Seek Common while Shelving Differences: A New Way for dealing with Noisy Labels

Zhitong Gao*
ShanghaiTech University
Shanghai, China
gaozht@shanghaitech.edu.cn

Ziyao Zeng*
ShanghaiTech University
Shanghai, China
zengzy@shanghaitech.edu.cn

Abstract

Deep Neural Network has achieved remarkable performance in Image Classification. In Image Classification, learning with noisy labels is an essential task due to its ubiquity and large damage to Deep Neural Network. Based on memorization effects of deep neural networks, " Co-teaching [3]" cross-trains two nets using the small-loss trick and the state of-the-art approach"JoCoR [14]" improves "Co-teaching [3]" by focusing on the "agreement" of the two networks. In this work, we combine both the benefits of "cross training" and "agreement" by introducing a tri-net framework called SCSD (Seek Common while Shelving Differences). Extensive experimental results on corrupted data from benchmark datasets including MNIST, CIFAR-10, CIFAR-100 demonstrate that SCSD is superior to many state-of-the-art approaches for learning with noisy labels in Image Classification. To improve practicability of SCSD, we further use accuracy gap between small loss samples and large loss samples to do early-stopping. Experiments have shown that our early-stopping method can help us find a better stopping point during training.

1. Introduction

Image Classification is quite simple if all training data are properly labelled. But what if there are plentiful mislabelled training data? Consider a set $\{x_i, y_i\}$ of image x_i and its corresponding label y_i which is generated from a latent true label t_i. In common supervised problems, the given labels y and true labels t are assumed to be the same, since their differences are minus and will not affect model's learning. In noisy label problem, the difference between given labels and true labels are large and the amount of mislabelled data is huge, so they will do damage to learning models.

Without considering the differences between given label and true label, most supervised algorithms rely heavily on the quality of given labels. In real world, the high quality of labels are obtained at the cost of money and manpower. In some tasks with complicated labels, like biology experiments which usually obtain weak outcome, getting a purely clean labels can be seen as impossible. On the other hand, Deep Neural Networks (DNN) are studied to be heavily rely on the quality of labels[2]. With strong memorization ability, DNN can overfit to noisy labels, making its generalization ability poorly.

There are two broad ways to deal with noisy labels. One direction focus on label correction. Another focus on learning with noisy labels based on the principle for DNN to learn image features. Our work belongs to the latter one. Among the methods in learning with noisy labels, some focus on estimating the latent noisy transition matrix [7] [12]. However, the noise transition matrix is hard to estimate, especially with large class numbers. An alternative approach is training on selected or weighted samples, e.g., Mentornet[4], gradient-based reweight[11] and Co-teaching[3]. Mentornet and gradient-based reweight rely on a small clean dataset of similar data (like pre-train on clean Cifar-10 dataset to learn noisy Cifar-100 dataset), which is sometimes hard to get in real world. By contrast, Co-teaching does not need any additional clean data and is more practicable in real world applications. It's based on the observation of how DNN learns image features. Furthermore, the state-of-the-art method JoCoR[14] has shown excellent performance in learning with noisy labels by adding co-regularization to reduce the diversity of two networks during training. However, the joint update way is easy for error to propagate.

Motivated by Co-teaching, we try to use "cross update" to prevent error propagate and at the same time leverage the "agreement" strategy in JoCoR. To achieve this goal, we propose a tri-net structure named SCSD (Seek Common while Shelving Difference), Specifically, every two nets select a relative clean set by agreement, and use it to update the other network. In this way, we utilize the "agreement"

* indicates equal contributions.

strategy in selection step to get a set of cleaner samples and utilize "cross update" strategy to prevent error propagate. A co-regularization term is also added to force the three nets to converge.

To show that SCSD significantly improves the robustness of deep learning on noisy labels, we conduct extensive experiments on MNIST, CIFAR-10 and CIFAR-100 datasets with different simulated noise pattern and noise level. Empirical results demonstrate that the robustness of deep models trained by our proposed approach is superior to many state-of-the-art approaches. Furthermore, the ablation studies clearly demonstrate the effectiveness of "Seek Common" and "Shelve Difference".

For noisy label problem, where a clean validation set is assumed not available, it is hard to select hyperparameters, especially the epochs to stop training. Current works mainly compare methods with fixed epochs. However, in practice, we want to find a best model during training. Motivated by these, we propose an early-stopping method that can tell users when to stop the training and find a best model.

2. Related work

In this section we first introduce the "Memorization Effect" for DNNs, then explain the "Sample Selection" thought which is based on the "Memorization Effect". After that, we review works that use multiple nets for joint training.

Memorization Effect. It is studied in [2] that when deep learning models are trained on typical datasets with mostly correct labels, they do not memorize the data. So in the beginning of training, they learn the dominant patterns shared among the data samples. It has been conjectured that this behavior is due to the distributed and hierarchical representation inherent in the design of the deep learning models and the explicit regularization techniques that are commonly used when training them [2].

Sample Selection. An intuitive way to deal with noisy label is to select relative clean samples for models to update. We can treat images with wrong labels as images hard to learn since the correlation between labels and images are quite small. On the other hand, clean data has higher correlation between images and labels so DNN will learn their features faster. Considering Memorization Effect, DNN will learn noisy data slower resulting in lower speed for loss to decrease during training, so the loss for clean samples will be smaller than noisy samples at each epoch. Thus a promising method of handling noisy labels is to train models on small-loss instances[11][4] [8] [3] [14] [15].

Besides directly use small loss to do sample selection, some methods tend to fit a mixture model to distinguish noisy samples from clean samples. [1] fit a two-component Beta Mixture Model (BMM) to the max-normalized loss to model the distribution of clean and noisy samples. However, BMM tends to produce undesirable flat distributions and fails when the label noise is asymmetric. [6] improves it by fitting a two-component Gaussian Mixture Model (GMM) to losses using the Expectation-Maximization algorithm. However, the EM algorithm requires several iterations to converge, which need much longer time than directly using small loss selection. And this procedure can not obtain an accurate noise ratio compared with manually estimating noise ratio using a small portion of noisy training data, which is claimed to used in Co-teaching, JoCoR, and our proposed method.

Joint training. A number of studies have proposed methods involving joint training of more than one model. For example, [8] suggested simultaneously training two separate but identical networks with random initialization, and only updating the network parameters when the predictions of the two networks differed. This idea was developed into co-teaching[3], whereby the two networks identified label-noise-free samples in their mini-batches and shared the update information with the other network. Co-teaching was further improved in [15], where the authors suggested to focus the training on data samples with lower loss values in order to reduce the risk of training on data with incorrect labels. Along the same lines, [14] propose to train two classifiers simultaneously with one joint loss, which is composed of regular supervised part and Co-Regularized part.

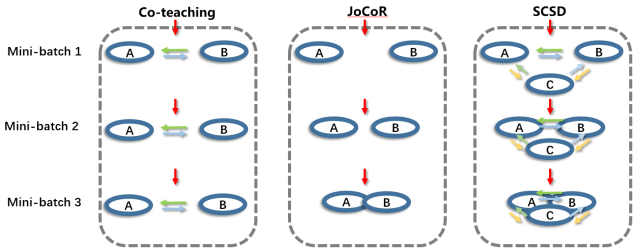


Figure 1. Comparision of SCSD and related works.

3. The Proposed Approach

In this section, we first introduce the motivation for SCSD (Section 3.1), then dive into the details of SCSD (Section 3.2), in which we describe three main ideas in SCSD: Joint selection, Cross Update and Co-Regularization. After that, we compare SCSD with other existed works (Section3.3). In section 3.4, we propose a new way to do early stopping to further improve our method's practicability.

3.1. Motivation

As mentioned in related work, current methods mainly use two networks to do sample selection. Co-teaching[3] uses a "cross updates" methods to reduce error propagation.

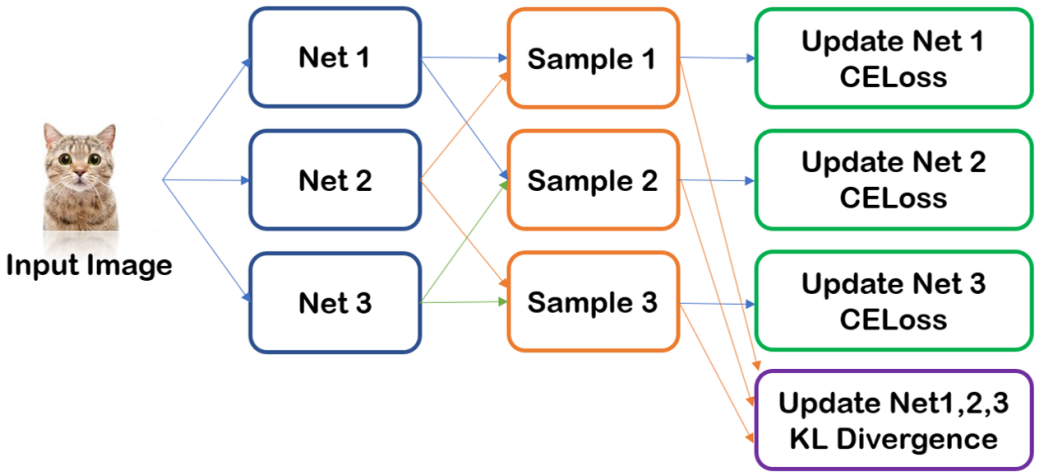


Figure 2. Framework of our proposed SCSD

JoCoR[14] improves co-teaching by introducing "agreement" of networks. In two networks' setting, it is hard to maintain both the benefits of cross updates and agreement. However, we can utilize both by adding a new network, that is a tri-net setting. In which, every two nets seek agreement samples to update another net, a joint co-regularization term for three nets are added to enforce converge. In this way, every net is fed with different samples, which prevents error flow to directly goes into next iteration. Also, the agreement of different nets is used to make each selection more clean and co-regularization term enforces networks to be agreed on those highly agreed samples (which are assumed to be clean).

3.2. SCSD

Before explaining our methods, we define annotations that will be used later. For multi-class classification with M classes, we suppose the dataset with N samples is given as $D = \{\boldsymbol{x}_i, y_i\}_{i=1}^{N}$, where $\boldsymbol{x}_i$ is the i -th instance with its observed label as $y_i \in \{1, \ldots, M\}$. We formulate the proposed SCSD approach with three deep neural networks denoted by $f(\boldsymbol{x}, \boldsymbol{\Theta}_1)$, $f(\boldsymbol{x}, \boldsymbol{\Theta}_2)$, and $f(\boldsymbol{x}, \boldsymbol{\Theta}_3)$, while $\boldsymbol{p}_1 = \left[p_1^1, p_1^2, \ldots, p_1^M\right]$, $\boldsymbol{p}_2 = \left[p_2^1, p_2^2, \ldots, p_2^M\right]$ and $\boldsymbol{p}_3 = \left[p_3^1, p_3^2, \ldots, p_3^M\right]$ denote their prediction probabilities of instance x_i, respectively. In other words, $\boldsymbol{p}_1$, $\boldsymbol{p}_2$ and $\boldsymbol{p}_3$ are the outputs of the "softmax" layer in Θ_1 , Θ_2 and Θ_3

Next, we will describe SCSD in details. The whole structure can be seen in Figure 2

3.2.1 Joint selection

We adopt small-loss selection as mentioned in the related work. For details, we update $R(t)$, which controls how many small-loss data should be selected in each training epoch. At the beginning of training, we keep more small-loss data (with a large $R(t)$) in each mini-batch since deep networks would fit clean data first. With the increase of epochs, we reduce $R(t)$ gradually until reaching $1-\tau$, keeping fewer examples in each mini-batch.

Besides small loss, the agreement of models can also help us find cleaner samples. Following JoCoR, we select clean samples using a joint loss that considers both classification loss and contrastive loss. For example, the joint loss for network 1 and network 2 is defined as:

$$l^{1,2}(\mathbf{x_i}) = (1-\lambda) * l_{sup}^{1,2}(\mathbf{x_i}, y_i) + \lambda * l_{con}^{1,2}(\mathbf{x_i}) \tag{1}$$

where classification loss is the sum of two network's Cross Entropy loss:

$$\begin{aligned} \ell_{\text{sup}}^{1,2}\left(\boldsymbol{x}_i, y_i\right) &= \ell_{\text{Cl}}\left(\boldsymbol{x}_i, y_i\right) + \ell_{\text{C2}}\left(\boldsymbol{x}_i, y_i\right) \\ &= -\sum_{i=1}^{N}\sum_{m=1}^{M} y_i \log\left(p_1^m\left(\boldsymbol{x}_i\right)\right) \\ &\quad -\sum_{i=1}^{N}\sum_{m=1}^{M} y_i \log\left(p_2^m\left(\boldsymbol{x}_i\right)\right) \end{aligned} \tag{2}$$

The contrastive loss is used to evaluate the match of different networks' predictions p_1, p_2. To simplify implementation, we could use the symmetric Kullback-Leibler(KL) Divergence to surrogate this term.

$$\begin{aligned} \ell_{\text{con}} &= D_{\text{KL}}\left(\boldsymbol{p}_1 \| \boldsymbol{p}_2\right) + D_{\text{KL}}\left(\boldsymbol{p}_2 \| \boldsymbol{p}_1\right) \\ &= \sum_{i=1}^{N}\sum_{m=1}^{M} p_1^m\left(\boldsymbol{x}_i\right) \log \frac{p_1^m\left(\boldsymbol{x}_i\right)}{p_2^m\left(\boldsymbol{x}_i\right)} \\ &\quad + \sum_{i=1}^{N}\sum_{m=1}^{M} p_2^m\left(\boldsymbol{x}_i\right) \log \frac{p_2^m\left(\boldsymbol{x}_i\right)}{p_1^m\left(\boldsymbol{x}_i\right)} \end{aligned} \tag{3}$$

The joint loss of network 1 and network 2 is then used to select a subset of small-joint-loss samples:

$$\tilde{D}_n^{1,2} = \arg\min_{D'_n:|D'_n|\geq R(t)|D_n|} \ell^{1,2}\left(D'_n\right) \tag{4}$$

Similarly, we can get another two small-joint-loss sample sets by computing the joint loss for the other two pairs.

$$\tilde{D}_n^{2,3} = \arg\min_{D'_n:|D'_n|\geq R(t)|D_n|} \ell^{2,3}\left(D'_n\right) \tag{5}$$

$$\tilde{D}_n^{1,3} = \arg\min_{D'_n:|D'_n|\geq R(t)|D_n|} \ell^{1,3}\left(D'_n\right) \tag{6}$$

where $\ell^{2,3}(\mathbf{x_i})$ and $\ell^{1,3}(\mathbf{x_i})$ are defined similarly as $\ell^{1,2}(\mathbf{x_i})$.

3.2.2 Cross Update

Motivated by co-teaching, we use cross update of networks to prevent error propagate. Each network is updated using the data selected by the joint loss of the other two networks.

$$\theta_1 = \theta_1 - \eta\nabla\ell\left(\theta_1, \tilde{D}^{2,3}\right) \tag{7}$$

$$\theta_2 = \theta_2 - \eta\nabla\ell\left(\theta_2, \tilde{D}^{1,3}\right) \tag{8}$$

$$\theta_3 = \theta_3 - \eta\nabla\ell\left(\theta_3, \tilde{D}^{1,2}\right) \tag{9}$$

3.2.3 Co-Regularization

After using joint selection to select relative clean samples and using these samples to cross update networks, we now introduce our loss function, which contains both a normal classification loss and a regularization term. For example, the loss for network 1 is defined as:

$$\ell(\theta_1, \tilde{D}^{2,3}) = (1-\lambda)*l_{sup}(\theta_1, \tilde{D}^{2,3}) + \lambda*l_{con}(\tilde{D}^{2,3}) \tag{10}$$

The classification loss is the sum of selected data's cross entropy loss.

$$l_{sup}(\theta_1, \tilde{D}^{2,3}) = -\sum_{\tilde{D}^{2,3}}\sum_{m=1}^{M} y_i \log\left(p_1^m\left(\boldsymbol{x}_i\right)\right) \tag{11}$$

The regularization term is added to enforce three networks reaching agreement on the selected samples. And is implemented as the sum of three symmetirc KL Divergence.

$$\begin{aligned} l_{con}(\tilde{D}^{2,3}) &= \sum_{k,l=1,2,3} D_{KL}(\mathbf{p_k}||\mathbf{p_l}) \\ &= \sum_{k,l=1,2,3}\sum_{\tilde{D}^{2,3}}\sum_{m=1}^{M} p_1^m\left(\boldsymbol{x}_i\right)\log\frac{p_1^m\left(\boldsymbol{x}_i\right)}{p_2^m\left(\boldsymbol{x}_i\right)} \end{aligned} \tag{12}$$

Similarly, loss for network 2 and network 3 are given below:

$$\ell(\theta_2, \tilde{D}^{1,3}) = (1-\lambda)*l_{sup}(\theta_2, \tilde{D}^{1,3}) + \lambda*l_{con}(\tilde{D}^{1,3}) \tag{13}$$

$$\ell(\theta_3, \tilde{D}^{1,2}) = (1-\lambda)*l_{sup}(\theta_3, \tilde{D}^{1,2}) + \lambda*l_{con}(\tilde{D}^{1,2}) \tag{14}$$

	Co-teaching	JoCoR	SCSD
joint selection	✗	✓	✓
cross update	✓	✗	✓
co-regularization	✗	✓	✓

Table 1. Comparison of state-of-the-art and related techniques with our SCSD approach.

3.2.4 Relation to other approaches

We compare SCSD with other related approaches in Table 1 and Figure 1. The table mainly compare the differences of methods and the figure mainly compare the differences of structures. Specifically, Co-teaching updates parameters of networks by the "cross update" strategy to reduce the accumulated error flow. JoCoR use two networks' agreement to joint select samples and add co-regularization to joint training the two networks. Our SCSD combines all of these three techniques with the compromise of adding a net. Every two networks joint select clean samples and use it to cross update the other network. A co-regularization term is also added to enforce converge of the three.

3.2.5 Early stopping

Motivation. Current works in learning with noisy labels mainly focus on model's robust training and focus on comparing methods with fixed epochs. However, in practice, we want to find a best model during training. And in a real world problem, the epoch to be trained is a hyperparameter, which is hard to determined without clean validation set. Motivated by these, we propose an early-stopping method that can tell users when to stop the training and find a best model.

Method. Based on "Memorization Effect" and "Small Loss Tricks", we can divide all samples into two sets: a small loss set and a large loss set. The small loss set is assumed to contain mostly clean samples and the large loss set is assumed to contain mostly noisy samples. At the beginning of training, models prefer to learn clean samples and have not overfit to noise. So the accuracy of clean samples will increase, while the accuracy of noisy samples will remain low, or slightly increase. The accuracy gap between clean samples and noisy samples will increase. When network begins to overfit noise, the accuracy for noisy samples will increase. And the clean samples' accuracy will remains almost unchanged (since models have learn them well). So, the accuracy gap will decrease.

If we plot the accuracy gap during training with noisy labels, it will first increase then decrease. After the highest point, the benefit for clean samples will be suppressed by the damage of noisy labels. So, the highest point is where we want to stop our network from continue training.

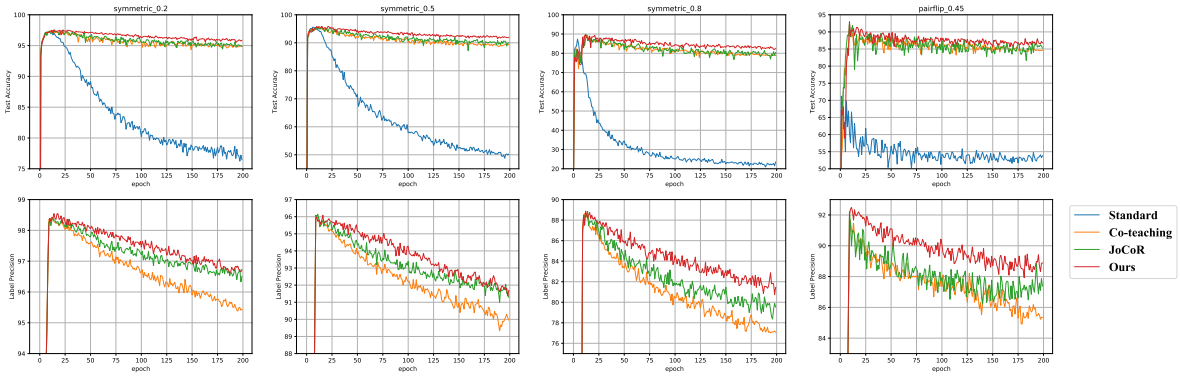


Figure 3. Results on MNIST dataset.T op: test accuracy(%) vs. epochs; bottom: label precision(%) vs. epochs.

Table 2. Average test accuracy (%) on MNIST over the last 10 epochs.

	Standard	Co-teaching	JoCoR	Ours
Symmetric 0.2	77.24	94.86	95.04	**95.83**
Symmetric 0.5	49.98	89.20	90.01	**91.95**
Symmetric 0.8	22.40	78.89	79.23	**82.69**
Pairflip 0.45	53.43	84.68	85.79	**86.88**

4. Experiments for Robust Training

In this section, we first check the robustness of SCSD by comparing it with some state-of-the-art approaches, then analyze the impact of seeking common and shelving differences by ablation study.

4.1. Experiment setup

Datasets. We verify the effectiveness of our approach on three benchmark datasets: MNIST, CIFAR-10 and CIFAR-100. The information of these datasets are summarized in 5 These datasets are popularly used for the evaluation of learning with noisy labels in previous literature[5][10].

Since all datasets are clean, following [9][10] we need to corrupt these datasets manually by the label transition matrix Q, where $Q_{ij} = Pr(\hat{y} = j|y = i)$ given that noisy $\hat{y}$ is flipped from clean y. Assume that the matrix Q has two representative structures: (1) Symmetry flipping; (2) Pair flipping: a simulation of fine-grained classification with noisy labels, where labellers may make mistakes only within very similar classes. Figure 6 shows an example of noise transition matrix

Baselines. We compare SCSD with the following state-of-art algorithms, and implement all methods with default parameters by PyTorch, and conduct all the experiments on a GeForce RTX 2080 Ti GPU.

(i) Co-teaching [3], which trains two networks simultaneously and cross-updates parameters of peer networks.

(ii) JoCoR [14], which trains two deep neural networks and consists of agreement-update step and Co-Regularization.

(iii) As a simple baseline, we compare SCSD with the standard deep network that directly trains on noisy datasets (abbreviated as Standard).

Network Structure. For MNIST, we use a 2-layer MLP. For CIFAR-10, we use a network architecture with 2 convolutional layers and 3 fully connected layers. For CIFAR-100, the 7-layer network architecture in our paper follows [13]. The network structures are summarized in 6

Optimizer. Adam optimizer (momentum=0.9) is with an initial learning rate of 0.001, and the batch size is set to 128 and we run 200 epochs. The learning rate is linearly decayed to zero from 80 to 200 epochs. As deep networks are highly nonconvex, even with the same network and optimization method, different initializations can lead to different local optimal. Thus, following [8], we also take two networks with the same architecture but different initializations as two classifiers.

Initialization. Assume that the noise rate τ is known. To conduct a fair comparison in benchmark datasets, we set the ratio of small-loss samples $\lambda(e)$ as identical as Co-teaching:

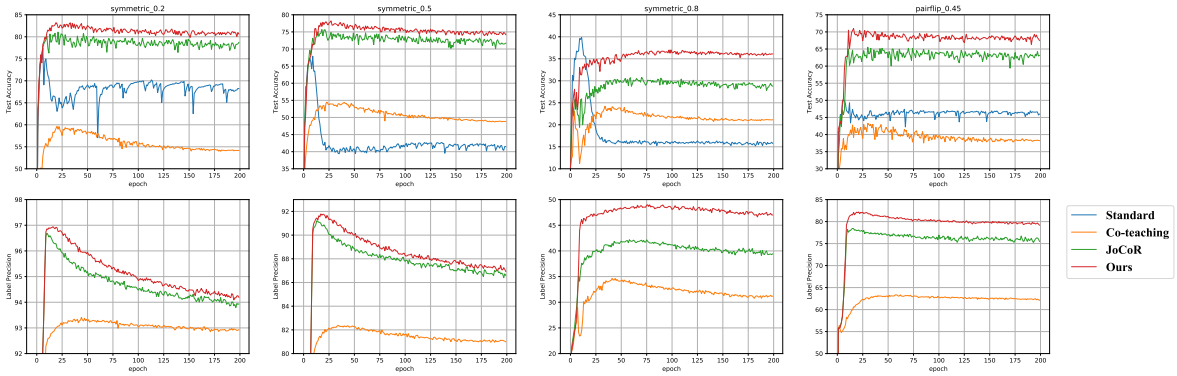


Figure 4. Results on CIFAR-10 dataset.T op: test accuracy(%) vs. epochs; bottom: label precision(%) vs. epochs.

Table 3. Average test accuracy (%) on CIFAR-10 over the last 10 epochs.

	Standard	Co-teaching	JoCoR	Ours
Symmetric 0.2	67.91	54.18	78.05	**80.79**
Symmetric 0.5	41.30	48.83	71.78	**74.49**
Symmetric 0.8	15.83	21.13	28.80	**36.02**
Pairflip 0.45	46.46	38.26	63.16	**68.21**

$$\lambda(e) = 1 - \min\{\frac{e}{E_k}\tau, \tau\}, \tag{15}$$

where $E_k = 10$. If τ is not known in advanced, τ can be inferred manually using a small portion of noisy training dataset[7][16]. Note that $\lambda(e)$ only depends on the memorization effect of deep networks but not any specific datasets.

As for λ in our loss function [Eq.10], we keep it as 0.1 for all experiments settings for fairness, since our setting doesn't requires an additional clean validation set to tune all parameters.

Measurement. To measure the performance, we use the test accuracy, i.e., *test accuracy = (# of correct predictions) / (# of test).* Intuitively, higher test accuracy means that the algorithm is more robust to the label noise. Besides, we also use the label precision in each mini-batch, i.e., *label precision = (# of clean labels) / (# of all selected labels).* Specifically, we sample $R(t)$ of small-loss instances in each mini-batch and then calculate the ratio of clean labels in the small-loss instances. Intuitively, higher label precision means less noisy instances in the mini-batch after sample selection, so the algorithm with higher label precision is also more robust to the label noise.

4.2. Comparision with the State-of-the-Arts

Results on MNIST. At the top of Figure 3, it shows test accuracy vs. epochs on MNIST. In all four plots, we can see the memorization effect of networks, i.e., test accuracy of Standard first reaches a very high level and then gradually decreases. Thus, a good robust training method should stop or alleviate the decreasing process. On this point, SCSD consistently achieves higher accuracy than all the other baselines in all four cases.

We can compare the test accuracy of different algorithms in detail in Table 2. All new approaches work better than Standard obviously, which demonstrates their robustness. Among them, SCSD works significantly better than other methods, especially in high noise settings.

To explain such excellent performance, we plot label precision vs. epochs at the bottom of Figure 3. Only Co-teaching, JoCoR, and SCSD are considered here, as they include sample selection during training. Note that SCSD reaches high label precision in all four cases and gives far more label precision than other methods in symmetric-0.8 and pairflip 0.45.

Results on CIFAR-10. Table 3 shows test accuracy on CIFAR-10. As we can see, SCSD performs the best in all four cases again. Note that in symmetric 0.2 and pairflip 0.45 co-teaching performs lower than standard method, which demonstrate the limits of the method. By contrast, JoCoR and SCSD are more general. In Symmetric 0.8 and Pairflip 0.45, our method improves JoCoR by 3% – 4%, which demonstrate SCSD is much more robust in extremely noise settings.

Figure 4 shows test accuracy and label precision vs.

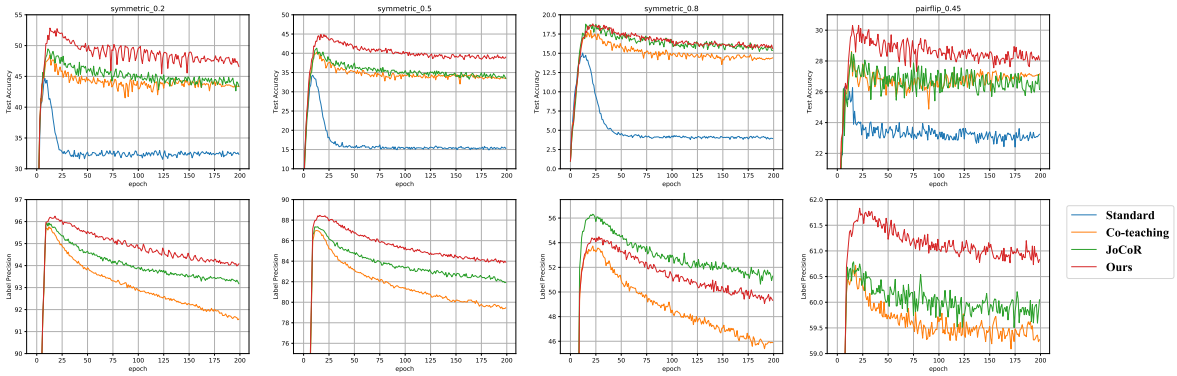


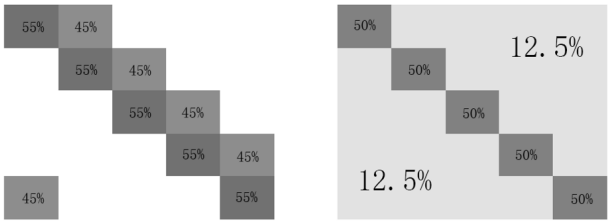
Figure 5. Results on CIFAR-100 dataset.T op: test accuracy(%) vs. epochs; bottom: label precision(%) vs. epochs.

Table 4. Average test accuracy (%) on CIFAR-100 over the last 10 epochs.

	Standard	Co-teaching	JoCoR	Ours
Symmetric 0.2	32.48	43.4	43.88	**47.29**
Symmetric 0.5	15.39	33.52	33.87	**38.97**
Symmetric 0.8	4.0	14.29	15.63	**15.91**
Pairflip 0.45	23.14	27.07	26.41	**28.25**

Table 5. Summary of data sets used in the experiments.

	# of train	# of test	# of class	size
MNIST	60,000	10,000	10	28 x 28
CIFAR-10	50,000	10,000	10	32 x 32
CIFAR-100	50,000	10,000	100	32 x 32



(a) Pair ($\epsilon = 45\%$). (b) Symmetry ($\epsilon = 50\%$).

Figure 6. Transition matrices of different noise types (using 5 classes as an example).

epochs. SCSD outperforms all the other comparing approaches on both test accuracy and label precision. On label precision, an interesting phenomenon is that in the Symmetry-80% case SCSD continues increasing after JoCoR begins to decrease and consistently outperforms it in all the later epochs. The result shows that SCSD has better generalization ability than JoCoR.

Results on CIFAR-100. Then, we show our results on CIFAR-100. The test accuracy is shown in Table 4. Test accuracy and label precision vs. epochs are shown in Figure 5. Note that there are only 10 classes in MNIST and CIFAR-10 datasets, but 100 classes in CIFAR-100 dataset. Thus, overall the accuracy is much lower than previous ones in Tables 2 and 3. But SCSD still achieves high test accuracy on this dataset. In the easiest Symmetry 0.2 and Symmetry 0.5 cases, SCSD works significantly better than Co-teaching, JoCoR. In the hardest Symmetry-80% case, SCSD and JoCoR tie together but SCSD still gets higher testing accuracy. When it turns to pairflip 0.45 case, SCSD performs much better than other methods.

4.3. Ablation Study

To conduct ablation study for analyzing the effect of seeking common and shelving differences, we propose another two algorithms, the diferences of them are summarized in Table 7.

Seeking Common Only (SCO). This algorithm takes the same idea as JoCoR except that three networks instead of two are used. More precisely, three networks first jointly select a small-loss subset according to equation 1, where classification loss is now a sum of three networks' loss.

$$l_{sup}(\mathbf{x_i}, y_i) = l_{C1}(\mathbf{x_i}, y_i) + l_{C2}(\mathbf{x_i}, y_i) + l_{C3}(\mathbf{x_i}, y_i) \quad (16)$$

The contrastive loss is the sum of three symmetric Kullback-Leibler(KL) Divergence.

$$l_{con} = \sum_{i=1,2,3,j=1,2,3} D_{KL}(\mathbf{p_i}||\mathbf{p_j}) \quad (17)$$

Table 6. MLP and CNN models used in our experiments on MNIST, CIFAR-10, CIFAR-100

MLP on MNIST	CNN on CIFAR-10	CNN on CIFAR-100
28 X 28 Gray Image	32 x 32 RGB Image	32 x 32 RGB Image
Dense 28 x 28 ->256, ReLU	5 x 5 Conv, 6 ReLU 2 X 2 Max-pool	3 x 3 Conv, 64 BN, ReLU 3 X 3 Conv, 64 BN, ReLU 2 x 2 Max-pool
	5 x 5 Conv, 16 ReLU 2 x 2 Max-pool	3 x 3 Conv, 128 BN, ReLU 3 x 3 Conv, 128 BN, ReLU 2 x 2 Max-pool
	Dense 16 x 5 x 5 ->160, ReLU Dense 120 ->84, ReLU	3 x 3 Conv, 196 BN, ReLU 3 x 3 Conv, 196 BN, ReLU 2 x 2 Max-pool
Dense 256 ->10	Dense 84 ->10	Dense 256 ->100/10

Table 7. Ablation study of Seek Common and Shelving Difference

	SCO	SDO	SCSD
joint selection	✓	✓	✓
cross update	✗	✓	✓
co-regularization	✓	✗	✓

Table 8. Ablation study of average test accuracy (%) on CIFAR-10 over the last 10 epochs.

	SCO	SDO	SCSD
Symmetric 0.2	79.14	78.11	**80.79**
Symmetric 0.5	73.04	72.02	**74.49**
Symmetric 0.8	33.27	31.78	**36.02**
Pairflip 0.45	64.52	63.97	**68.21**

According to the joint loss, we can select a small loss samples, and joint update on these selected samples. Note that the key difference here is without the use of "cross update".

Shelving Difference Only (SDO). This algorithm combines joint selection and cross update but without adding co-regularization term in Equation 10.

We show the test accuracy of CIFAR-10 in Table 8. As we can see, SCSD performs much better than the others on all noise settings. This observation indicates the combination of "seeking common" and "shelving difference" help DNNs achieves better performance than any one of them. The results on MNIST and CIFAR-100 are similar, so we do not show here.

5. Experiments for Early stopping

Keep the same experiments setting as before, we conduct early stopping experiments on SCSD to see whether we can stop DNNs when it start to overfit. In Figure 7, we plot the test accuracy and training accuracy gap on CIFAR-100. As we can see, the trend of the two curves are similar. Accuracy gap reaches largest almost as the same time as test accuracy reaches largest. This proves that we can use accuracy gap to guide when to do early stopping. In Table 9, we summarize the test accuracy of using early stopping and without using early stopping (last ten epochs' average accuracy). We can see a large improvement of using early stopping. We also compare our ealry stopping results with the ideally best test accuracy. Note that in Symmetric 0.2, our early stopping result is almost the same as best accuracy. In high noise setting, early stopping gives a relatively good reference for users to determine when to update.

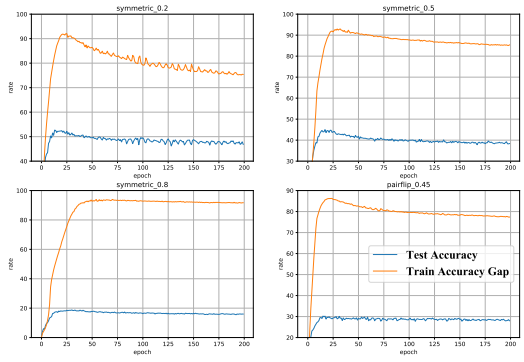


Figure 7. Results of early stopping on CIFAR-100 dataset.

Table 9. Test accuracy on CIFAR-100 with early stopping.

	Max Acc	Early Stop	Last Avg
Symmetric 0.2	52.67	52.20	47.34
Symmetric 0.5	45.01	42.74	38.60
Symmetric 0.8	18.76	17.15	15.90
Pairflip 0.45	30.22	28.89	28.28

6. Conclusion

The paper proposes an effective approach called SCSD to improve the robustness of deep neural networks with noisy labels. The key idea of SCSD is to train three classifiers simultaneously and encourage them to both seeking common and shelving difference. Similar to JoCoR, we select small-loss instances to update networks in each mini-batch data by the joint loss and cross trains three networks like Co-teaching. We conduct experiments on MNIST, CIFAR-10, CIFAR-100 to demonstrate that, SCSD can train deep models robustly with the slightly and extremely noisy supervision. Furthermore, the ablation studies clearly demonstrate the effectiveness of "Seek Common" and "Shelving Difference". An early-stopping algorithm based on the accuracy gap between small loss samples and large loss samples are then proposed to improve our methods' practicability, and experiments are done to show its feasibility.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction, 2019. 2

[2] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks, 2017. 1, 2

[3] 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. 2018. 1, 2, 5

[4] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels, 2018. 1, 2

[5] Ryuichi Kiryo, Gang Niu, Marthinus C. du Plessis, and Masashi Sugiyama. Positive-unlabeled learning with non-negative risk estimator, 2017. 5

[6] Junnan Li, Richard Socher, and Steven C. H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning, 2020. 2

[7] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(3):447–461, Mar 2016. 1, 6

[8] Eran Malach and Shai Shalev-Shwartz. Decoupling "when to update" from "how to update", 2018. 2, 5

[9] Giorgio Patrini, Alessandro Rozza, Aditya Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: a loss correction approach, 2017. 5

[10] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping, 2015. 5

[11] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning, 2019. 1, 2

[12] Tyler Sanderson and Clayton Scott. Class proportion estimation with application to multiclass anomaly rejection, 2014. 1

[13] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels, 2018. 5

[14] H. Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13723–13732, 2020. 1, 2, 3, 5

[15] Xingrui Yu, B. Han, J. Yao, Gang Niu, I. Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *ICML*, 2019. 2

[16] Xiyu Yu, Tongliang Liu, Mingming Gong, Kayhan Batmanghelich, and Dacheng Tao. An efficient and provable approach for mixture proportion estimation using linear independence assumption. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 6