Unsupervised Label Noise Modeling and Loss Correction

Eric Arazo [* 1] Diego Ortego [* 1] Paul Albert [1] Noel E. O'Connor [1] Kevin McGuinness [1]

Abstract

Despite being robust to small amounts of label noise, convolutional neural networks trained with stochastic gradient methods have been shown to easily fit random labels. When there are a mixture of correct and mislabelled targets, networks tend to fit the former before the latter. This suggests using a suitable two-component mixture model as an unsupervised generative model of sample loss values during training to allow online estimation of the probability that a sample is mislabelled. Specifically, we propose a beta mixture to estimate this probability and correct the loss by relying on the network prediction (the so-called bootstrapping loss). We further adapt *mixup* augmentation to drive our approach a step further. Experiments on CIFAR-10/100 and TinyImageNet demonstrate a robustness to label noise that substantially outperforms recent state-of-the-art. Source code is available at https://git.io/fjsvE and Appendix at https://arxiv.org/abs/1904.11238.

1. Introduction

Convolutional Neural Networks (CNNs) have recently become the par excellence base approach to deal with many computer vision tasks (DeTone et al., 2016; Ono et al., 2018; Beluch et al., 2018; Redmon et al., 2016; Zhao et al., 2017; Krishna et al., 2017). Their widespread use is attributable to their capability to model complex patterns (Ren et al., 2018) when vast amounts of labeled data are available. Obtaining such volumes of data, however, is not trivial and usually involves an error prone automatic or a manual labeling process (Wang et al., 2018a; Zlateski et al., 2018). These errors lead to *noisy samples*: samples annotated with incorrect or *noisy labels*. As a result, dealing with label
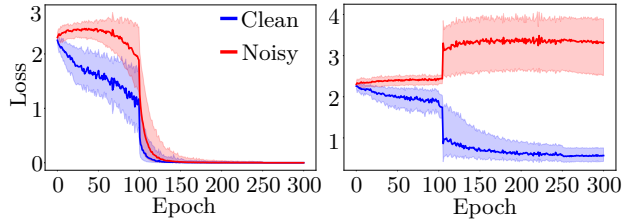
*Equal contribution [1]Insight Centre for Data Analytics, Dublin City University (DCU), Dublin, Ireland. Correspondence to: Eric Arazo <eric.arazo@insight-centre.org>, Diego Ortego <diego.ortego@insight-centre.org>.

Figure 1. Cross-entropy loss on CIFAR-10 under 80% label noise for clean and noisy samples. Left: training with cross-entropy loss results in fitting the noisy labels. Right: using our proposed objective prevents fitting label noise while also learning from the noisy samples. The heavy lines represent the median losses and the shaded areas are the interquartile ranges.

noise is a common adverse scenario that requires attention to ensure useful visual representations can be learnt (Jiang et al., 2018b; Wang et al., 2018a; Wu et al., 2018; Jiang et al., 2018a; Zlateski et al., 2018). Automatically obtained noisy labels have previously been demonstrated useful for learning visual representations (Pathak et al., 2017; Gidaris et al., 2018); however, a recent study on the generalization capabilities of deep networks (Zhang et al., 2017) demonstrates that noisy labels are easily fit by CNNs, harming generalization. This overfitting also arises in biases that networks encounter during training, e.g., when a dataset contains class imbalances (Alvi et al., 2018). However, before fitting label noise, CNNs fit the correctly labeled samples (*clean samples)* even under high-levels of corruption (Figure 1, left).

Existing literature on training with noisy labels focuses primarily on loss correction approaches (Reed et al., 2015; Hendrycks et al., 2018; Jiang et al., 2018b). A well-known approach is the bootstrapping loss (Reed et al., 2015), which introduces a perceptual consistency term in the learning objective that assigns a weight to the current network prediction to compensate for the erroneous guiding of noisy samples. Other approaches modify class probabilities (Patrini et al., 2017; Hendrycks et al., 2018) by estimating the noise associated with each class, thus computing a loss that guides the training process towards the correct classes. Still other approaches use curriculum learning to formulate a robust learning procedure (Jiang et al., 2018b; Ren et al., 2018). Curriculum learning (Bengio et al., 2009) is based on the idea that ordering training examples in a meaningful (e.g.

easy to hard) sequence might improve convergence and generalization. In the noisy label scenario, easy (hard) concepts are associated with clean (noisy) samples by re-weighting the loss for noisy samples so that they contribute less. Discarding noisy samples, however, potentially removes useful information about the data distribution. (Wang et al., 2018b) overcome this problem by introducing a similarity learning strategy that pulls representations of noisy samples away from clean ones. Finally, *mixup* data augmentation (Zhang et al., 2018) has recently demonstrated outstanding robustness against label noise without explicitly modeling it.

In light of these recent advances, this paper proposes a robust training procedure that avoids fitting noisy labels even under high levels of corruption (Figure 1, right), while using noisy samples for learning visual representations that achieve a high classification accuracy. Contrary to most successful recent approaches that assume the existence of a known set of clean data (Ren et al., 2018; Hendrycks et al., 2018), we propose an unsupervised model of label noise based exclusively on the loss on each sample. We argue that clean and noisy samples can be modeled by fitting a two-component (clean-noisy) beta mixture model (BMM) on the loss values. The posterior probabilities under the model are then used to implement a dynamically weighted bootstrapping loss, robustly dealing with noisy samples without discarding them. We provide experimental work demonstrating the strengths of our approach, which lead us to substantially outperform the related work. Our main contributions are as follows:

1. A simple yet effective unsupervised noise label modeling based on each sample loss.

2. A loss correction approach that exploits the unsupervised label noise model to correct each sample loss, thus preventing overfitting to label noise.

3. Pushing the state-of-the-art one step forward by combining our approach with mixup data augmentation (Zhang et al., 2018).

4. Guiding mixup data augmentation to achieve convergence even under extreme label noise.

2. Related work

Recent efforts to deal with label noise address two scenarios (Wang et al., 2018b): closed-set and open-set label noise. In the closed set scenario, the set of possible labels S is known and fixed. All samples, including noisy ones, have their true label in this set. In the open set scenario, the true label of a noisy sample x_i may be outside S; i.e. x_i may be an out-of-distribution sample (Liang et al., 2018). The remainder of this section briefly reviews related work in the closed-set scenario considered in (Zhang et al., 2017), upon which we base our approach.

Several types of noise can be studied in the closed-set scenario, namely *uniform* or *non-uniform* random label noise. The former is also known as symmetric label noise and implies ground-truth labels flipped to a different class with uniform random probability. Non-uniform or class-conditional label noise, on the other hand, has different flipping probabilities for each class (Hendrycks et al., 2018). Previous research (Patrini et al., 2017) suggests that uniform label noise is more challenging than non-uniform.

A simple approach to dealing with label noise is to remove the corrupted data. This is not only challenging because difficult samples may be confused with noisy ones (Wang et al., 2018b), but also implies not exploiting the noisy samples for representation learning. It has, however, recently been demonstrated (Ding et al., 2018) that it is useful to discard samples with a high probability of being incorrectly labeled and still use these samples in a semi-supervised setup.

Other approaches seek to relabel the noisy samples by modeling their noise through directed graphical models (Xiao et al., 2015), Conditional Random Fields (Vahdat, 2017), or CNNs (Veit et al., 2017). Unfortunately, to predict the true label, these approaches rely on the assumption that a small set of clean samples is always available, which limits their applicability. Tanaka et al. (Tanaka et al., 2018) have, however, recently demonstrated that it is possible to do unsupervised sample relabeling using the network predictions to predict hard or soft labels.

Loss correction approaches (Reed et al., 2015; Jiang et al., 2018b; Patrini et al., 2017; Zhang et al., 2018) modify either the loss directly, or the probabilities used to compute it, to compensate for the incorrect guidance provided by the noisy samples. (Reed et al., 2015) extend the loss with a perceptual term that introduces a certain reliance on the model prediction. Their approach is, however, limited in that the noise label always affects the objective. (Patrini et al., 2017) propose a backward method that weights the loss of each sample using the inverse of a noise transition matrix T, which specifies the probability of one label being flipped to another. (Patrini et al., 2017) presents a forward method that, instead of operating directly on the loss, goes back to the predicted probabilities to correct them by multiplying by the T matrix. (Hendrycks et al., 2018) corrects the predicted probabilities using a corruption matrix computed using a model trained on a clean set of samples and their prediction on the corrupted data. Other approaches focus on re-weighting the contribution of noisy samples on the loss. (Jiang et al., 2018b) proposes an alternating minimization framework in which a mentor network learns a curriculum (i.e. a weight for each sample) to guide a student network that learns under label noise conditions. Similarly, (Guo

et al., 2018) present a curriculum learning approach based on an unsupervised estimation on data complexity through its distribution in a feature space that benefits from training with both clean and noisy samples. (Ren et al., 2018) weights each sample in the loss based on the gradient directions in training compared to those on validation (i.e. in a clean set). Note that, as for relabeling approaches, the assumption of clean data availability limits the application of many of these approaches. Conversely, approaches like (Wang et al., 2018b) do not rely on clean data by performing unsupervised noise label detection to help re-weighting the loss, while not discarding noisy samples that are exploited in a similarity learning framework to pull their representations away from true samples of each class.

In contrast to the aforementioned literature, we propose to deal with noisy labels using exclusively the training loss of each sample without consulting any clean set. Specifically, we fit a two-component beta mixture model to the training loss of each sample to model clean and noisy samples. We use this unsupervised model to implement a loss correction approach that benefits both from bootstrapping (Reed et al., 2015) and mixup data augmentation (Zhang et al., 2018) to deal with the closed-set label noise scenario.

3. Learning with label noise

Image classification can be formulated as the problem of learning a model $h_\theta(x)$ from a set of training examples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ with $y_i \in \{0,1\}^C$ being the one-hot encoding ground-truth label corresponding to x_i. In our case, h_θ is a CNN and θ represents the model parameters (weights and biases). As we are considering classification under label noise, the label y_i can be noisy (i.e. x_i is a noisy sample). The parameters θ are fit by optimizing a loss function, e.g. categorical cross-entropy:

$$\ell(\theta) = \sum_{i=1}^N \ell_i(\theta) = -\sum_{i=1}^N y_i^T \log\left(h_\theta(x_i)\right), \quad (1)$$

where $h_\theta(x)$ are the softmax probabilities produced by the model and $\log(\cdot)$ is applied elementwise. The remainder of this section describes our noisy sample modeling technique and how to extend the loss in Eq. (1) based on this model to handle label noise. For notational simplicity, we use $\ell_i(\theta) = \ell_i$ and $h_\theta(x_i) = h_i$ in the remainder of the paper.

3.1. Label noise modeling

We aim to identify the noisy samples in the dataset $\mathcal{D}$ so that we can implement a loss correction approach (see Subsections 3.2 and 3.3). Our essential observation is simple: random labels take longer to learn than clean labels, meaning that noisy samples have higher loss during the early epochs of training (see Figure 1), allowing clean and noisy
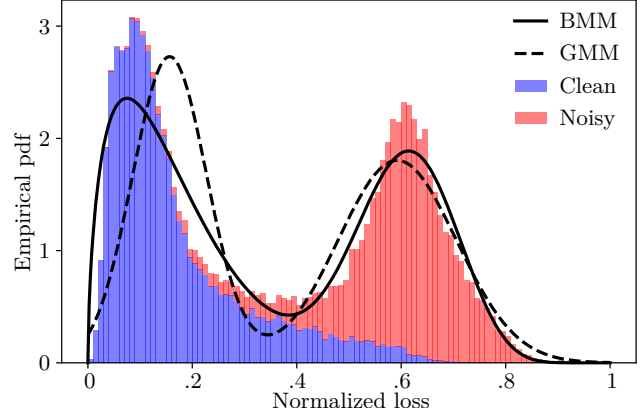

Figure 2. Empirical PDF and estimated GMM and BMM models for 50% label noise in CIFAR-10 after 10 epochs with standard cross-entropy loss and learning rate of 0.1 (remaining hyperparameters see in Subsection 4.1). Clean and noisy samples are colored for illustrative purposes. The BMM model better fits the skew toward zero loss of the noisy samples.

samples to be distinguished from the loss distribution alone (see Figure 2). Modern CNNs trained with stochastic gradient methods typically do not fit the noisy examples until substantial progress has been made in fitting the clean ones. Therefore, one can infer from the loss value if a sample is more likely to be clean or noisy. We propose to use a mixture distribution model for this purpose.

Mixture models are a widely used unsupervised modeling technique (Stauffer & Grimson, 1999; Permuter et al., 2006; Ma & Leijon, 2011), with the Gaussian Mixture Model (GMM) (Permuter et al., 2006) being the most popular. The probability density function (pdf) of a mixture model of K components on the loss ℓ is defined as:

$$p(\ell) = \sum_{k=1}^K \lambda_k \, p(\ell \mid k), \quad (2)$$

where λ_k are the mixing coefficients for the convex combination of each individual pdf $p(\ell \mid k)$. In our case, we can fit a two components GMM (i.e. $K = 2$ and $\ell \sim \mathcal{N}(\mu_k, \sum_k)$) to model the distribution of clean and noisy samples (Figure 2). Unfortunately, the Gaussian is a poor approximation to the clean set distribution, which exhibits high skew toward zero. The more flexible beta distribution (Ma & Leijon, 2011) allows modelling both symmetric and skewed distributions over $[0, 1]$; the beta mixture model (BMM) better approximates the loss distribution for mixtures of clean and noisy samples (Figure 2). Empirically, we also found the BMM improves ROC-AUC for clean-noisy label classification over the GMM by around 5 points for 80% label noise in CIFAR-10 when using the training objective in Section 3.3 (see Appendix A). The beta distribution over a (max) normalized loss $\ell \in [0, 1]$ is defined to have pdf:

$$p(\ell \mid \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\,\Gamma(\beta)} \ell^{\alpha-1} \left(1 - \ell\right)^{\beta-1}, \qquad (3)$$

where $\alpha, \beta > 0$ and $\Gamma(\cdot)$ is the Gamma function, and the mixture pdf is given by substituting the above into Eq. (2).

We use an Expectation Maximization (EM) procedure to fit the BMM to the observations. Specifically, we introduce latent variables $\gamma_k(\ell) = p(k|\ell)$ which are defined to be the posterior probability of the point ℓ having been generated by mixture component k. In the E-step we fix the parameters $\lambda_k, \alpha_k, \beta_k$ and update the latent variables using Bayes rule:

$$\gamma_k(\ell) = \frac{\lambda_k\, p(\ell \mid \alpha_k, \beta_k)}{\sum_{j=1}^{K} \lambda_j\, p(\ell \mid \alpha_j, \beta_j)}. \qquad (4)$$

Given fixed $\gamma_k(\ell)$, the M-step estimates the distribution parameters α_k, β_k using a weighted version of the method of moments:

$$\beta_k = \frac{\alpha_k\left(1 - \bar{\ell}_k\right)}{\bar{\ell}_k}, \;\; \alpha_k = \bar{\ell}_k \left(\frac{\bar{\ell}_k\left(1 - \bar{\ell}_k\right)}{s_k^2} - 1\right) \qquad (5)$$

with $\bar{\ell}_k$ being a weighted average of the losses $\{\ell_i\}_{i=1}^{N}$ corresponding to each training sample $\{x_i\}_{i=1}^{N}$, and s_k^2 being a weighted variance estimate:

$$\bar{\ell}_k = \frac{\sum_{i=1}^{N} \gamma_k(\ell_i)\,\ell_i}{\sum_{i=1}^{N} \gamma_k(\ell_i)}, \qquad (6)$$

$$s_k^2 = \frac{\sum_{i=1}^{N} \gamma_k(\ell_i)\left(\ell_i - \bar{\ell}_k\right)^2}{\sum_{i=1}^{N} \gamma_k(\ell_i)}. \qquad (7)$$

The updated mixing coefficients λ_k are then calculated in the usual way:

$$\lambda_k = \frac{1}{N} \sum_{i=1}^{N} \gamma_k(\ell_i). \qquad (8)$$

The above E and M-steps are then iterated until convergence or a maximum number of iterations (10 in our experiments) are reached. Note that the above algorithm becomes numerically unstable when the observations are very near zero and one. Our implementation simply sidesteps this issue by bounding the observations in $[\epsilon, 1 - \epsilon]$ instead of $[0, 1]$ ($\epsilon = 10^{-4}$ in our experiments).

Finally, we obtain the probability of a sample being clean or noisy through the posterior probability:

$$p(k \mid \ell_i) = \frac{p(k)\,p(\ell_i \mid k)}{p(\ell_i)}, \qquad (9)$$

where $k = 0\,(1)$ denotes clean (noisy) classes.

Note that the loss used to estimate the mixture distribution is always the standard cross-entropy loss (Figure 1) for all samples after every epoch. This not necessarily the loss used for training, which may contain a corrective component to deal with label noise.

3.2. Noise model for label correction

Carefully selecting a loss function to guide the learning process is of particular importance under label noise. Standard categorical cross-entropy loss (Eq. (1)) is ill-suited to the task as it encourages fitting label noise (Zhang et al., 2017). The static hard bootstrapping loss proposed in (Reed et al., 2015) provides a mechanism to deal with label noise by adding a perceptual term to the standard cross-entropy loss that helps to correct the training objective:

$$\ell_B = -\sum_{i=1}^{N} \left((1 - w_i)\,y_i + w_i z_i\right)^T \log\left(h_i\right), \qquad (10)$$

where w_i weights the model prediction z_i in the loss function. (Reed et al., 2015) use $w_i = 0.2, \forall i$. We refer to this approach as static hard bootstrapping. (Reed et al., 2015) also proposed a static soft bootstrapping loss ($w_i = 0.05, \forall i$) that uses the predicted softmax probabilities h_i instead of the class prediction z_i. Unfortunately, using a fixed weight for all samples does not prevent fitting the noisy ones (Table 1 in Subsection 4.2) and, more importantly, applying a small fixed weight w_i to the prediction (probabilities) z_i (h_i) limits the correction of a hypothetical noisy label y_i.

We propose dynamic hard and soft bootstrapping losses by using our noise model to individually weight each sample; i.e., w_i is dynamically set to $p(k = 1 \mid \ell_i)$ and the BMM model is estimated after each training epoch using the cross-entropy loss for each sample ℓ_i. Therefore, clean samples rely on their ground-truth label y_i ($1 - w_i$ is large), while noisy ones let their loss being dominated by their class prediction z_i or their predicted probabilities h_i (w_i is large), respectively, for hard and soft alternatives. Note that in mature stages of training the CNN model should provide a good estimation of the true class for noisy samples. Subsection 4.2 compares static and dynamic bootstrapping, showing that dynamic bootstrapping gives superior results.

3.3. Joint label correction and *mixup* data augmentation

Recently (Zhang et al., 2018) proposed a data augmentation technique named *mixup* that exhibits strong robustness to label noise. This technique trains on convex combinations of sample pairs (x_p and x_q) and corresponding labels (y_p and y_q):

$$x = \delta x_p + (1 - \delta)x_q, \qquad (11)$$

$$\ell = \delta \ell_p + (1 - \delta)\ell_q, \qquad (12)$$

where δ is randomly sampled from a beta distribution $\mathcal{B}e\left(\alpha, \beta\right)$, with $\alpha = \beta$ set to high values when learning with label noise so that δ tends to be close to 0.5. This combination regularizes the network to favor simple linear behavior between training samples, which reduces oscillations in regions far from them. Regarding label noise, *mixup*

provides a mechanism to combine clean and noisy samples, computing a more representative loss to guide the training process. Even when combining two noisy samples the loss computed can still be useful as one of the noisy samples may (by chance) contain the true label of the other one. As for preventing overfitting to noisy samples, the fact that samples and their labels are mixed favors learning structured data, while hindering learning the unstructured noise.

Mixup achieves robustness to label noise by appropriate combinations of training examples. Under high-levels of noise mixing samples that both have incorrect labels is prevalent, which reduces the effectiveness of the method. We propose to fuse *mixup* and our dynamic bootstrapping to implement a robust per-sample loss correction approach:

$$\ell^* = -\delta \left[\left((1 - w_p)\, y_p + w_p z_p \right)^T \log\left(h\right) \right] +$$

$$(1 - \delta) \left[\left((1 - w_q)\, y_q + w_q z_q \right)^T \log\left(h\right) \right], \quad (13)$$

The loss ℓ^* defines the hard alternative, while the soft one can be easily defined by replacing z_p and z_q by h_p and h_q. These hard and soft losses exploit mixup's advantages while correcting the labels through dynamic bootstrapping, i.e. the weights w_p and w_q that control the confidence in the ground-truth labels and network predictions are inferred from our unsupervised noise model: $w_p = p(k = 1 \mid \ell_p)$ and $w_q = p(k = 1 \mid \ell_q)$. We compute h_p, z_p, h_q and z_q by doing an extra forward pass, as it is not straightforward to obtain the predictions for samples p and q from the mixed probabilities h.

Ideally, the proposed loss ℓ^* would lead to a better model by trusting in progressively better predictions during training. For high-levels of label noise, however, the network predictions are unreliable and dynamic bootstrapping may not converge when combined with the complex signal that *mixup* provides. This is reasonable as under high levels of noise most of the samples are guided by the network's prediction in the bootstrapping loss, encouraging the network to predict the same class to minimize the loss. We apply the regularization term used in (Tanaka et al., 2018), which seeks preventing the assignment of all samples to a single class, to overcome this issue:

$$R = \sum_{c=1}^{C} p_c \log \left(\frac{p_c}{\overline{h}_c} \right), \quad (14)$$

where p_c denotes the prior probability distribution for class c and $\overline{h}_c$ is the mean softmax probability of the model for class c across all samples in the dataset. Note that we assume a uniform distribution for the prior probabilities (i.e. $p_c = 1/C$), while approximating $\overline{h}_c$ using mini-batches as done in (Tanaka et al., 2018). We add the term ηR to ℓ^* (Eq. (13)) with η being the regularization coefficient (set to one in

Table 1. Validation accuracy on CIFAR-10 for static bootstrapping and the proposed dynamic bootstrapping. Key: CE (cross-entropy loss), ST (static bootstrapping), DY (dynamic bootstrapping), S (soft), and H (hard). Bold indicates best performance.

Alg./Noise level (%)		0	20	50	80
CE	Best	93.8	**89.7**	**84.8**	67.8
	Last	93.7	81.8	55.9	25.3
ST-S	Best	**93.9**	**89.7**	84.8	67.8
	Last	**93.9**	81.7	55.9	24.8
ST-H	Best	93.8	**89.7**	**84.8**	68.0
	Last	93.8	81.4	56.4	25.7
DY-S	Best	93.6	**89.7**	**84.8**	67.8
	Last	93.4	83.3	57.0	27.8
DY-H	Best	93.3	**89.7**	**84.8**	**71.7**
	Last	92.9	**83.4**	**65.0**	**64.2**

all the experiments). Subsection 4.3 presents the results of this approach and Subsection 4.5 demonstrates its superior performance in comparison to the state-of-the-art.

4. Experiments

4.1. Datasets and implementation details

We thoroughly validate our approach in two well-known image classification datasets: CIFAR-10 and CIFAR-100. The former contains 10 classes, while the latter has 100 classes. Both have 50K color images for training and 10K for validation with resolution 32×32. We use a PreAct ResNet-18 (He et al., 2016) and train it using SGD and batch size of 128. We use two different schemes for the learning rate policy and number of epochs depending on whether *mixup* is used (see Appendix B for further details). We further experiment on TinyImageNet (subset of ImageNet (Deng et al., 2009)) and Clothing1M (Xiao et al., 2015) datasets to test the generality of our approach far from CIFAR data (Subsection 4.6). TinyImageNet contains 200 classes with 100K training images, 10K validation, 10K test with resolution 64×64, while Clothing1M contains 14 classes with 1M real-world noisy training samples and clean training subsets (47K), validation (14K) and test (10K).

We follow (Zhang et al., 2017; 2018; Tanaka et al., 2018) criterion for label noise addition, which consists of randomly selecting labels for a percentage of the training data using all possible labels (i.e. the true label could be randomly maintained). Note that there is another popular label noise criterion (Jiang et al., 2018b; Wang et al., 2018b) in which the true label is not selected when performing random labeling. We also run our proposed approach under these conditions in Subsection 4.5 for comparison.

Table 2. Validation accuracy on CIFAR-10 (top) and CIFAR-100 (bottom) for joint mixup and bootstrapping. Key: CE (cross-entropy), M (mixup), DYR (dynamic bootstrapping + regularization from Eq. 14), S (soft), and H (hard). Bold indicates best performance.

Alg./Noise level (%)		0	20	50	80
CE	Best	94.7	86.8	79.8	63.3
	Last	94.6	82.9	58.4	26.3
M (Zhang et al., 2018)	Best	**95.3**	**95.6**	87.1	71.6
	Last	**95.2**	92.3	77.6	46.7
M-DYR-S	Best	93.3	93.5	89.7	77.3
	Last	93.0	93.1	89.3	74.1
M-DYR-H	Best	93.6	94.0	**92.0**	**86.8**
	Last	93.4	**93.8**	91.9	**86.6**
Alg./Noise level (%)		0	20	50	80
CE	Best	**76.1**	62.0	46.6	19.9
	Last	**75.9**	62.0	37.7	8.9
M (Zhang et al., 2018)	Best	74.8	67.8	57.3	30.8
	Last	74.4	66.0	46.6	17.6
M-DYR-S	Best	71.9	67.9	**61.7**	38.8
	Last	67.4	67.5	**58.9**	34.0
M-DYR-H	Best	70.3	**68.7**	**61.7**	**48.2**
	Last	66.2	**68.5**	58.8	**47.6**

4.2. Static and dynamic loss correction

Table 1 presents the results for static (ST) and dynamic (DY) bootstrapping in CIFAR-10. Although ST achieves performance comparable to DY (except for 80% noise where DY is much better), after the final epoch (last) the performance of DY outperforms ST. The improvements are particularly remarkable for 80% of label noise (from 25.7% of ST-H to 64.2 of DY-H). Comparing soft and hard alternatives: hard bootstrapping gives superior performance, which is consistent with the findings of the original paper (Reed et al., 2015). The overall results demonstrate that applying per-sample weights (DY) benefits training by allowing to fully correct noisy labels.

4.3. Joint *mixup* and dynamic loss correction

The proposed dynamic hard bootstrapping exhibits better performance than the state-of-the-art static version (Reed et al., 2015). It is, however, not better than the performance of *mixup* data augmentation, which exhibits excellent robustness to label noise (M in Table 2). The fusion approach from Eq. (13) (M-DYR-H) and its soft alternative (M-DYR-S), which combines the per-sample weighting of dynamic bootstrapping and robustness to fitting noise labels of *mixup*, achieves a remarkable improvement in accuracy under high noise levels. Table 2 reports outstanding accuracy for 80% of label noise, a case where we improve upon *mixup* (Zhang
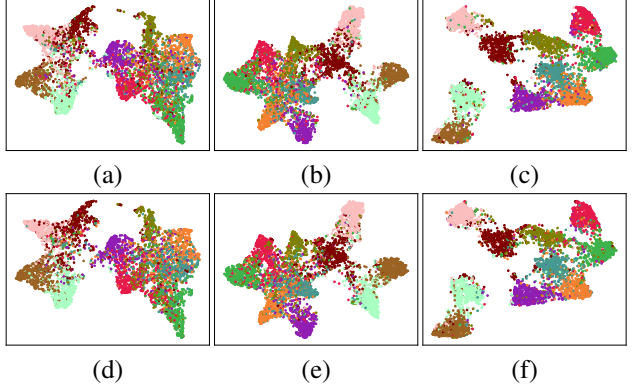

(a)   (b)   (c)

(d)   (e)   (f)

Figure 3. UMAP (McInnes et al., 2018) embeddings for training (top) with 80% of label noise and validation (bottom) on CIFAR-10 with (a)(d) cross-entropy loss from Eq. 1, (b)(e) mixup (Zhang et al., 2018) and (c)(f) our proposed M-DYR-H.

et al., 2018) in best (last) accuracy of 71.6 (46.7) in CIFAR-10 and 30.8 (17.6) in CIFAR-100 to 86.8 (86.6) and 48.2 (47.2) using the hard alternative (M-DYR-H). It is important to highlight that we achieve quite similar best and last performance for all levels of label noise in CIFAR datasets, indicating that the proposed method is robust to varying noise levels. Figure 3 shows uniform manifold approximation and projection (UMAP) embeddings (McInnes et al., 2018) of the 512 features in the penultimate fully-connected layer of PreAct ResNet-18 trained using our method, and compares them with those found using cross-entropy and *mixup*. The separation among classes appears visually more distinct using the proposed objective.

4.4. On the limits of the proposed approach

Table 3 explores convergence under extreme label noise conditions, showing that the proposed approach M-DYR-H fails to converge in CIFAR-10 with 90% label noise. Here we propose minor modifications to achieve convergence.

When clean and noisy samples are combined by *mixup* they are given the same importance of approximately $\delta = 0.5$ (as $\alpha = \beta = 32$). While noisy samples benefit from mixing with clean ones, clean samples are contaminated by noisy ones, whose training objective is incorrectly modified. We propose a dynamic *mixup* strategy in the input that uses a different δ for each sample to reduce the contribution of noisy samples when they are mixed with clean ones:

$$x = \left(\frac{\delta_p}{\delta_p + \delta_q} \right) x_p + \left(\frac{\delta_q}{\delta_p + \delta_q} \right) x_q, \qquad (15)$$

where $\delta_p = p(k = 0 \mid \ell_p)$ and $\delta_q = p(k = 0 \mid \ell_q)$, i.e. we use the noise probability from our BMM to guide *mixup* in the input. Note that for clean-clean and noisy-noisy cases, the behavior remains similar to *mixup* with $\alpha = \beta = 32$, which leads to $\delta \approx 0.5$ (i.e. $\delta_p \approx \delta_q \Rightarrow \delta_p/(\delta_p + \delta_q) \approx 0.5$).

Table 3. Validation accuracy on CIFAR-10 (top) and CIFAR-100 (bottom) with extreme label noise. Key: M (mixup), MD (dynamic mixup), DYR (dynamic bootstrapping + reg. from Eq. (14)), H (hard), and SH (soft to hard). (*) denotes that we have run the algorithm. Bold indicates best performance.

Alg./Noise level (%)		70	80	85	90
M-DYR-H	Best	**89.6**	**86.8**	71.6	40.8
	Last	**89.6**	**86.6**	71.4	9.9
MD-DYR-H	Best	86.6	83.2	**79.4**	56.7
	Last	85.2	80.5	**77.3**	50.0
MD-DYR-SH	Best	84.6	82.4	79.1	**69.1**
	Last	80.8	77.8	73.9	**68.7**
Alg./Noise level (%)		70	80	85	90
M-DYR-H	Best	**54.4**	**48.2**	**29.9**	12.5
	Last	**52.5**	**47.6**	**29.4**	8.6
MD-DYR-H	Best	54.4	47.7	19.8	13.5
	Last	50.8	41.7	8.3	3.9
MD-DYR-SH	Best	53.1	41.6	28.8	**24.3**
	Last	47.7	35.4	24.4	**20.5**

Table 4. Comparison with the state-of-the-art in terms of validation accuracy on CIFAR-10 (top) and CIFAR-100 (bottom). Key: M (mixup), MD (dynamic mixup), DYR (dynamic bootstrapping + reg. from Eq. 14), H (hard) and SH (soft to hard). (*) denotes that we have run the algorithm. Bold indicates best performance.

Alg./Noise level (%)		0	20	50	80	90
(Reed et al., 2015)*	Best	94.7	86.8	79.8	63.3	42.9
	Last	94.6	82.9	58.4	26.8	17.0
(Patrini et al., 2017)*	Best	94.7	86.8	79.8	63.3	42.9
	Last	94.6	83.1	59.4	26.2	18.8
(Zhang et al., 2018)*	Best	**95.3**	**95.6**	87.1	71.6	52.2
	Last	**95.2**	92.3	77.6	46.7	43.9
M-DYR-H	Best	93.6	94.0	**92.0**	**86.8**	40.8
	Last	93.4	**93.8**	91.9	**86.6**	9.9
MD-DYR-SH	Best	93.6	93.8	90.6	82.4	**69.1**
	Last	92.7	93.6	90.3	77.8	**68.7**
Alg./Noise level (%)		0	20	50	80	90
(Reed et al., 2015)*	Best	**76.1**	62.1	46.6	19.9	10.2
	Last	**75.9**	62.0	37.9	8.9	3.8
(Patrini et al., 2017)*	Best	75.4	61.5	46.6	19.9	10.2
	Last	75.2	61.4	37.3	9.0	3.4
(Zhang et al., 2018)*	Best	74.8	67.8	57.3	30.8	14.6
	Last	74.4	66.0	46.6	17.6	8.1
M-DYR-H	Best	70.3	68.7	61.7	**48.2**	12.5
	Last	66.2	68.5	58.8	**47.6**	8.6
MD-DYR-SH	Best	73.3	**73.9**	**66.1**	41.6	**24.3**
	Last	71.3	**73.4**	**65.4**	35.4	**20.5**

This configuration simplifies the input to the network when mixing a sample whose label is potentially useless, while retaining the strengths of *mixup* for clean-clean and noisy-noisy combinations. This is used with the original *mixup* strategy (Eq. (13)) to benefit from the regularization that an additional label provides. Table 3 presents the results of this approach (MD-DYR-H), which exhibits more stable convergence for 90% label noise in both datasets.

Table 2 reported that hard bootstrapping works better than the soft alternative. Unfortunately, hard bootstrapping under high levels of label noise causes large variations in the loss that lead to drops in performance. To ameliorate such instabilities, we propose a decreasing softmax technique (Vermorel & Mohri, 2005) to progressively move from a soft to a hard dynamic bootstrapping. This is implemented by modifying the softmax temperature T in:

$$h_{ij} = \frac{\exp(s_{ij}/T)}{\sum_{k=1}^{N} \exp(s_{ik}/T)}, \qquad (16)$$

where s_{ij} denotes the score obtained in the last layer of the CNN model class j of sample x_i. By default $T = 1$ gives the soft alternative of Eq. (13). To move from soft to hard bootstrapping we linearly reduce the temperature for h_p and h_q until we reach a final temperature in a certain epoch ($T = 0.001$ and epoch 200 in our experiments). We experimented with linear, logarithmic, tanh, and step-down temperature decays with similar results. This decreasing softmax MD-DYR-SH obtains much improved accuracy for 90% of label noise (69.1 for CIFAR-10 and 24.3 for CIFAR-100), while slightly decreasing accuracy compared to M-DYR-H and MD-DYR-H at lower noise levels. Note

that we significantly outperform the best state-of-the-art we are aware for 90% of label noise, which is 58.3% and 58.0% for best and last validation accuracies (reported in (Tanaka et al., 2018) with a PreAct ResNet-32 on CIFAR-10). The training process is slightly modified to introduce dynamic *mixup* (epoch 106) before bootstrapping (epoch 111) for MD-DYR-H and MD-DYR-SH.

4.5. Comparison with related approaches

Table 4 compares with related works for different levels of label noise using a common architecture and the 300 epochs training scheme (see Subsection 4.1) . We introduce bootstrapping in epoch 105 for (Reed et al., 2015) for the proposed methods, estimate the T matrix of (Patrini et al., 2017) in epoch 75 (as done in (Hendrycks et al., 2018)), and use the configuration reported in (Zhang et al., 2018) for *mixup*. We outperform the related work in the presence of label noise, obtaining remarkable improvements for high levels of noise (80% and 90%) where the compared approaches do not learn as well from the noisy samples (see best accuracy) and do not prevent fitting noisy labels (see last accuracy).

As noted in Subsection 4.1, when introducing label noise the true label can be excluded from the candidates. In this case

Table 5. Comparison with the state-of-the-art in terms of validation accuracy on CIFAR-10 (top) and CIFAR-100 (bottom). Key: M (mixup), MD (dynamic mixup), DYR (dynamic bootstrapping + reg. from Eq. 14), H (hard), SH (soft to hard), WRN (Wide ResNet), PRN (PreActivation ResNet, and GCNN (Generic CNN). Bold indicates best performance.

Algorithm	Architecture	Noise level (%)			
		20	40	60	80
(Jiang et al., 2018b)	WRN-101	92.0	89.0	-	49.0
(Ma et al., 2018)	GCNN-12	85.1	83.4	72.8	-
(Ren et al., 2018)	WRN-28	-	86.9	-	-
(Wang et al., 2018b)	GCNN-7	81.4	78.2	-	-
M-DYR-H	PRN-18	**94.0**	**92.8**	**90.3**	46.3
MD-DYR-SH	PRN-18	93.8	92.3	86.1	**74.1**

Algorithm	Architecture	Noise level (%)			
		20	40	60	80
(Jiang et al., 2018b)	WRN-101	73.0	68.0	-	35.0
(Ma et al., 2018)	RN-44	62.2	52.0	42.3	-
(Ren et al., 2018)	WRN-28	-	61.3	-	-
M-DYR-H	PRN-18	70.0	64.4	58.1	**45.5**
MD-DYR-SH	PRN-18	**73.7**	**70.1**	**59.5**	39.5

Table 6. Comparison of test accuracy on TinyImageNet. Key: M (mixup) , DYR (dynamic bootstrapping + reg. from Eq. 14), H (hard), and SH (soft to hard). (*) denotes that we have run the algorithm. Bold indicates best performance.

Alg./Noise level (%)		20	50	80
(Zhang et al., 2018)*	Best	53.2	41.7	18.9
	Last	49.4	31.1	8.7
M-DYR-H	Best	51.8	44.4	18.3
	Last	51.6	43.6	17.7
MD-DYR-SH	Best	**60.0**	**50.4**	**24.4**
	Last	**59.8**	**50.0**	**19.6**

label noise is defined as the percentage of incorrect labels instead of random ones (i.e. the criterion followed in previous experiments), a criterion adopted by several other authors (Jiang et al., 2018b; Ma et al., 2018; Ren et al., 2018; Wang et al., 2018b). We also run our proposed approach under this setup to allow quantitative comparison (Table 5). The proposed method outperforms all related work in CIFAR-10 and CIFAR-100 with MD-DYR-SH, while the results for M-DYR-H are slightly below those of (Jiang et al., 2018b) for low label noise levels in CIFAR-100. Nevertheless, these results should be interpreted with care due to the different architectures employed and the use of sets of clean data during training in (Jiang et al., 2018b) and (Ren et al., 2018).

4.6. Generalization of the proposed approach

Table 6 shows the results of the proposed approaches M-DYR-H and MD-DYR-SH compared to *mixup* (Zhang et al., 2018) on TinyImageNet to demonstrate that our approach is useful far from CIFAR data. The proposed approach clearly outperforms (Zhang et al., 2018) for different levels of label noise, obtaining consistent results with the CIFAR experiments. Note that we use the same network, hyperparameters, and learning rate policy as with CIFAR. Furthermore, we tested our approach in real-world label noise by evaluating our method on Clothing1M (Xiao et al., 2015), which contains non-uniform label noise with label flips concentrated in classes sharing similar visual patterns with the true class. We followed a similar network and procedure as (Tanaka et al., 2018) with ImageNet pre-trained weights and ResNet-50, obtaining over 71% test accuracy, which falls short of the state-of-the-art (72.23% (Tanaka et al., 2018)). We found that finetuning a pre-trained network for one epoch, as done in (Tanaka et al., 2018), easily fits label noise limiting our unsupervised label noise model. We believe this occurs due to the structured noise and the small learning rate. Training with cross-entropy alone gives test accuracy over 69%, suggesting that the configurations used might be suboptimal.

5. Conclusions

This paper presented a novel approach on training under label noise with CNNs that does not require any set of clean data. We proposed to fit a beta mixture model to the cross-entropy loss of each sample and model label noise in an unsupervised way. This model is used to implement a dynamic bootstrapping loss that relies either on the network prediction or the ground-truth (and potentially noisy) labels depending on the mixture model. We combined this dynamic bootstrapping with mixup data augmentation to implement an incredibly robust loss correction approach. We conducted extensive experiments on CIFAR-10 and CIFAR-100 to show the strengths and weaknesses of our approach demonstrating outstanding performance. We further proposed to use our beta mixture model to guide the combination of *mixup* data augmentation to assure reliable convergence under extreme noise levels. The approach generalizes well to TinyImageNet but shows some limitations under non-uniform noise in Clothing1M that we will explore in future research.

Acknowledgements

References

Alvi, M., Zisserman, A., and Nellaker, C. Turning a Blind Eye: Explicit Removal of Biases and Variation from Deep Neural Network Embeddings. *arXiv:1809.02169*, 2018.

1

Beluch, W., Genewein, T., Nürnberger, A., and Köhler, J. The Power of Ensembles for Active Learning in Image Classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

Bengio, Y., Louradour, J., Collobert, R., and Weston, J. Curriculum learning. In *International Conference on Machine Learning (ICML)*, 2009. 1

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 4.1

DeTone, D., Malisiewicz, T., and Rabinovich, A. Deep image homography estimation. *arXiv:1606.03798*, 2016. 1

Ding, Y., Wang, L., Fan, D., and Gong, B. A Semi-Supervised Two-Stage Approach to Learning from Noisy Labels. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018. 2

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised Representation Learning by Predicting Image Rotations. In *International Conference on Learning Representations (ICLR)*, 2018. 1

Guo, S., Huang, W., Zhang, H., Zhuang, C., Dong, D., Scott, M., and Huang, D. CurriculumNet: Weakly Supervised Learning from Large-Scale Web Images. In *European Conference on Computer Vision (ECCV)*, 2018. 2

He, K., Zhang, X., Ren, S., and Sun, J. Identity Mappings in Deep Residual Networks. In *European Conference on Computer Vision (ECCV)*, 2016. 4.1

Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. Using Trusted Data to Train Deep Networks on Labels Corrupted by Severe Noise. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 1, 2, 4.5

Jiang, J., Ma, J., Wang, Z., Chen, C., and Liu, X. Hyperspectral Image Classification in the Presence of Noisy Labels. *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–15, 2018a. 1

Jiang, L., Zhou, Z., Leung, T., Li, L., and Fei-Fei, L. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. In *International Conference on Machine Learning (ICML)*, 2018b. 1, 1, 2, 4.1, 4.4, 4.5

Krishna, R., Hata, K., Ren, F., Fei-Fei, L., and Niebles, J. C. Dense-Captioning Events in Videos. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. 1

Liang, S., Li, Y., and Srikant, R. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. In *International Conference on Learning Representations (ICLR)*, 2018. 2

Ma, X., Wang, Y., Houle, M., Zhou, S., Erfani, S., Xia, S.-T., Wijewickrema, S., and Bailey, J. Dimensionality-Driven Learning with Noisy Labels. In *International Conference on Machine Learning (ICML)*, 2018. 4.4, 4.5

Ma, Z. and Leijon, A. Bayesian Estimation of Beta Mixture Models with Variational Inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11): 2160–2173, 2011. 3.1, 3.1

McInnes, L., Healy, J., Saul, N., and Großberger, L. UMAP: uniform manifold approximation and projection. *The Journal of Open Source Software*, 3(29):861, 2018. 3, 4.3

Ono, Y., Trulls, E., Fua, P., and Moo Yi, K. LF-Net: Learning Local Features from Images. *arXiv: 1805.09662*, 2018. 1

Pathak, D., Girshick, R., Dollár, P., Darrell, T., and Hariharan, B. Learning Features by Watching Objects Move. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 4.4, 4.5

Permuter, H., Francos, J., and Jermyn, I. A study of Gaussian mixture models of color and texture features for image classification and segmentation. *Pattern Recognition*, 39(4):695–706, 2006. 3.1

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. In *International Conference on Learning Representations (ICLR)*, 2015. 1, 2, 3.2, 3.2, 4.2, 4.3, 4.4, 4.5

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to Reweight Examples for Robust Deep Learning. In *International Conference on Machine Learning (ICML)*, 2018. 1, 1, 2, 4.4, 4.5

Stauffer, C. and Grimson, W. E. L. Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pp. 246–252, 1999. 3.1

Tanaka, D., Ikami, D., Yamasaki, T., and Aizawa, K. Joint Optimization Framework for Learning with Noisy Labels. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3.3, 3.3, 4.1, 4.4, 4.6

Vahdat, A. Toward Robustness against Label Noise in Training Deep Discriminative Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2

Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., and Belongie, S. Learning From Noisy Large-Scale Datasets With Minimal Supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

Vermorel, J. and Mohri, M. Multi-armed Bandit Algorithms and Empirical Evaluation. In *European Conference on Machine Learning (ECML)*, 2005. 4.4

Wang, F., Chen, L., Li, C., Huang, S., Chen, Y., Qian, C., and Change Loy, C. The Devil of Face Recognition is in the Noise. In *European Conference on Computer Vision (ECCV)*, 2018a. 1

Wang, Y., Liu, W., Ma, X., Bailey, J., Zha, H., Song, L., and Xia, S.-T. Iterative Learning With Open-Set Noisy Labels. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018b. 1, 2, 4.1, 4.4, 4.5

Wu, X., He, R., Sun, Z., and Tan, T. A Light CNN for Deep Face Representation With Noisy Labels. *IEEE Transactions on Information Forensics and Security*, 13 (11):2884–2896, 2018. 1

Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X. Learning from massive noisy labeled data for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2, 4.1, 4.6

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires re-thinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017. 1, 2, 3.2, 4.1

Zhang, H., Cisse, M., Dauphin, Y., and Lopez-Paz, D. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 3, 2, 3.3, 4.1, 4.2, 4.3, 3, 4.4, 4.5, 4.6

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. Pyramid Scene Parsing Network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

Zlateski, A., Jaroensri, R., Sharma, P., and Durand, F. On the Importance of Label Quality for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1