Leveraging Adaptive Loss Scaling for Noisy Label Filtering

Kirill Pupkov, Igor Ignashin, Aleksandr Beznosikov

2025

Abstract

In machine learning, numerous challenges can degrade model performance, including noisy features and incorrect labeling in the training data. In this paper, we investigate the use of Adaptive Loss Scaling to address these issues. Our method is more effective at separating clean and noisy labels than standard empirical risk minimization. We demonstrate the effectiveness of our approach on the CIFAR10 dataset with a fraction of noisy labels.

1 Introduction

Machine learning models are often applied under the assumption that training data are correctly labeled and accurately represent the true data distribution. However, in practice, training sets often contain noisy labels, negatively affecting model performance. Recent work has also shown that sufficiently large convolutional networks can easily fit a random labeling of training data (Zhang et al., 2017). Various approaches have been proposed in the literature to address the problem of noisy labels in training data.

One particular family of approaches focuses on reweighting training examples. Some methods utilize meta-learning to learn these weights (Ren et al., 2018), while others use specialized networks to determine them (Jiang et al., 2018). However, such methods usually rely on the availability of a smaller clean training set, which is not always present. Recent work (Beznosikov, 2025) discusses how the task of assigning weights can be formulated as a min-max optimization problem and proposes a method for solving it using the ALSO optimizer. Notably, it does not require any clean set and can be used as a drop-in replacement for Adam (Kingma, 2014) optimizer.

Other approaches focus on modifying the loss function to be more robust to label noise. Reed et al. (2014) proposed a "bootstrapping" approach that includes model predictions into the loss function to prevent overfitting. Arazo et al. (2019) later added dynamic coefficients to this approach using unsupervised loss modelling. Lu and He (2022) further improved this approach by using ensembled model predictions from previous epochs.

In this paper, we propose a method to leverage the weights learned by a model trained with a slightly modified version of the ALSO optimizer to accurately identify mislabeled training examples.

The main contributions of our paper include:

- **A new method for label filtering.** We propose a novel approach for detecting mislabeled examples using the ALSO optimizer.
- **Empirical evaluation.** We extensively evaluate our approach and compare it against baseline.

2 Preliminaries

2.1 Supervised Classification

We address a C-class supervised classification task. Let $\mathcal{X} \subset \mathbb{R}^d$ be the input feature space and $\mathcal{Y} \subset \{0,1\}^C$ the one-hot encoded label space. We are given a training set $D = \{(x_i, y_i)\}_{i=1}^N$. Our goal is to find $\theta \in \Theta \subset \mathbb{R}^k$ such that $q_\theta : \mathcal{X} \to \mathbb{R}^C$ minimizes an empirical risk, solving the following optimization problem:

$$\min_{\theta \in \Theta} \left\{ \frac{1}{N} \sum_{i=1}^{N} f_i(\theta) + \frac{\tau}{2} \|\theta\|_2^2 \right\}, \tag{1}$$

Where $f_i(\theta) = \ell(y_i, q_\theta(x_i))$ for a chosen loss function ℓ, typically cross-entropy. $\tau > 0$ is a regularization parameter, and $\frac{\tau}{2}\|\theta\|_2^2$ is the regularization term to prevent overfitting.

2.2 Noisy Labels

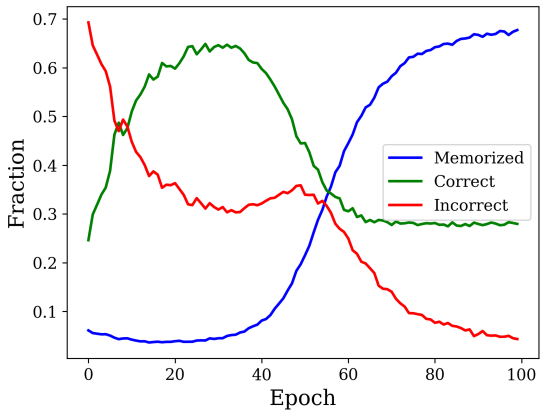
The challenge addressed in this paper is learning from a noisy training set $\tilde{D} = \{(x_i, \tilde{y}_i)\}_{i=1}^N$, where labels $\tilde{y}_i$ are potentially corrupted versions of the true labels y_i with some probability η.

Minimizing empirical risk (1) via SGD can lead to overfitting on the mislabeled examples, degrading model performance. Following Lu and He (2022), we observe the existence of a so-called "turning point", before which the model learns predominantly from clean examples, but after which it starts memorizing noisy labels (Figure 1).

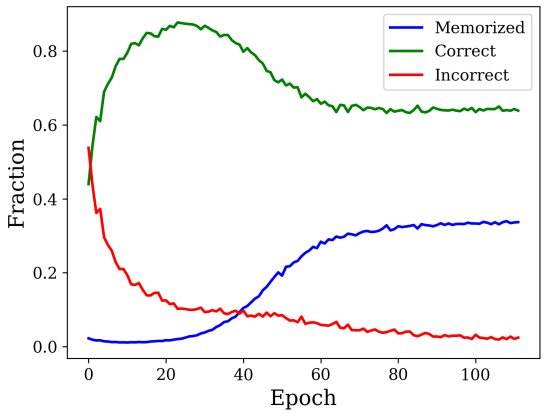
2.3 Noise Models

To assess the algorithm, we generate synthetic noise. The common noise types in the literature include:

- **Class-Conditional Noise**: Label corruption depends only on the true class, often modeled by a noise transition matrix.



(a) symmetric noise, $\eta = 0.8$



(b) symmetric noise, $\eta = 0.4$

Figure 1: Training the ResNet-18 model on CIFAR10 with noisy labels results in memorization. Predicted labels are classified as *correct* (predicted label matches the ground truth), *memorized* (predicted label matches the incorrect noisy label), or *incorrect* (predicted label matches neither the ground truth nor the noisy label).

- **Instance-Dependent Noise**: Label corruption depends on both the true class and the input features x_i (Zhang et al., 2021)

In this paper, we only work with class-conditional noise, as it is much simpler to simulate. Following standard practice, we divide noise into symmetric (uniform) and asymmetric (non-uniform) types:

- **Symmetric noise:** For each training example, the true label is replaced with a label chosen randomly from the set of all labels (including the true one) with probability η.
- **Asymmetric noise:** For each training example, the true label is replaced with a predetermined incorrect label (depending on the true class y_i) with probability η. Typically, the incorrect label is chosen to be a similar or easily confusable class.

Asymmetric noise better models real-world label corruption, as annotators are more likely to confuse similar classes.

2.4 ALSO

Beznosikov (2025) proposed the following reformulation of Equation 1:

$$\max_{\pi \in \Delta_{N-1}} \min_{\theta \in \Theta} \left\{ \sum_{i=1}^{N} \pi_i f_i(\theta) + \frac{\tau}{2} \|\theta\|_2^2 - \tau \mathrm{KL}[\pi \| \hat{\pi}] \right\}, \tag{2}$$

where $\tau > 0$ is the regularization parameter (temperature), $\mathrm{KL}[\cdot\|\cdot]$ denotes the KL-divergence between two distributions, and $\hat{\pi}$ is a prior distribution of weights from Δ_{N-1}.

A common choice for $\hat{\pi}$ is the uniform distribution, $\hat{\pi} = \mathcal{U}(\mathbf{1}, N)$. This formulation encourages higher weights for samples that are more difficult to classify correctly, helping to identify potentially mislabeled data. The authors solve the optimization problem in Equation 2 using different variations of the Mirror-Prox algorithm, constructing the ALSO optimizer.

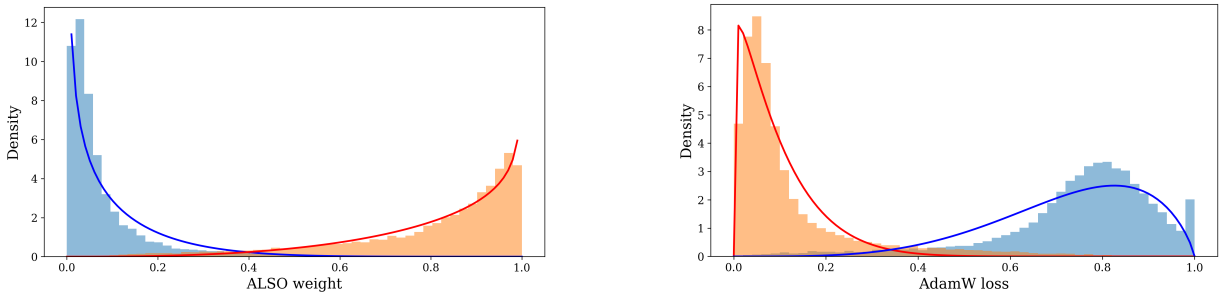
3 Method

3.1 ALSO tweak

We apply the ALSO optimizer to datasets with synthetic noise. However, we notice that assigning higher weights to mislabeled objects encourages the model to learn from them faster, leading to quicker overfitting. To counter this, we use a modified version of Equation 2 in which the maximum over π is replaced with a minimum. We observe that such a modification ensures that the model is much less prone to overfitting.

3.2 Loss Modelling

Following Arazo et al. (2019), we model the weights as a Beta mixture model, where the probability density function of the loss is given by $p(l) = \sum_{k=1}^{2} w_k \cdot p(l|k)$. The two components correspond to weights for clean and corrupted objects. Here, $p(l|k)$ is the probability density function of a Beta distribution with parameters α_k and β_k. We fit a Beta Mixture Model to the ALSO weights using the same EM algorithm described in Arazo et al. (2019). We observe that the ALSO weights fit the Beta Mixture Model much better than the loss values from classical optimizers (see Figure 2).



(a) ALSO weights at the end of training

(b) AdamW losses at "turning point"

Figure 2: Beta Mixture Model fitted to ALSO weights and AdamW losses under 0.4 symmetric noise. ALSO separates the classes better

4 Experiments

4.1 Setup

We use the CIFAR10 dataset (Krizhevsky et al., 2009) with standard augmentations (RandomCrop and RandomHorizontalFlip). For the network architecture we use ResNet-18 and ResNet-34 (He et al., 2016) We add random symmetric or asymmetric noise at different levels. We train both AdamW and ALSO for 100 epochs and collect the training loss (in the case of AdamW) and the weights (in the case of ALSO) after each epoch. Then, we fit a Beta Mixture Model for each epoch and use it to estimate each object's probability of being corrupted.

We evaluate the performance of our approach using ROC-AUC, which measures how well our method separates clean and noisy labels after each epoch.

4.2 Results

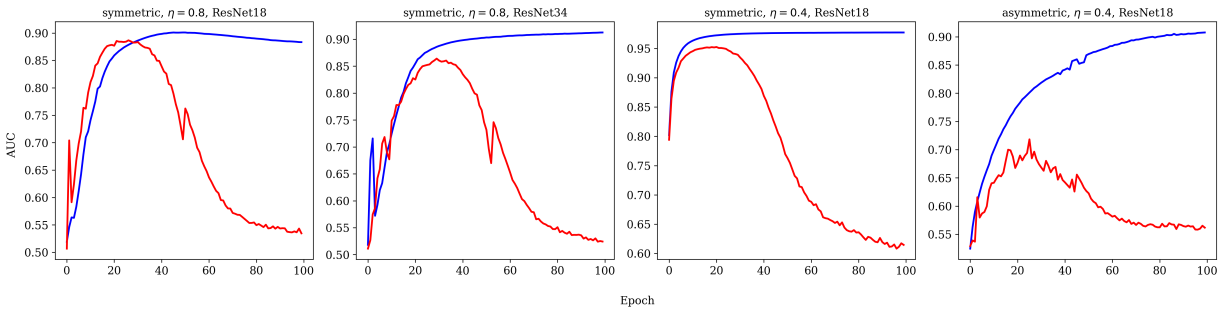


Figure 3: Measured ROC-AUC for different scenarios. The blue line represents ALSO, and the red line represents AdamW.

As shown in Figure 3, our method consistently outperforms the baseline approach in identifying mislabeled examples, achieving an AUC greater than 0.9 in all scenarios. An additional advantage of our method is its resistance to overfitting - as a result, the AUC remains stable and does not experience a sudden drop. This property is particularly valuable in practice, as it eliminates the need for early stopping that can be difficult to apply when a clean validation set is not available.

5 Conclusion

In this paper, we introduced an approach for identifying mislabeled examples in training data by leveraging a modified Adaptive Loss Scaling (ALSO) optimizer. We demonstrated that by changing the optimization objective of ALSO to minimize over sample weights, we can prevent the model from overfitting to noisy labels. By fitting a Beta Mixture Model to the learned sample weights, our method achieves superior separation between clean and

noisy examples compared to approaches relying on loss values from standard optimizers like AdamW.

References

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., and McGuinness, K. (2019). Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning (ICML)*.

Beznosikov (2025). Mirror-prox algorithm with linear convergence rate and its application for dynamic loss scaling.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. (2018). Mentornet: Learning data-driven curriculum for very noisy supervision. In *ICML*.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A., Nair, V., and Hinton, G. (2009). Cifar-10 (canadian institute for advanced research).

Lu, Y. and He, W. (2022). Selc: Self-ensemble label correction improves learning with noisy labels. *IJCAI*.

Reed, S., Lee, H., Szegedy, C., Erhan, D., and Rabinovich, A. (2014). Training deep neural networks on noisy labels with bootstrapping.

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

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

Zhang, Y., Zheng, S., Wu, P., Goswami, M., and Chen, C. (2021). Learning with feature-dependent label noise: A progressive approach. In *ICLR*.