

Class-conditional Importance Weighting for Deep Learning with Noisy Labels

Bhalaji Nagarajan^{1,†}^a, Ricardo Marques^{1,†}^b, Marcos Mejia¹^c and Petia Radeva^{1,2,*}^d

¹*Dept. de Matemàtiques i Informàtica, Universitat de Barcelona, Barcelona, Spain*

²*Computer Vision Center, Cerdanyola (Barcelona), Spain*

Keywords: Noisy Labeling, Loss Correction, Class-conditional Importance Weighting, Learning with Noisy Labels.

Abstract: Large-scale accurate labels are very important to the Deep Neural Networks to train them and assure high performance. However, it is very expensive to create a clean dataset since usually it relies on human interaction. To this purpose, the labelling process is made cheap with a trade-off of having noisy labels. Learning with Noisy Labels is an active area of research being at the same time very challenging. The recent advances in Self-supervised learning and robust loss functions have helped in advancing noisy label research. In this paper, we propose a loss correction method that relies on dynamic weights computed based on the model training. We extend the existing Contrast to Divide algorithm coupled with DivideMix using a new class-conditional weighted scheme. We validate the method using the standard noise experiments and achieved encouraging results.

1 INTRODUCTION


Deep Neural Networks (DNNs) tend to show an incredible upshot in performance when trained with large-scale labeled data under supervised environments (Krizhevsky et al., 2012). The strong and implicit assumption in training any DNN is that the dataset is clean and reliable. However, in real-world it is difficult to meet this assumption owing to the expensive cost and the time required to create such large high-quality datasets (Liao et al., 2021). The labelling cost is reduced substantially by crowd-sourcing the labelling process or by using an automated labelling system. However, this inherently leads to having errors in the labels.


Recent advances in DNNs show that it is possible to create learning algorithms that abide to less accurate training data (Sun et al., 2017; Pham et al., 2021; Ghiasi et al., 2021). However the DNNs have a tendency to overfit on the label noise (Zhang et al., 2021a). There are two common approaches to tackle the problem of overfitting on noisy labels - Semi-


Supervised Learning (SSL) and Learning with Noisy Labels (LNL) (Zheltonozhskii et al., 2021). SSL uses scarce high-quality labelled data to learn representations of large amount of unlabelled data (Hendrycks et al., 2019). LNL approach uses less expensive annotations, but uses noisy labels as a trade-off (Natarajan et al., 2013). Both approaches are closely related to each other and are often used in combination to help DNNs learn from less accurate samples (Zheltonozhskii et al., 2021; Li et al., 2020; Chen et al., 2021).


LNL has been already studied both in machine and deep learning (Frénay and Verleysen, 2013; Frénay et al., 2014; Nigam et al., 2020; Cordeiro and Carneiro, 2020). The objective of any LNL algorithm is to find the best estimator for a dataset distribution learnt from the original distribution with noise. It is necessary for the DNN to learn the noise structure and estimate the parameters accordingly. In many LNL approaches, there is short ‘warm-up’ phase where supervised learning or self-learning is used before dealing with the label noise. By using the warm up, it is possible to model the loss into a Mixture Model (Arazo et al., 2019). The main reasoning for using this phase is based on the behaviour of DNNs to learn the clean samples faster than the noisy samples (Arpit et al., 2017).

The next phase of LNL deals with adapting the noise of the distribution and achieve robust classi-

^a <https://orcid.org/0000-0003-2473-2057>

^b <https://orcid.org/0000-0001-8261-4409>

^c <https://orcid.org/0000-0002-6839-8436>

^d <https://orcid.org/0000-0003-0047-5172>

* IAPR Fellow

† Joint first authors

fiers. Several strategies have been proposed to make the LNL network learn the dataset distribution without the noise (Algan and Ulusoy, 2021). The commonly used Cross-Entropy Loss and Mean Absolute Error are not robust to the underlying noise (Ma et al., 2020) and it is important for the objective functions to be robust to the noise of the underlying distribution. Loss correction methods help in increasing the robustness of losses by modifying the loss functions based on the weights of the labels. In this paper, we propose a class-conditional loss correction method based on the importance of classes. The loss is adapted during each step of the training using weights computed from the classifier scores. This adjustment is carried out such that the classes that are weakly learned are emphasized better during the learning process. To validate the proposed method, we use the Contrast to Divide framework (Zheltonozhskii et al., 2021) and correct the loss during the training phase. Below, we outline the main contributions of this work.

- First, we propose the weighed version of the loss function for LNL. By weighting the unlabeled part of the training data, it is made possible to give more importance to the less learnt or hard to learn classes.
- Second, we do an extensive analysis of various components in the loss function and study the progression of the LNL framework. Moreover, we show improvement with respect to the state of art on LNL on a public dataset.

The rest of the paper is organized as follows. In Section 2, we briefly discuss the related work. We present the details of the proposed technique in Section 3. The experiments and evaluations used to validate the proposed method is explained in Section 4 followed by conclusion in Section 5.

2 RELATED WORK

There are several works in the literature on learning with noisy labels. In this section, we briefly review the recent literature that are relevant to our proposed method.

2.1 Learning with Noisy Labels

There are several classes of LNL algorithms, broadly falling into loss modifications and noise detection schemes. Some methods use label correction (Xiao et al., 2015; Li et al., 2017), where the noisy labels are corrected using inferences made by DNNs, which are in-turn trained only on clean labels, while other

methods use loss correction schemes. In this class of algorithms, the network aims at increasing its robustness towards noise by modifying the loss function (Han et al., 2018; Ma et al., 2020). A computationally efficient method based on noise similarity labels was used instead of learning from noisy class labels and was able to reduce the noise rate (Wu et al., 2021). In general, similarity-based approaches have been effective in many LNL algorithms where using a noise transition matrix serves as a bridge between the clean and noisy samples (Hsu and Kira, 2015; Hsu et al., 2019; Wu et al., 2020).

The loss correction methods are based on modifying the loss function with weights during the training of DNNs. Common problems with the existing loss functions are over-fitting of noise and under-learning. Importance weighting schemes have been effective in making the losses more robust to noise (Liu and Tao, 2015; Zhang and Sabuncu, 2018; Yu et al., 2019; Zhang and Pfister, 2021). The Symmetric Cross Entropy loss was created using a Reverse Cross Entropy term along with the Cross Entropy term to make the loss more robust to noise and achieve better learning of the samples (Wang et al., 2019). Normalization techniques proved to make the commonly used loss functions more robust to noise and also by using two robust loss functions to create an Active Passive Loss helped in boosting each other's performance (Ma et al., 2020). Backward and forward noise transition matrices, which are based on matrix inversion and multiplication were pre-computed and shew to increase the robustness of the loss function (Patrini et al., 2017). In the above discussed methods, the basic assumption in a relabeling approach is to have clean labels, which is also a limitation of these algorithms.

Another variation of LNL algorithms focuses on new learning schemes adapted to noisy labels (Malach and Shalev-Shwartz, 2017; Yu et al., 2019). DivideMix (Li et al., 2020) uses a co-teaching strategy to learn two networks simultaneously, so that one network learns from the other networks' confident samples. This algorithm uses a loss to fit a Gaussian Mixture Model in order to divide the samples into labeled and unlabeled set. A Beta-mixture model was also used to model the losses for learning the noise in an unsupervised manner (Arazo et al., 2019). Selective Negative Learning and Positive Learning were used to selectively apply positive learning on expected-to-be-clean data, which is obtained by Negative Learning, where complimentary labels were used instead of the actual labels (Kim et al., 2019). This approach proved to be very effective compared to the normal positive selection of samples. Early

Learning Regularization (Liu et al., 2020) learned the clean samples first, followed by noisy samples in later epochs. This method was beneficial as it prevented the network from memorization of the noisy samples. Data augmentation is also an effective means to combat the noisy label problem (Berthelot et al., 2019b; Li et al., 2020; Berthelot et al., 2019a; Sohn et al., 2020). AugDesc (Nishi et al., 2021) used weak augmentations to learn the loss and strong augmentations to improve the generalizations.

Most of the literature presented above, uses a combination of different LNL schemes to make the network robust to noise. In this paper, we propose a loss correction scheme on top of the already effective DivideMix and Contrast to Divide learning schemes to enhance the learning of models.

2.2 Self-supervised and Semi-supervised Learning

Semi-supervised learning algorithms utilize the unlabeled data by performing providing pseudo-labels to the unlabeled data and adding constraints to the objective functions. Regularization could be consistency regularization or entropy minimization. MixMatch (Berthelot et al., 2019b) combined both the regularization methods to produce labels to the unlabeled classes. ReMixMatch (Berthelot et al., 2019a) and FixMatch (Sohn et al., 2020) were adaptations of MixMatch, which used weakly augmented images to produce labels and predict against the strongly augmented images. It is also beneficial to remove wrong labels that have high levels of noise. By using only a portion of the training set which is correct, the same performance could be achieved (Ding et al., 2018; Kong et al., 2019).

Self-Supervised Learning (SSL) algorithms learn representations in a task-agnostic environment so that the representations are meaningful irrespectively of the labels. Contrastive loss has been vital in the recent success of SSL algorithms, which clusters data points based on the (dis-)similarity of classes (Wang and Liu, 2021). By using these representations, any downstream task could be well learned by the DNNs. SSL algorithms have been widely used in solving the noisy label problems. Since the networks are learned without labels, they are able to produce features that are robust to noise (Cheng et al., 2021). Data re-labeling helps in increasing the effectiveness of DNNs. The performance was boosted by using a parallel network to learn the portion of clean labels (Mandal et al., 2020). Supervised learning and self-supervised learning can also be used together as a co-learning scheme as this could maximize the learning behaviour using

both the constraints (Tan et al., 2021; Huang et al., 2021). Contrastive DivideMix (Zhang et al., 2021b) fuses the contrastive and semi-supervised learning algorithms.

DivideMix (Li et al., 2020) uses a semi-supervised training phase. It uses the MixMatch algorithm to perform label co-refinement and co-guessing on labeled and unlabeled samples. This works on per-sample loss behaviour and has been an effective technique to model the noise. One of the bottleneck in this method is the warm-up phase. This was overcome using the Contrast to Divide (Zheltonozhskii et al., 2021) method. Instead of using a supervised learning in DivideMix, this algorithm used a self-supervised learning method. In our proposed approach, we add an importance weighting scheme that would enable the algorithms to focus selectively on the classes.

3 IMPORTANCE WEIGHTING

In this section, we first brief the rationale behind the approach. We provide background information followed by the proposed weighted scheme.

3.1 Rationale

Our approach is motivated by the observation that learning is unbalanced across classes, that is, after a given number of epochs, the accuracy of the model tends to vary significantly over different classes. Our hypothesis is that, by focusing the learning effort in those classes for which the model is currently less efficient, the overall accuracy of the model can be improved. To test this hypothesis, we propose to enhance the DivideMix algorithm (Li et al., 2020) with a class-conditional importance weighting scheme which assigns a larger weight to the classes for which the model has a poorer performance.

3.2 Background

At each epoch, the DivideMix algorithm, on which we build, separates the training set into two disjoint sets: a set \mathcal{X} containing potentially clean data, and a set \mathcal{U} containing potentially noisy data. This separation between clean and noisy data is made by fitting a Gaussian mixture model to the softmax output of a pretrained network (Li et al., 2020). The loss function used for training thus combines the losses on both the potentially clean and noisy sets, and is given by (Li et al., 2020):

$$\mathcal{L} = \mathcal{L}_{\mathcal{X}} + \lambda_u \mathcal{L}_{\mathcal{U}} + \lambda_r \mathcal{L}_{\text{reg}}, \quad (1)$$

where \mathcal{L}_X is the cross-entropy loss over the augmented and mixed *clean* data \mathcal{X}' ; $\mathcal{L}_{\mathcal{U}}$ is a mean squared error loss over the augmented and mixed *noisy* data \mathcal{U}' ; and finally, \mathcal{L}_{reg} is a regularization term used to encourage the model to evenly distribute its predictions across all classes.

The loss $\mathcal{L}_{\mathcal{U}}$ of the noisy data is defined as:

$$\mathcal{L}_{\mathcal{U}} = \frac{1}{|\mathcal{U}'|} \sum_{(x,p) \in \mathcal{U}'} \|p - p_{\theta}(x)\|_2^2,$$

where $|\mathcal{U}'|$ is the number of noisy samples at the current epoch, p is the label assigned to each noisy sample x through co-guessing (Li et al., 2020), and $p_{\theta}(x)$ is the model prediction for x given the current model parameters θ . The regularization term \mathcal{L}_{reg} , in its turn, is given by:

$$\mathcal{L}_{\text{reg}} = \sum_c \pi_c \log \left(\pi_c \left(\frac{1}{|S|} \sum_{x \in S} p_{\theta}^c(x) \right)^{-1} \right), \quad (2)$$

where $S = \mathcal{X}' + \mathcal{U}'$, and $\pi_c = 1/C$ is a uniform prior distribution over the probability of each class in S . Providing a uniform prior distribution $\pi_c = 1/C$ in Equation (2) causes the loss to be minimal when the model yields exactly the same number of predictions for all classes in the data set.

3.3 Class-conditional Importance Weighted Loss

We now describe the approach taken to assign a weight for each class. Let \mathbf{f} be a vector of C elements, C being the number of classes in the data set. Each element $f_c \in \mathbf{f}$ is given by $f_c = 1 - F_1^c$, where F_1^c represents the F_1 score for a particular class c at the current epoch. The vector \mathbf{f} is then smoothed over a window of n_e epochs, yielding \mathbf{f}' . Then, the weight vector \mathbf{w} is computed as:

$$\mathbf{w} = \frac{\max(\lambda_w, \mathbf{f}')}{|\max(\lambda_w, \mathbf{f}')|} \times C, \quad (3)$$

where λ_w is a hyperparameter which has the role of limiting how far the resulting weights can deviate from the value 1. Hence, the resulting weights $\mathbf{w} = \{w_1, \dots, w_C\}$ of Equation (3) take a larger value for those classes for which the F_1 score is smaller.

Furthermore, to account for the importance of each class in the learning phase, we introduce the weights \mathbf{w} (Equation (3)) in the loss function $\mathcal{L}_{\mathcal{U}}$ of the unlabeled set (Equation (3.2)), yielding:

$$\mathcal{L}_{\mathcal{U}} = \frac{1}{|\mathcal{U}'|} \sum_{(x,p) \in \mathcal{U}'} w_c \|p - p_{\theta}(x)\|_2^2, \quad (4)$$

where w_c is the weight of class c , and c is the class of the image x . Finally, to apply the weights to the regularization loss \mathcal{L}_{reg} , we simply replace the uniform prior distribution $\pi_c = 1/C$ used in DivideMix (Equation (2)) by a non-uniform prior based on the weights \mathbf{w} , such that:

$$\pi_c = \frac{w_c}{C}, \quad (5)$$

where w_c is the class weight according to Equation (3), and C is the number of classes in the dataset. This way, when the model performs poorly for a given class due to not choosing that same class as many times as it should, using the prior specified in Equation (5) will encourage the model to increase the number of prediction for this same class. Figure 1 shows the pipeline of our proposed approach.

4 EXPERIMENTS

We evaluate our proposed framework following the common methodology in synthetic noise benchmarks. We use CIFAR-10 (Krizhevsky and Hinton, 2009) to validate the method, varying the amount of injected noise. We measure the performance of the networks using accuracy as an evaluation metric. We provide the accuracy over five runs for each noise ratio, following the results presented in Contrast to Divide (Zheltonozhskii et al., 2021).

4.1 Implementation Details

Similarly to (Li et al., 2020), we used a PreAct ResNet-18 architecture (He et al., 2016) for DivideMix. Moreover, we coupled DivideMix with *Contrast to Divide* (C2D), which has been shown to considerably boost the original DivideMix algorithm (Zheltonozhskii et al., 2021). As regards the injected noise in the CIFAR-10 data set, we used two types of label noise: symmetric and asymmetric. Given a target noise ratio, the symmetric noise is generated by randomly substituting the original label by a randomly selected new label chosen with a uniform probability over the rest of the class labels. Regarding the asymmetric noise, we follow (Zheltonozhskii et al., 2021) who designed the noise so as to mimic the structure of real-world noise labels substituting the labels with those of the most similar classes. In each experiment, the networks are optimized during 360 epochs.

Regarding the λ_r hyperparameter, we follow (Li et al., 2020) and set its value to 1. To compute the weights per class, we set $\lambda_w = 0.1$ in Equation (3), and use a smoothing window of 5 epochs (i.e., $n_e = 5$). The problem of selecting λ_u is discussed below.

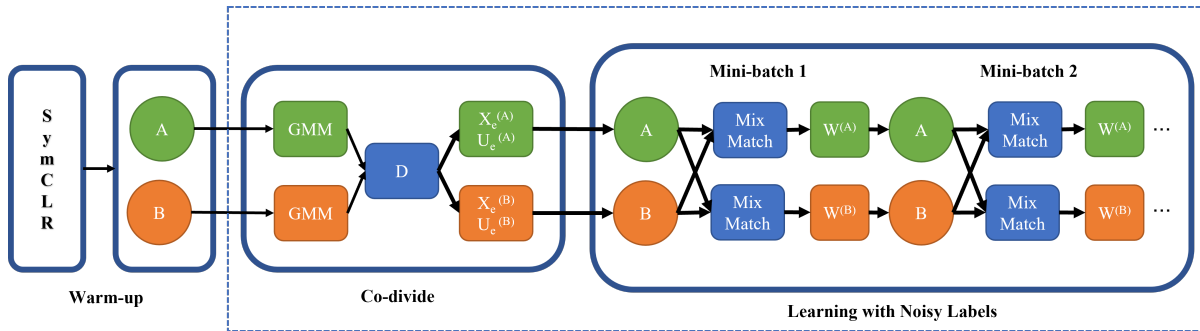


Figure 1: Pipeline of our proposed approach. The symCLR component pre-trains networks A and B. Then, these undergo the DivideMix warm-up phase where the two networks are trained on all the data set during a small number of epochs using standard cross-entropy loss. Then, at each epoch, the co-divide is applied to divide the data set in two disjoint sets, yielding the set of the clean and of the noisy labels (\mathcal{X} and \mathcal{U} , respectively). For each mini-batch, networks A and B are then trained separately using MixMatch and our proposed weighting scheme.

Table 1: Study of the optimal λ_u value. The table shows the peak and final accuracy on CIFAR-10 for $5 \leq \lambda_u \leq 50$.

Method		5	10	15	20	25	30	35	40	45	50
Weighted C2D +DM (90%)	Peak	91.73	92.45	92.92	93.50	93.50	93.60	93.58	93.69	93.52	93.66
	Final	91.45	92.35	92.73	93.33	93.48	93.53	93.47	93.28	93.42	93.57

4.1.1 Selection of λ_u

As in (Li et al., 2020) and (Zheltonozhskii et al., 2021), the performance of our proposed approach can vary significantly depending on the used parameter λ_u , i.e., the hyperparameter specifying the weight of the unsupervised loss $\mathcal{L}_{\mathcal{U}}$ in the final loss (see Equation (1)). Therefore, in Table 1, we provide a detailed analysis of the effect of this hyperparameter on the final accuracy reached by our method when considering 90% of symmetric noise. The results show that the optimal λ_u for our method with a noise ratio of 90% is of 40. This value is roughly in-line with the one (50) reported by (Li et al., 2020).

Following a similar approach, we collected a set of selected λ_u values for each considered noise ratio value. We can observe that the optimal λ_u value found seems to decrease with the noise ratio present in the dataset. This seems to indicate that, the larger the amount of noisy labels present in the used dataset, the more relevant the loss $\mathcal{L}_{\mathcal{U}}$ of the unlabeled data becomes in the learning process. The set of selected λ_u values for each considered noise ratio value are shown in Table 2. These values are used henceforth in all experiments for each corresponding noise value.

Table 2: Table with the selected λ_u values for each noise ratio.

Noise Ratio	20%	50%	80%	90%	40% (asym)
λ_u	0	25	30	40	0

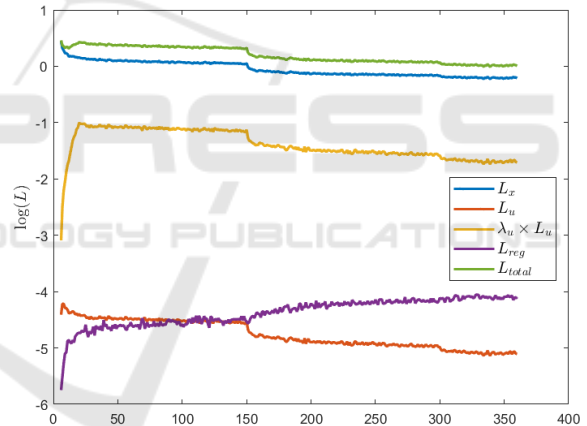


Figure 2: Different terms of the loss function as a function of the epoch. The plot was generated for CIFAR-10 with 80% of symmetric noise and $\lambda_u = 30$.

At this stage, it is interesting to analyze the role of the λ_u hyperparameter in the total loss value of Equation (1). Figure 2 depicts the different terms of the used loss function. We can observe that the hyperparameter λ_u acts as a scaling factor which brings $\mathcal{L}_{\mathcal{U}}$ (in red) up to a magnitude in which it can actually influence the final loss function shape (in orange the product $\lambda_u \times \mathcal{L}_{\mathcal{U}}$, and in green the final loss, denoted by L_{total} in Figure 2). It is also apparent from the curves that the loss of the labeled data ($\mathcal{L}_{\mathcal{X}}$, in blue) is the one that dominates the total loss shape. Finally, the regularization loss \mathcal{L}_{reg} seems to have a rather marginal role on the overall optimization process.

Table 3: Peak and final accuracy (% , mean \pm std over five runs) on CIFAR-10. DivideMix and C2D+DM results are obtained from literature.

Method		20%	50%	80%	90%	40% (asym)
DivideMix	Peak	96.1	94.6	93.2	76.0	-
	Final	95.7	94.4	92.9	75.4	-
C2D+DM	Peak	96.43 \pm 0.07	95.32 \pm 0.12	94.40\pm0.04	93.57 \pm 0.09	93.45 \pm 0.07
	Final	96.23 \pm 0.09	95.15 \pm 0.16	94.30\pm0.12	93.42 \pm 0.04	90.75 \pm 0.35
Weighted C2D+DM (ours)	Peak	96.50\pm0.07	95.79\pm0.06	94.40\pm0.05	93.70\pm0.16	93.62\pm0.09
	Final	96.40\pm0.21	95.56\pm0.07	94.24 \pm 0.09	93.54\pm0.13	92.83\pm0.21

4.2 Results and Analysis

The results for the application of our proposed method to the CIFAR-10 dataset are shown in Table 3, where a comparison with the results of the original method is provided. The results show that, when using our importance weighting scheme, the accuracy results generally improve over that of C2D+DM, and it never performs worse. Indeed, except for a noise level of 80%, our method delivers consistent improvements over its non-weighted counterpart. This confirms our hypothesis that the overall efficiency of the algorithm can be improved by focusing the learning effort in those classes that the model is having more difficulty to learn. Moreover, it also validates our weighting strategy based on the F_1 score proposed in Equation (3).

A detailed illustration of the weights values throughout the learning process is provided in Figure 3. It shows that the weights for a given class remain coherent through the learning phase, since we are able to clearly identify each class (corresponding to a particular color) through the weights plot. They also show that the weights for each class converge to a particular value, which is determined by the F_1 score that the model is able to get for each particular class as the learning progresses.

4.3 Ablation Study

In this section, we study the effect of weights in the $\mathcal{L}_{\mathcal{U}}$ and \mathcal{L}_{reg} terms individually. We show the results of this ablation study in Table 4. They show that when the weights are applied to only one of the two considered terms ($\mathcal{L}_{\mathcal{U}}$ and \mathcal{L}_{reg}), the accuracy is inferior to the case in which the weights are included in both losses.

5 CONCLUSIONS

In this paper, we propose a class-conditional dynamically weighted Contrast to Divide algorithm, where

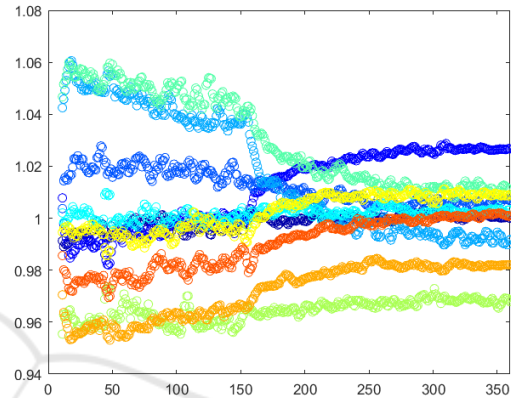


Figure 3: Illustration of the weights assigned to each class (y-axis) during 355 training epochs (x-axis, 5 warm-up epochs + 355 of DivideMix). The results are generated using the weights of a single network. Each color corresponds to a different class in the dataset (total of 10).

Table 4: Ablation study. The entries for C2D+DM and weighted C2D+DM is mean over five runs, whereas the other two are mean over two runs.

Method		80% ($\lambda_u = 30$)	90% ($\lambda_u = 40$)
C2D+DM	Peak	94.40\pm0.04	93.57 \pm 0.09
	Final	94.30\pm0.12	93.42 \pm 0.04
Weights in $\mathcal{L}_{\mathcal{U}}$ Only	Peak	94.23 \pm 0.12	93.68 \pm 0.09
	Final	94.11 \pm 0.16	93.51 \pm 0.16
Weights in \mathcal{L}_{reg} Only	Peak	94.28 \pm 0.06	93.58 \pm 0.09
	Final	94.11 \pm 0.05	93.37 \pm 0.11
Weighted C2D+DM	Peak	94.40\pm0.05	93.70\pm0.16
	Final	94.24 \pm 0.09	93.54\pm0.13

the weights emphasize the learning behaviour of individual classes. Here, we use a per-class importance weighting scheme based on F_1 -score obtained in each epoch. Our importance weighting approach proved to outperform the state of the art for the CIFAR-10 data set in all the noise rates. We studied the behavior of λ_u in different noise rates and also analysed the weights throughout the learning process. The results prove the effectiveness of the proposed scheme on an existing state of the art LNL approach.

Although, the algorithm has shown performance improvements, it is important to study the behaviour in more complex data sets such as CIFAR-100, Clothing 1M and WebVision. In this paper, we have used F1-score to create the weights, however, other methods have to be studied to compute the weights per class, which can eventually improve the results presented here. This information regarding the performance per class (i.e., F_1 score or other) can be used to improve other stages of the original DivideMix algorithm, such as, for example, the division between clean and noisy data.

ACKNOWLEDGEMENTS

This work was partially funded by TIN2018-095232-B-C21, SGR-2017 1742, Greenhabit EIT Digital program and CERCA Programme / Generalitat de Catalunya. Bhalaji Nagarajan acknowledges the support of FPI Becas, MICINN, Spain. We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPUs.

REFERENCES

- Algan, G. and Ulusoy, I. (2021). Image classification with deep learning in the presence of noisy labels: A survey. *Knowledge-Based Systems*, 215:106771.
- Arazo, E., Ortego, D., Albert, P., O'Connor, N., and McGuinness, K. (2019). Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning*, pages 312–321. PMLR.
- Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M. S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al. (2017). A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pages 233–242. PMLR.
- Berthelot, D., Carlini, N., Cubuk, E. D., Kurakin, A., Sohn, K., Zhang, H., and Raffel, C. (2019a). Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*.
- Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. (2019b). Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32.
- Chen, Y., Shen, X., Hu, S. X., and Suykens, J. A. (2021). Boosting co-teaching with compression regularization for label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2688–2692.
- Cheng, H., Zhu, Z., Sun, X., and Liu, Y. (2021). Demystifying how self-supervised features improve training from noisy labels. *arXiv preprint arXiv:2110.09022*.
- Cordeiro, F. R. and Carneiro, G. (2020). A survey on deep learning with noisy labels: How to train your model when you cannot trust on the annotations? In *2020 33rd SIBGRAP Conference on Graphics, Patterns and Images (SIBGRAP)*, pages 9–16. IEEE.
- Ding, Y., Wang, L., Fan, D., and Gong, B. (2018). A semi-supervised two-stage approach to learning from noisy labels. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1215–1224. IEEE.
- Frénay, B., Kabán, A., et al. (2014). A comprehensive introduction to label noise. In *ESANN*. Citeseer.
- Frénay, B. and Verleysen, M. (2013). Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869.
- Ghiasi, G., Zoph, B., Cubuk, E. D., Le, Q. V., and Lin, T.-Y. (2021). Multi-task self-training for learning general representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8856–8865.
- Han, B., Yao, J., Niu, G., Zhou, M., Tsang, I., Zhang, Y., and Sugiyama, M. (2018). Masking: A new perspective of noisy supervision. *Advances in Neural Information Processing Systems*, 31:5836–5846.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.
- Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. (2019). Using self-supervised learning can improve model robustness and uncertainty. *Advances in Neural Information Processing Systems*, 32:15663–15674.
- Hsu, Y.-C. and Kira, Z. (2015). Neural network-based clustering using pairwise constraints. *arXiv preprint arXiv:1511.06321*.
- Hsu, Y.-C., Lv, Z., Schlosser, J., Odom, P., and Kira, Z. (2019). Multi-class classification without multi-class labels. *arXiv preprint arXiv:1901.00544*.
- Huang, L., Zhang, C., and Zhang, H. (2021). Self-adaptive training: Bridging the supervised and self-supervised learning. *arXiv preprint arXiv:2101.08732*.
- Kim, Y., Yim, J., Yun, J., and Kim, J. (2019). Nlnl: Negative learning for noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 101–110.
- Kong, K., Lee, J., Kwak, Y., Kang, M., Kim, S. G., and Song, W.-J. (2019). Recycling: Semi-supervised learning with noisy labels in deep neural networks. *IEEE Access*, 7:66998–67005.
- Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105.
- Li, J., Socher, R., and Hoi, S. C. (2020). Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*.

- Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., and Li, L.-J. (2017). Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1910–1918.
- Liao, Y.-H., Kar, A., and Fidler, S. (2021). Towards good practices for efficiently annotating large-scale image classification datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4350–4359.
- Liu, S., Niles-Weed, J., Razavian, N., and Fernandez-Granda, C. (2020). Early-learning regularization prevents memorization of noisy labels. *Advances in Neural Information Processing Systems*, 33.
- Liu, T. and Tao, D. (2015). Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461.
- Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., and Bailey, J. (2020). Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*, pages 6543–6553. PMLR.
- Malach, E. and Shalev-Shwartz, S. (2017). “Decoupling” when to update” from” how to update”. *Advances in Neural Information Processing Systems*, 30:960–970.
- Mandal, D., Bharadwaj, S., and Biswas, S. (2020). A novel self-supervised re-labeling approach for training with noisy labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1381–1390.
- Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. (2013). Learning with noisy labels. *Advances in neural information processing systems*, 26:1196–1204.
- Nigam, N., Dutta, T., and Gupta, H. P. (2020). Impact of noisy labels in learning techniques: a survey. In *Advances in data and information sciences*, pages 403–411. Springer.
- Nishi, K., Ding, Y., Rich, A., and Hollerer, T. (2021). Augmentation strategies for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8022–8031.
- Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. (2017). Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952.
- Pham, H., Dai, Z., Xie, Q., and Le, Q. V. (2021). Meta pseudo labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11557–11568.
- Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*.
- Sun, C., Shrivastava, A., Singh, S., and Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852.
- Tan, C., Xia, J., Wu, L., and Li, S. Z. (2021). Co-learning: Learning from noisy labels with self-supervision. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1405–1413.
- Wang, F. and Liu, H. (2021). Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2495–2504.
- Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., and Bailey, J. (2019). Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 322–330.
- Wu, S., Xia, X., Liu, T., Han, B., Gong, M., Wang, N., Liu, H., and Niu, G. (2020). Multi-class classification from noisy-similarity-labeled data. *arXiv preprint arXiv:2002.06508*.
- Wu, S., Xia, X., Liu, T., Han, B., Gong, M., Wang, N., Liu, H., and Niu, G. (2021). Class2simi: A noise reduction perspective on learning with noisy labels. In *International Conference on Machine Learning*, pages 11285–11295. PMLR.
- Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X. (2015). Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699.
- Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., and Sugiyama, M. (2019). How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173. PMLR.
- Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2021a). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.
- Zhang, X., Liu, Z., Xiao, K., Shen, T., Huang, J., Yang, W., Samaras, D., and Han, X. (2021b). Codim: Learning with noisy labels via contrastive semi-supervised learning. *arXiv preprint arXiv:2111.11652*.
- Zhang, Z. and Pfister, T. (2021). Learning fast sample reweighting without reward data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 725–734.
- Zhang, Z. and Sabuncu, M. R. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. In *32nd Conference on Neural Information Processing Systems (NeurIPS)*.
- Zheltonozhskii, E., Baskin, C., Mendelson, A., Bronstein, A. M., and Litany, O. (2021). Contrast to divide: Self-supervised pre-training for learning with noisy labels. *arXiv preprint arXiv:2103.13646*.