5	1687.9
CoT+	537.7	1790.57
MR	807.4	8130.87
MN	1138.4	8891.6
CL	730.15	1254.3
CCE	**229.27**	**825.68**

Efficiency of BARE. Table 1 shows the typical run times for 200 epochs of training with all the algorithms. It can be seen from the table that the proposed algorithm takes roughly the same time as the usual training with CCE loss



(a) (b) (c)

Figure 6: (a & b): Test accuracies when estimated (symmetric) noise rate, $\eta = 0.5$, and true noise rate, $\eta = 0.7$, for MNIST & CIFAR-10 resp.; (c): sample fraction values for $\eta = 0.4$ (class-conditional noise) on CIFAR-10

whereas all other baselines are significantly more expensive computationally. In case of MR and MN, the run times are around 8 times that of BARE for CIFAR-10.

Sensitivity to noise rates. Some of the baselines schemes such as CoT, CoT+, and CL require knowledge of true noise rates beforehand. (In fact, in the simulations shown so far, we have used the actual noise rate for these baselines). This information is typically unavailable in practice. One can estimate the noise rates but there would be inevitable errors in estimation. Figure 6 shows the effect of mis-specification of noise rates for these 3 baselines schemes. As can be seen from these figures, while the algorithms can exhibit robust learning when the true noise rate is known, the performance deteriorates if the estimated noise rate is erroneous. BARE does not have this issue because it does not need any information on noise rate.

Sensitivity to batch size. To show the insensitivity to batch size, we show in Table 2 results on MNIST & CIFAR-10 for both types of label noise and three batch sizes: 64, 128 (used in paper), and 256.

5. Conclusions

We proposed an adaptive sample selection scheme, BARE, for robust learning under label noise. The algorithm relies on statistics of scores (posterior probabilities) of all samples in a minibatch to select samples from that minibatch. The current algorithms for sample selection in literature rely on heuristics such as cross-training multiple networks or meta-learning of sample weights which is often computationally expensive. They may also need knowledge of noise rates or some data with clean labels which may not be easily available. In contrast, BARE neither needs an extra data set with clean labels nor does it need any knowledge of the noise rates, nor does it need to learn multiple networks. Furthermore, it has no hyperparameters in the selection algorithm. Comparisons with baseline schemes on benchmark datasets show the effectiveness of the proposed algorithm both in terms of performance metrics and computational complexity. In addition, performance figures in terms of precision and recall show that BARE is very reliable in selecting clean samples. This, combined with the fact that there are no additional hyperparameters to tune, shows the advantage that BARE can offer for robust learning under label noise.

Table 2: Test Accuracy (%) of BARE on MNIST & CIFAR-10 with batch sizes $\in \{64, 128, 256\}$

DATASET	NOISE (η)	BATCH SIZE	TEST ACCURACY
MNIST	50% (SYM.)	64	95.31 ± 0.16
		128	94.38 ± 0.13
		256	94.44 ± 0.48
MNIST	45% (CC)	64	93.31 ± 0.63
		128	94.11 ± 0.77
		256	94.68 ± 0.63
CIFAR-10	30% (SYM.)	64	76.77 ± 0.38
		128	75.85 ± 0.41
		256	74.56 ± 0.53
CIFAR-10	40% (CC)	64	71.87 ± 0.28
		128	70.63 ± 0.46
		256	69.03 ± 0.35

The mini-batch statistics used in BARE are class-specific. Hence, one may wonder whether such statistics would be reliable when the number of classes is large and hence is comparable to the mini-batch size. Our preliminary investigations show that the method delivers good performance even on a 101-class dataset with a minibatch size of 128 (see Table 18 in Supplementary material). A possible approach for tackling large number of classes would be to make mini-batches in such a way that any given mini-batch contains examples of only a few of the classes (though for a full epoch there would be no class-imbalance). More investigations are needed to study this aspect of BARE.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning*, pages 312–321. PMLR, 2019.

[2] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning*, pages 312–321. PMLR, 2019.

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

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[5] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014.

[6] Nontawat Charoenphakdee, Jongyeong Lee, and Masashi Sugiyama. On symmetric losses for learning from corrupted labels. In *International Conference on Machine Learning*, pages 961–970. PMLR, 2019.

[7] Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng. Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise. *arXiv preprint arXiv:2012.05458*, 2020.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[9] Aritra Ghosh, Himanshu Kumar, and PS Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 1919–1925, 2017.

[10] Aritra Ghosh and Andrew Lan. Contrastive learning improves model robustness under label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2703–2708, June 2021.

[11] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pages 8527–8537, 2018.

[12] Charles R Harris, K Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020.

[13] Simon Jenni and Paolo Favaro. Deep bilevel learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 618–633, 2018.

[14] Lu Jiang, Deyu Meng, Shoou-I Yu, Zhenzhong Lan, Shiguang Shan, and Alexander Hauptmann. Self-paced learning with diversity. In *Advances in Neural Information Processing Systems*, pages 2078–2086, 2014.

[15] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander Hauptmann. Self-paced curriculum learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

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

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Yajing Kong, Liu Liu, Jun Wang, and Dacheng Tao. Adaptive curriculum learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5067–5076, 2021.

[19] Alex Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. PhD thesis, University of Toronto, 2009.

[20] H. Kumar and P. S. Sastry. Robust loss functions for learning multi-class classifiers. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 687–692, 2018.

[21] M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems-Volume 1*, pages 1189–1197, 2010.

[22] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[23] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5447–5456, 2018.

[24] Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020.

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

[26] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342, 2020.

[27] Yueming Lyu and Ivor W. Tsang. Curriculum loss: Robust learning and generalization against label corruption. In *International Conference on Learning Representations*, 2020.

[28] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*, pages 6543–6553. PMLR, 2020.

[29] Xingjun Ma, Yisen Wang, Michael E. Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3355–3364, 2018.

[30] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pages 3355–3364. PMLR, 2018.

[31] Eran Malach and Shai Shalev-Shwartz. Decoupling" when to update" from" how to update". In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 961–971, 2017.

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

[33] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alche-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[35] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[37] Scott E Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *ICLR (Workshop)*, 2015.

[38] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4334–4343, 2018.

[39] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, pages 1919–1930, 2019.

[40] Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *ICML*, pages 5907–5915, 2019.

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

[42] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330, 2019.

[43] Zhen Wang, Guosheng Hu, and Qinghua Hu. Training noise-robust deep neural networks via meta-learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4524–4533, 2020.

[44] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. 2020.

[45] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 33, 2020.

[46] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015.

[47] Quanming Yao, Hansi Yang, Bo Han, Gang Niu, and James Tin-Yau Kwok. Searching to exploit memorization effect in learning with noisy labels. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10789–10798. PMLR, 2020.

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

[49] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *Proceedings of the 36th International Conference on Machine Learning*, pages 7164–7173, 2019.

[50] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning

requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[51] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[52] Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. Learning with feature-dependent label noise: A progressive approach. In *International Conference on Learning Representations*, 2020.

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

[54] Evgenii Zheltonozhskii, Chaim Baskin, Avi Mendelson, Alex M Bronstein, and Or Litany. Contrast to divide: Self-supervised pre-training for learning with noisy labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1657–1667, 2022.

[55] Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen. Error-bounded correction of noisy labels. In *International Conference on Machine Learning*, pages 11447–11457. PMLR, 2020.

[56] Tianyi Zhou and Jeff Bilmes. Minimax curriculum learning: Machine teaching with desirable difficulties and scheduled diversity. In *International Conference on Learning Representations*, 2018.

[57] Zhaowei Zhu, Tongliang Liu, and Yang Liu. A second-order approach to learning with instance-dependent label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10113–10123, 2021.

[58] Zhaowei Zhu, Yiwen Song, and Yang Liu. Clusterability as an alternative to anchor points when learning with noisy labels. In *International Conference on Machine Learning*, pages 12912–12923. PMLR, 2021.