Learning from Noisy Labels with Deep Neural Networks: combining sample selection and an adapted loss function

Alexandrine Lanson and Adam David

May 2023

Abstract

Learning from noisy labels is a key challenge in machine learning real-world situations. There are two major families of solutions for dealing with noisy labels: 1) detecting and cleansing noisy labels before training; 2) directly training noise-robust models on unclean data. Here, we consider two methods taken from the literature, one belonging to the first family and using a cyclical learning rate to detect noisy labels (O2U-Net, [2]), the second one coming from the other family and using combinations of robust loss functions (Active Passive Loss, [4]). We compare each method individually and combine both in a last experiment, with symmetric noise and two different noise rates (20% and 60%). We find that for in our framework, the three methods give better results than a base model not adapted for noisy labels, and that the combination method improves the base by 10.6 points for 20% noise and the sample selection method by 20.5 points for 60% noise in terms of accuracy.

1 Introduction

Deep learning methods have shown impressive performance in numerous machine learning tasks, handling large-scale datasets. However, their success is dependent on the availability of correctly labeled data, which requires a tremendous amount of expert time and resources. In real-world datasets, noisy labels are ubiquitous, with the ratio of corrupted labels in such datasets being reported to range from 8% to 38.5% [5]. The negative impact of label noise on the training of deep neural networks (DNNs) is mainly caused by overfitting, as DNNs ave a high capacity to remember noisy labels and therefore generalize badly [6].

There are two major families of solutions for dealing with noisy labels: 1) detecting and cleansing noisy labels before training; 2) directly training noise-robust models on unclean data (which can be divided into three main approaches as shown in Figure 1).

For this work, we will try one method for each solution: the first one uses multi-round learning, a technique that iteratively refines the dataset of clean examples by repeating the training round, while the second one uses a combination of robust loss functions. We will then combine both methods for our last experiment.

The first method is presented in Huang et al., 2019 [2] and is called O2U-Net, O2U standing for Overfitting to Underfitting. O2U-Net trains the model multiple times with a cyclical learning rate; based on the assumption that the gradient computation is dominated by the clean samples when the network is underfitting, the higher the loss of a sample, the higher the probability of being a noisy one. Thus, by calculating and ranking the normalized average loss of every sample, the mislabeled samples can be identified and removed to eventually train the model on cleaner data.

The second method (Ma et al., 2020 [4]) combines two types of robust loss functions (*active* and *passive*) that mutually boost each other to overcome the underfitting problem faced by loss functions robust to noisy labels.

We provide a theoretical framework as well as an implementation of our work than can be found in a github repository: `https://github.com/adam-dvd/DNN_noisy_labels`.

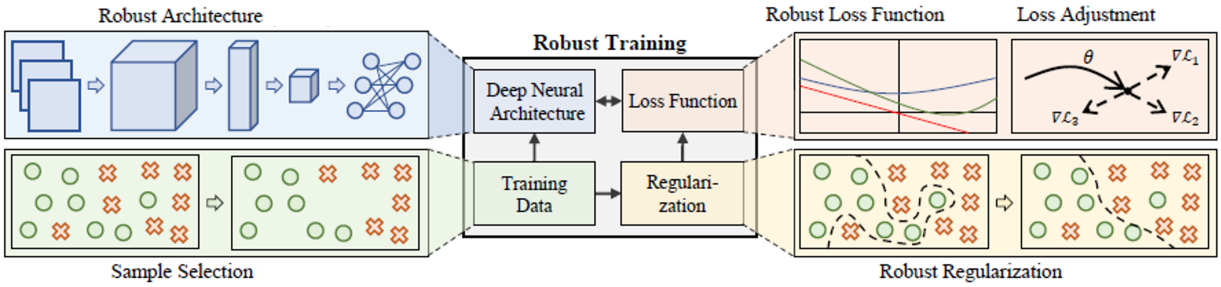

Figure 1: Categorization of recent deep learning methods for overcomming noisy labels. In this work, we will focus on a sample selection technique (02U-Net [2]) and a Robust Loss Function (Active Passive losses [4]). Image from [5].

2 Theoretical Framing

2.1 Noisy labels

For this work, we consider a C-class classification problem using a DNN. Let $\mathcal{X} \subset R^d$ be the feature space and $\mathcal{Y} = \{1, ..., C\}$ be the ground-truth label spacer. In a typical classification problem, we are provided with a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ obtained from an unknown joint distribution $\mathcal{P}_\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, where each (x_i, y_i) is independent and identically distributed. Here, we consider situations where we have a noisy training dataset $\hat{\mathcal{D}} = \{(x_i, \hat{y}_i)\}_{i=1}^N$ obtained from a noisy distribution $\mathcal{P}_{\hat{\mathcal{D}}}$ over $\mathcal{X} \times \hat{\mathcal{Y}}$, where $\hat{y}$ is a noisy label which may not be true.

??? We consider instance-independent label noise, that is the corruption process is conditionally independent of data features $\{x_i\}$ when the true label is given. We denote T the noise transition matrix, where $T_{ij} = P(\hat{y} = j|y = i, x) = P(\hat{y} = j|y = i) = \tau_{ij}$ defines the probability that a true label $y = i$ is flipped to j. Moreover, we will restrict the study to uniform noise, which corresponds to $T_{ij} = \frac{\tau}{C-1}$ for $i \neq j$ (C being the number of classes) and $T_{ii} = 1-\tau$, where τ is the constant noise level. We also restrict ourselves to situations where we are provided with clean validation and test sets.

The goal of our classification task on noisy labels is to learn the mapping function $f(\cdot, \Theta) : \mathcal{X} \to 1, ..., C$ of the DNN parametrized by Θ such that the parameter Θ minimizes the empirical risk $\mathcal{R}_{\hat{\mathcal{D}}(f)}$. Let's consider a mini-batch $\mathcal{B}_t = \{(x_i, \hat{y}_i)\}_{i=1}^b$ containing b examples obtained randomly from $\hat{\mathcal{D}}$ at time t. The DNN parameter Θ_t is updated along the descent direction of the empirical risk on mini-batch $\mathcal{B}_t$,

$$\Theta_{t+1} = \Theta_t - \eta \nabla \left(\frac{1}{|\mathcal{B}_t|} \sum_{(x, \hat{y} \in \mathcal{B}_y)} l(f(x; \theta_t), \hat{y}) \right)$$

where l is a chosen loss function and η the learning rate.

In the framework of sample selection, that is selecting true labels from a noisy training dataset to render the DNN more robust from noisy labels, let $\mathcal{C}_t \subseteq \mathcal{B}_t$ be the identified clean examples at time t. The DNN is updated only for the selected clean examples $\mathcal{C}_t$ (replace $\mathcal{B}_t$ by $\mathcal{C}_t$ in the previous equation), while other examples from the mini-batch (likely to be false labeled) are excluded of the training. The second method considered here, active passive loss (APL), concerns the choice of the loss function l. More details about these two methods are given below.

2.2 O2U-Net

O2U-Net's [2] intuition relies on the training process of DNNs: given a dataset, the network tends to first learn from the samples which are 'easy' to fit (here, clean samples), while the 'hard' samples (here, noisy samples) are usually learned at the late stage of the training. Thus, the losses of noisy labels are first larger than those of clean samples, but at the end, the losses generated from noisy and clean labels are indistinguishable as both of them are memorized by the network. The aim is therefore to track the loss of every sample at different stages of training and find a way to know when the noisy labels are overfitted. Indeed, with usual training, the status of the network would change from underfitting to overfitting only once and would not provide sufficient statistics to find when it occurs. The idea of the authors is to introduce multiple rounds of status transfer in training, changing between underfitting and overfitting using a cyclical learning rate. A large learning rate is first applied, then linearly decreases, and is then reset to the original value to jump from overfitting to underfitting.

In the cyclical train, suppose the maximum cyclical learning rate is r_1, and the minimum learning rate is r_2, where $r_1 > r_2$. The equation for learning rate adjustment during the cyclically training used

by the authors is as follows:

$$s(t) = \frac{(1 + ((t-1) \bmod c))}{c};$$
$$r(t) = (1 - s(t)) \times r_1 + s(t) \times r_2,$$

where t refers to the t th epoch in the cyclical training, c is the total number of epochs in each cyclical round and $r(t)$ is the learning rate applied at t. The number of rounds is arbitrary, determined when enougn statistics are gathered.

The whole training process of O2U-net comprises three steps: 1) pre-training of the network on the original dataset including noisy labels with a constant learning rate and a large batch size; 2) cyclical training, with a smaller batch size to make the network more easily transfer from overfitting to underfitting, and with multiple rounds based on the cyclical learning rate. After the whole cyclical training, the average of the normalized losses of every sample is computed. All the average losses are then ranked in descending order, and the top k% of samples are removed from the original dataset as noisy labels, where k depends on the prior knowledge on the dataset; 3) Training on cleaned data.

2.3 Active passive loss (APL)

In their paper, Ma et al. [4] note that loss functions robust to noisy labels suffer from a problem of underfitting. They first show that a simple normalization can make any loss function robust to noisy labels; that is, with $\mathcal{L}$ a loss function,

$$\mathcal{L}_{\text{norm}} = \frac{\mathcal{L}(f(\boldsymbol{x}), y)}{\sum_{j=1}^{K} \mathcal{L}(f(\boldsymbol{x}), j)}.$$

is robust. However, if such a property is sought in our framework, the underfitting mentionned above is caused by the extra terms introduced into the denominator by this normalization. The normalized cross entropy (NCE) is an example of such a situation: NCE is in the form of $\frac{P}{P+Q}$, where $P = -\log(p_y)$ and $Q = -\sum_{k \neq y} \log(p_k)$. During training, the Q term may increase even when P is fixed (eg. p_y is fixed), and it reaches the highest value when all $p_{k \neq y}$ equals to $(1 - p_y)/(K - 1)$ (eg. the highest entropy). This implies that the network may learn nothing for the prediction (as p_y is fixed) even when the loss decreases (as Q increases), causing the underfitting problem.

The authors then propose a characterization of existing loss functions into two types: 1) "Active" loss, which only explicitly maximizes the probability of being in the labeled class, and 2) "Passive" loss, which also explicitly minimizes the probabilities of being in other classes. Formal definitions are:

Definition 1. (Active loss function) $\mathcal{L}_{\text{Active}}$ is an active loss function if $\forall(\boldsymbol{x}, y) \in \mathcal{D} \forall k \neq y \ell(f(\boldsymbol{x}), k) = 0$.
Definition 2. (Passive loss function) $\mathcal{L}_{\text{Passive}}$ is a passive loss function if $\forall(\boldsymbol{x}, y) \in \mathcal{D} \exists k \neq y \ell(f(\boldsymbol{x}), k) \neq 0$.

The idea of the authors is to combine a robust active loss and a robust passive loss into an APL framework for both robust and sufficient learning. Formally,

$$\mathcal{L}_{\text{APL}} = \alpha \cdot \mathcal{L}_{\text{Active}} + \beta \cdot \mathcal{L}_{\text{Passive}},$$

where, $\alpha, \beta > 0$ are parameters to balance the two terms. The resulting APL is robust (we will not detail the demonstration here, it can be found in the paper).

Using the previous example of the NCE, the passive loss explicitly minimizes (at least one component of) the Q term discussed above so that it wont increase when p_y is fixed. This directly addresses the underfitting issue of a robust active loss. Therefore, according to the authors, APL losses can leverage both the robustness and the convergence advantages.

3 Experiments and Protocol

Choice of dataset We perform our experimentations on CIFAR10, a dataset widely used in the literature [5]. CIFAR-10 is a clean dataset that consists of 60000 32x32 colour images separated in 10 classes, with 6000 images per class. Because our computational power is limited, we chose to reduce the dataset's size and keep only four classes (dog, cat, deer, horse), which gives 20000 (4*5000) training images and 4000 (4*1000) test images [3]; we denote this dataset CIFAR4. As it is clean, we generate synthetic noisy labels by randomly flipping the original ones, creating uniform flipping. We conduct experiments with two percentages of noisy labels: 20% and 60%.

Method We will compare four different methods: the first one uses a network mentioned below without any specific adaptation to noisy labels; the second one uses the idea of [2] and cyclically changes the learning rate to identify noisy data and remove them from the sample; the third one relies on the idea of [4] described in introduction, using a combination of NCE and RCE (Reverse Cross Entropy) that we call NCE-RCE. Finally, we will combine both methods by first applying the O2U-Net method then use the NCE-RCE loss after having cleaned the dataset.

Network architecture For all methods described above, we use the same network, the 9-layer convolutional neural network (CNN) used in the O2U-Net paper, that is said to be a 'standard

test bed for weakly-supervised learning' [1]. It's architecture is given in Figure 2.

CNN on CIFAR-4
32×32 RGB Image
3×3 conv, 128 LReLU
3×3 conv, 128 LReLU
3×3 conv, 128 LReLU
2×2 max-pool, stride 2
dropout, $p = 0.25$
3×3 conv, 256 LReLU
3×3 conv, 256 LReLU
3×3 conv, 256 LReLU
2×2 max-pool, stride 2
dropout, $p = 0.25$
3×3 conv, 512 LReLU
3×3 conv, 256 LReLU
3×3 conv, 128 LReLU
avg-pool
dense 128→4

Figure 2: CNN model used in our experiments with CIFAR4. Image adapted from [1].

Concerning parameters, we try to stick as much as possible to both studied papers. We use torch's stochastic gradient descent as optimizer with a momentum of 0.9 and a weigth decay of $1e-4$. The initial learning rate is set to 0.01. For O2U-Net, the learning rate varies between $r_1 = 0.01$ and $r_2 = 0.001$, with c, the total number of epochs in each cyclical round, set to 10. Moreover, for both noise rates, we set the value of the forget rate (the percentage of data removed from the initial dataset) to that of the noise rate.

The loss used for the O2U-Net only experiment is the cross-entropy; we choose NCE-RCE for the active passive loss experiment with $\beta = \alpha = 1$ as recommended in the corresponding paper for datasets such as CIFAR10.

We train each method with the same total number of epochs, set to 100. For the base model - respectively the active-passive loss method, all epochs are allocated to a training phase with the CE - respectively NCE-RCE loss. For O2U-Net, we split these into 30 for the first stage, 20 for the second stage and 50 for the final training (recall the three steps described in section 2.2). This choice is motivated by the fact that the remaining noise in the cleaned step during the second stage stabilizes that after around 20 epochs. We chose this setting because we our computational power is limited, and the time of training is an important variable for us and for DNNs training in general. One might argue that other factors influence the time of training, thus that comparing models with the same number of epochs is not sufficient; however, we believe using the same network and the same parameters whenever possible assures a fair comparison between methods.

We also combine both O2U-Net and APLoss in the last experiment; to do so, we use the filter mask created during O2U-Net's second stage to clean the dataset and then train the model on the new data with the NCE-RCE loss.

Evaluation metric A typical metric used to assess method's performance in our framework is test accuracy. As we consider only a balanced dataset with symmetric noise, this metric seems appropriate to measure our model's performance.

4 Results

Table 4 shows the accuracies obtained for noise rates of 20% and 60% and for the four methods considered: base model (training the CNN on the initial dataset with a CE loss), APLoss (training the same CNN using NCE-RCE loss), O2U-Net (same CNN, CE loss, detecting and removing noisy samples) and finally a combination of APL and O2U-Net (same CNN, the two first stages are the same as O2U-Net alone, but the last stage uses NCE-RCE). We find that similar results are obtained for O2U-Net and the combination of O2U-Net and APLoss: for 20% noise, the best accuracy is given by O2U-Net & APLoss and for 60%, by O2U-Net alone (all results being much better than for 60% as expected; note that for O2U-Net and O2U-Net & APLoss, the sample is reduced to 16000 after cleaning for 20% noise and 8000 for 60% noise).

On Figure 3 are plotted the accuracies of each method for both noise rates (20% on the top chart, 60% on the bottom chart).

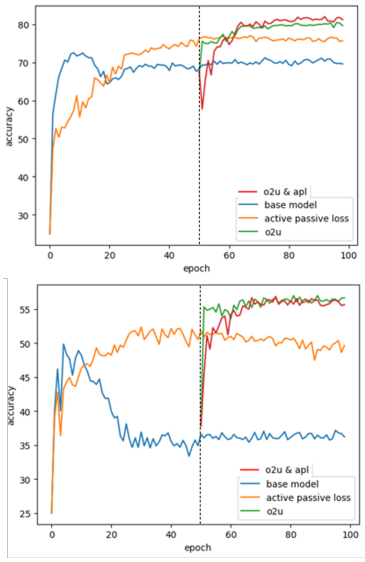

Figure 3: Accuracies vs epochs for all methods and (top) 20% noise, (bottom) 60% noise on CIFAR4. Methods are: base model, active passive loss, O2U-Net (o2u) and a combination of O2U-Net and active passive loss (o2u & apl). The dotted line marks epoch 50; base model and active passive loss are trained on 100 epochs, while we represent only the third stage trained on 50 epochs for O2U-Net and the combination (o2u & apl), that is the stage after cleaning the initial dataset (the first 50 stages are dedicated to initializing weights and detecting noisy samples, see section 3). Note the values' scope is not the same on both y-axes.

Method \ Dataset	CIFAR-4 20%: test acc.	CIFAR-4 20%: noise acc.	CIFAR-4 60%: test acc.	CIFAR-4 60%: noise acc.
Base	69.6		36.2	
O2U-Net	79.6	76.1	**56.7**	77.6
APLoss	75.7		49.6	
O2U-Net & APLoss	**81.2**	76.1	55.7	77.6

Table 1: Models accuracies on CIFAR4 (CIFAR10 reduced to 4 classes) dataset with noise rates set to 20% and 60% for the different methods studied. The noise accuracy is also given (masked noisy data divided by total masked data), and is the same for O2U-Net and O2U-Net&APLoss as the same filter mask was applied.

The base model accuracy curve clearly exhibits the overfitting problem highlighted by Ma et al. [4] faced by non robust loss fonctions such as Cross Entropy (CE). The results for the combination of two robust losses, that is Normalized Cross Entropy and Reverse Cross Entropy, are in agreement with the findings of the corresponding paper [4] and show a clear improvement compared to the base model, an improvement more pronounced for a high noise rate.

Best results are obtained for methods using O2U-Net technique, that belongs to the sample selection family. Adding the NCE-RCE loss shows no significant improvement for both noise rates, which suggests removing noisy samples for a noisy dataset is in our framework a better solution than training a robust model on the initial dataset. Nevertheless, the APL method allows for full exploration of the dataset and requires no prior knowledge such as the noise rate, while here, we removed a percentage of data equal to the known noise rate. The advantages of APL are therefore potentially only marginally exploited here when both methods are combined.

5 Conclusion

This project allowed to study two families of solutions to adress the challenge of learning with noisy labels. By training all methods with the same number of epochs, we find the best results are given for methods involving the sample selection technique from Huang et al. [2].

These results have to be replaced in context. Indeed, due to a very limited computational power, we had to restrict our experiments to one dataset with synthetic symmetric noise. This results obtained here might not directly apply to real-world situations; the next step would be to test the methods considered on datasets such as WebVision, with a more approximate noise rate and complex noise distribution. Training the model on more epochs and more data would also be more in line with the resources typically used (the papers studied here used around 200 epochs, and CIFAR10 as the smallest dataset). Other types of noise could also be considered, such as pair flip noise (a dog can only be misclassified as a cat, a deer as a horse, etc), as well as other noise rates.

For the O2U-Net setup, we chose to set the forget rate (the percentage of data removed from the initial dataset) to the known noise rate, which gives good results compared to other methods; however, in a more realistic context, the noise rate might not be known. Thus, it would be interesting to investigate the loss distribution during the second step, and more precisely loss gaps between samples to try to identify the percentage of noisy labels in the initial dataset.

For further investigation, an interesting problem that could be studied would be a binary classification problem, where noise detection as performed in the second stage of O2U-Net could be used not to remove the noisy samples but to change their labels to the other class, in order to keep a maximum amount of data for the model's training and allow for full exploration of the dataset.

References

[1] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pages 8535–8545, 2018.

[2] Jinchi Huang, Lie Qu, Rongfei Jia, and Binqiang Zhao. O2u-net: A simple noisy label detection approach for deep neural networks. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3325–3333, 2019.

[3] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The CIFAR-10 and CIFAR-100 datasets, 2014.

[4] Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels, 2020.

[5] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from

noisy labels with deep neural networks: A survey, 2022.

[6] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, 2017.