A Semi-Supervised Two-Stage Approach to Learning from Noisy Labels

Yifan Ding[†] Liqiang Wang[†] Deliang Fan[‡] Boqing Gong
[†]Department of Computer Science, University of Central Florida Tencent AI Lab
[‡]Department of Electrical Engineering, University of Central Florida Bellevue, WA 98004
yf.ding@knights.ucf.edu, lwang@cs.ucf.edu, dfan@ucf.edu boqinggo@outlook.com

Abstract

The recent success of deep neural networks is powered in part by large-scale well-labeled training data. However, it is a daunting task to laboriously annotate an ImageNet-like dateset. On the contrary, it is fairly convenient, fast, and cheap to collect training images from the Web along with their noisy labels. This signifies the need of alternative approaches to training deep neural networks using such noisy labels. Existing methods tackling this problem either try to identify and correct the wrong labels or reweigh the data terms in the loss function according to the inferred noisy rates. Both strategies inevitably incur errors for some of the data points. In this paper, we contend that it is actually better to ignore the labels of some of the data points than to keep them if the labels are incorrect, especially when the noisy rate is high. After all, the wrong labels could mislead a neural network to a bad local optimum. We suggest a two-stage framework for the learning from noisy labels. In the first stage, we identify a small portion of images from the noisy training set of which the labels are correct with a high probability. The noisy labels of the other images are ignored. In the second stage, we train a deep neural network in a semi-supervised manner. This framework effectively takes advantage of the whole training set and yet only a portion of its labels that are most likely correct. Experiments on three datasets verify the effectiveness of our approach especially when the noisy rate is high.

1. Introduction

With the recent development of deep neural networks, we have witnessed great advancements in visual recognition tasks such as image classification [42, 50, 43, 45], object detection [7, 54, 44, 13], and semantic segmentation [28, 5, 15, 8]. Take the famed object recognition challenge ILSVRC [42] for instance, the Inception-ResNet-v2 [49] achieves a remarkable top-5 accuracy of 95.3% in 2017. The success of the deep neural networks is powered in part by large-scale well-labeled training data. However, it is actually a very daunting task to laboriously annotate an ImageNet-like training set.

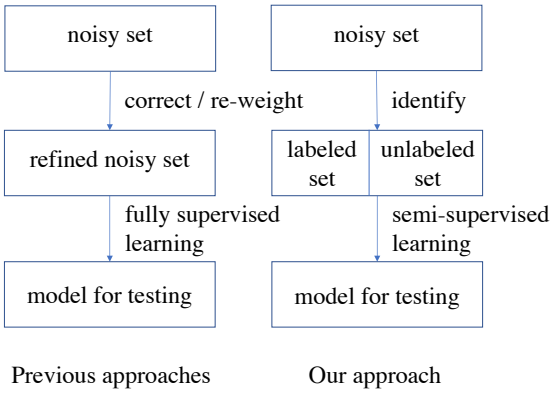

Figure 1. Comparison between our framework and the one of existing methods for learning from noisy labels. Instead of trying to correct or reweigh all the labels of the noisy data, we ignore the ambiguous ones and train a deep neural network in a semi-supervised fashion.

On the contrary, it is fairly convenient, fast, and cheap to collect training images from the Web along with their noisy labels. This signifies the need of alternative approaches to train deep neural networks using such noisy labels. Indeed, the very first source of training data is often from the Web when we face a new visual recognition task. Therefore, the methods that effectively learn from the noisy labels can significantly reduce the human labeling efforts, even to zero effort in some scenarios.

There has been a rich line of recent works that aim to address the problem of learning from noisy labels. We categorize them to two groups: one directly learns from the noisy labels and the other relies on an extra set of clean data. For the former, a label cleansing module is often applied in order to identify the correctly labeled data [4, 11, 3]. Alternatively, one may model the noise to reweigh the data terms in the loss functions [27, 36, 34]. The performance of these algorithms heavily depend on the precision of the label

cleansing or the estimated noisy rates. They perform well when the noise rates can be safely managed [29, 46, 34, 36], but could suffer from the ambiguity between mislabeled examples and "hard cases" — the data points whose labels are correct but hard to be captured by the neural networks' classification boundary.

For the second group of methods, an extra set of clean data is used to guide the learning agent through the noisy data. Li et al. [26] enforce the network trained from the noisy data to imitate the behavior of another network learned from the clean set. Vahdat [55] constructs an undirected graphical model to represent the relationship between the clean and noisy data. Veit et al. [58] also use a secondary network to clean the labels of the noisy data such that the main network receives more accurate supervision than from the original training set. An absence of the clean set would prevent these methods from being applied to some situations. Besides, like the approaches in the first group, they still aim to correct the labels of the noisy set and could make mistakes in this procedure.

Despite their promising results, both groups of the existing methods come with a common caveat — they attempt to correct the noisy labels or reweigh the terms of all the data points no matter how difficult it is to do so. This inevitably incurs errors for some of the data points. In this paper, we contend that it is better to completely ignore the labels of some of the data points than to keep their wrong labels. After all, the wrong labels could mislead the training procedure of the network. We suggest a two-stage framework for the learning from noisy labels. In the first stage, we identify a small portion of data points from the noisy training set of which the labels are correct with a high probability. The noisy labels of the other data points are then removed. In the second stage, we train a deep neural network in a semi-supervised manner. This framework effectively takes advantage of the whole training set and yet only a portion of its labels which is most likely correct. Figure 1 contrasts our framework to the one taken by most existing methods.

It is worth noting that the first stage of our framework can be implemented in a variety of ways. Many existing methods for learning from the noisy labels are actually applicable. This paper presents our preliminary study by a simple and efficient self-refining method for the first stage and leaves the exploration of more sophisticated approaches to the future work. In particular, we rank all the data points within each class and then keep the labels of the top few. The ranking is performed by the multi-way classification neural network learned from the original training set when there is no clean set available, and by a binary classifier of each class which is trained to differentiate the data of clean and noisy labels when the clean set is given. In the second stage, we apply the temporal ensembling [23] to train a deep neural network in the semi-supervised manner. To the best of our knowledge, this is the first time that the temporal ensembling is tested on a large set of natural images — around 1M images of which the resolutions are about 256x256.

It is also worth noting that when there exists a small clean set in addition to the noisy training set, the semi-supervised learning approach [25] has been considered a baseline in the experiments of [59]. In a sharp contrast to the observations of [37], we show that, under our two-stage framework, semi-supervised learning methods can actually give rise to state-of-the-art results for the task of learning from noisy labels.

The rest of this paper is organized as follows. Section 2 discusses several related areas to our method. In Section 3, our two-stage approach is described in details. Section 4 shows the experimental results of our approach in two diminutive datasets and a large-scale real noisy dataset. Finally, we conclude the paper in Section 5.

2. Related work

Our approach is broadly related to four research topics: learning from noisy labels, the robustness of neural networks, semi-supervised learning, and Webly-supervised learning. We discuss each of them as below.

Learning from noisy labels: The purpose of learning from noisy labels is to deal with noisy labels in the training data and reduce its negative influence toward the accuracy of classifiers. Following the review in [10], the algorithms of learning from noisy labels can be grouped into three clusters: noise-robust approaches [52, 53] which depend on the robustness of neural networks and do not really deal with noise, label noise-tolerant methods which usually make use of some side information like the noisy rate in each class to design models that account for the label noise [19, 37, 20, 26, 40], and label noise cleansing methods. In the third category, different approaches are proposed to either remove or correct the noisy labels. [48, 1] identify the mislabeled sample and reassign correct labels to them while [18, 33] delete possibly noisy samples. [31] removes and meanwhile corrects noisy labels of bio-informatics data sets. [47, 59] add an extra noisy layer to match the neural network outputs with the noisy label distribution. [38] proposes a noisy estimator using kernel mean embedding.

The robustness of neural networks: it is worth mentioning that the stochastic nature of the training algorithms of neural networks tolerates noisy labels by itself to some extent. [41] shows that deep neural networks are able to learn from the majority clean data while the gradient updates from noisy samples cancel out in each batch of training samples. [47] proves that a standard Convnet model [22] is surprisingly robust to label noise. [56] also finds that learning algorithms based on CNN features and part localization are robust to mislabeled training examples when the error rate is not too high. This ability makes it possible to refine

the noisy set or correct some labels by the network directly learned from the noisy labels.

Semi-supervised learning methods: Semi-supervised methods are proposed to learn in the presence of both labeled and unlabeled data [61], which is naturally applicable to the problems of learning from noisy labels when there is an extra clean set, e.g., by concealing the labels of the noisy set. Ladder network [39] is one of the methods that introduces lateral connections into an encoder-decoder network and it is trained to simultaneously minimize the sum of supervised and unsupervised cost functions by back-propagation in a layer-wise manner. Our work is mostly related to [23] which proposes a simpler but more efficient Π model and a temporal model which only minimizes the difference of two predicted probabilities of the same inputs accumulated in different epochs. More recent semi-supervised learning methods [32, 51, 35] use adversarial samples to regularize the networks. Some graph based methods propagate labels among the training data [60, 9]. Intermediate predictions by the model under training are used as pseudo labels [25] to reinforce the model. [2] uses a graph and labeled samples to push away mislabeled ones.

Webly-supervised learning methods: Webly-supervised learning methods specify the training data gathered from the web [59, 62, 11]. Usually, the gathered web data are companied with noise and have a very large scale; therefore, it is commonly impossible to know the exact noisy rate of each class. Under such a condition, estimation using a small portion of clean data [37] or noise modeling methods [25, 59] are often applied. While in our approach, we refrain from modeling the label noise and use semi-supervised learning instead to automatically propagate the labels of the mostly correct ones to the remaining training data points.

3. Approach

We describe our semi-supervised two-stage approach to learning from noisy labels in this section. It consists of two main components. In the first stage, we identify some data points from the noisy training set for which there exist strong indications that their labels are correct. In the second stage, a deep neural network is trained in a semi-supervised learning fashion over all the data points and yet using only the labels selected in the first stage. As a result, the network can be readily applied to classify previously unseen test data. As below we detail the implementations of the two components in this paper, and we stress that other realizations of the two-stage framework (cf. Figure 1) can be explored in the future.

3.1. Stage 1: identifying likely correct labels

The goal of this stage is to mine the data points for which the labels are likely correct from the noisy training set. The selected data and their labels will be the seed for the semi-supervised learning approach in the second stage. Additionally, it is important to note that we have to also construct a good validation set in order to monitor the training procedure and choose proper hyper-parameters in the next stage.

We pre-train a deep neural network classifier using the noisy data. The trained neural network is then used to mine the seed data points for the second stage. The first two panels of Figure 2 illustrate this procedure.

In particular, we examine the data points class by class. First of all, we remove the labels of the data points for which the neural networks' predictions differ from their original noisy labels. This corresponds to the prediction consistency module in the first panel of Figure 2. For the remaining images $\{x_i\}$ of a particular class c, we then rank them non-increasingly using the network's prediction $P(c|x_i)$. We keep the images for which the predicted probabilities are as high as 0.9. If less than 10% of the images of this class are kept after that, we add more images down the ranking list such that the labels of the top 10% of the images of that class pass the screening. We have also tested other percentages (e.g., 5%) and do not observe any significant change of the experimental results. Table 1 shows the results (accuracies) on CIFAR10 when we keep the labels of four different percentages of the training set.

Table 1. Performance of different percentages of labeled data points of the training set. (CIFAR-10, see Section 4 for detailed experimental setups).

labeled %	$p:0$	sy.$p:0.2$	asy.$p:0.2$	asy.$p:0.6$
5%	87.9	84.2	85.5	75.6
10%	88.0	84.5	85.6	75.8
20%	87.9	85.1	85.5	76.2
50%	88.1	84.5	85.6	74.8

When an extra clean set is available, more advanced techniques can actually be applied to mine the noisy set, e.g., using a graph between the clean and noisy sets [55] or a secondary neural network trained from the clean set [58]. We take an easy alternative and find it works well in the experiments. Specifically, we train a binary classifier for each class by assigning positive labels to the clean data of the class and negative labels to the data of other classes. After that, the classifier is used to classify the noisy data points of the corresponding class. We remove the labels of the images which are classified to the negative class. The second panel of Figure 2 demonstrates this procedure.

3.2. Stage 2: semi-supervised learning

Recall that we aim to lean a good classifier neural network that can perform well at the test stage, not to correct the labels of the training set at all. In Stage 2, we train the classifier in a semi-supervised way using all the data points

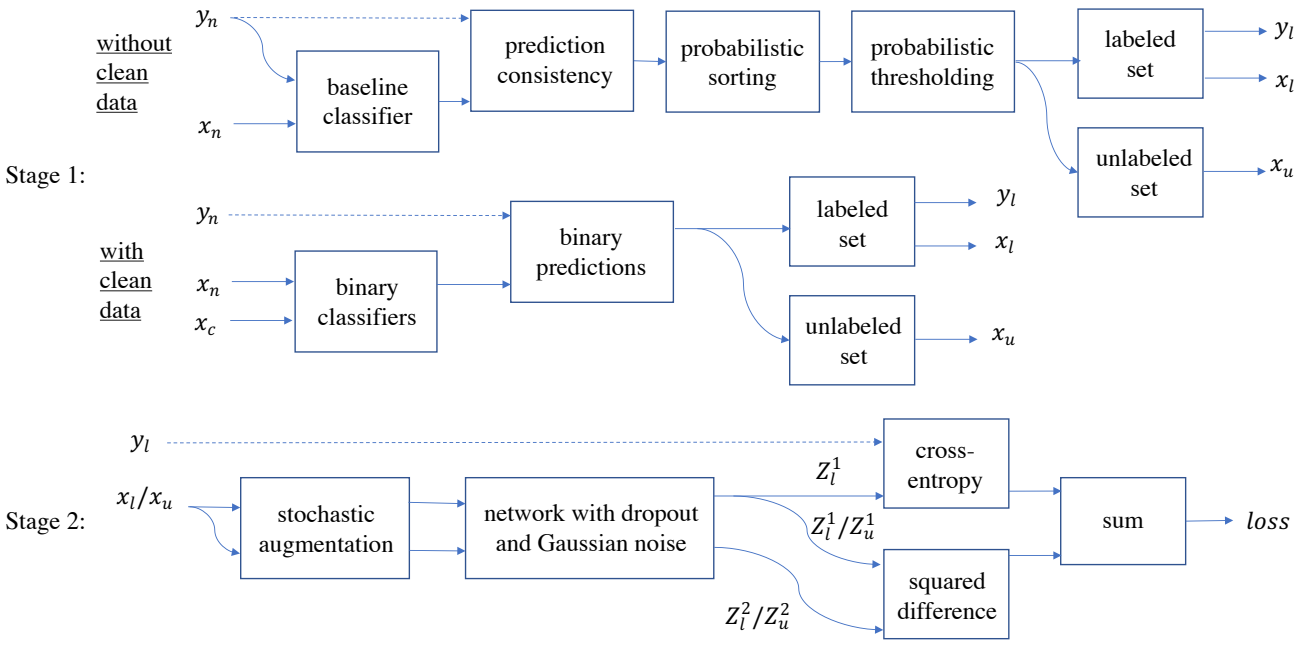

Figure 2. Overview of the two stages in our approach. In Stage 1, we mine some examples from the noisy training set such that their labels are more likely correct than the others'. After that, we employ a semi-supervised learning method in Stage 2 to train a deep neural network. We use the subscripts n and c to represent the noisy and clean data, respectively. The subscripts l and u respectively stand for the labeled and unlabeled data points that are identified by Stage 1 and then input to Stage 2.

Algorithm 1 The semi-supervised learning method used in Stage 2 of our method

Require: x_i = training stimuli
Require: y_i = labels for labeled data
Require: $|N|$ = number of samples in one mini batch
Require: $|N/2|$ = number of labeled data in one mini batch
Require: $f_\theta(x)$ = stochastic neural network with trainable parameters θ
Require: $g(x)$ = stochastic input augmentation function
for e in [1, numepochs] **do**
  for *each mini batch* B **do**
    $Z_i^1 \in B \leftarrow f_\theta(g(x_i \in B))$ evaluate network outputs for augmented inputs
    $Z_i^2 \in B \leftarrow f_\theta(g(x_i \in B))$ evaluate again the same inputs
    $loss \leftarrow -\frac{1}{N/2} \sum_{X_i \in X_l} \log Z_i[y_i]$ supervised loss component
    $+\frac{1}{N} \sum_{i=1}^{N} (||Z_i^1 - Z_i^2||_2)^2$ unsupervised loss component
    update θ using, e.g., SGD update network parameters
return θ

of the training set and yet keeping only the labels identified in Stage 1.

In particular, we use the methods proposed by [23] in our second stage. We improve the implementation of its Π model for the learning from large-scale natural images [59] whose labels are noisy, thanks to its efficiency. For the other two smaller datasets, we directly borrow the original temporal ensembling model from [23] whose performance is superior over the simplified Π model and yet has to be trained for many more epochs. We elaborate the Π method in this section and refer the readers to [23] for the details of the temporal ensembeling.

The main idea of the Π method is to regularize the network such that it generates about the same outputs for the same input image that undergoes data augmentation and/or dropout twice. This is a reasonable regularization because

the image labels are supposed to remain the same no matter how one augments the images.

We explain the main idea using the third panel of Figure 2 and Algorithm 1. Note that the subscripts are used to differentiate the labeled data points x_l and those with no labels x_u in the figure. Given an input image, no matter it is labeled or not, we apply some simple augmentations like horizontal flip and horizontal and vertical shift, and also add Gaussian noise. We further perturb the convolutional layers of the network by dropout. As a result, the same image x_i actually incurs different output vectors z_i^1 and z_i^2 by the softmax layer of the network. Accordingly, a consistency regularization can be defined to push them close to each other,

$$R = 1/N \sum_{i=1}^{N} ||z_i^1 - z_i^2||_2^2 \quad (1)$$

where N denotes the number of images in a mini-batch.

Therefore, the overall cost function for training the network is the following,

$$L = -1/M \sum_{j=1}^{M} \log z_j[y_j] + \alpha/N \sum_{i=1}^{N} ||z_i^1 - z_i^2||_2^2 \quad (2)$$

where the first term is the conventional cross-entropy loss over the M labeled data points in a mini-batch. The notation $z_j[y_j]$ is to index the y_j-th element in the output vector z_j and y_j is the label of the data point x_j.

Since there are far more unlabeled data points than the labeled ones after the pruning of Stage 1, we sample the labeled data more frequently than the unlabeled in order to provide the network effective gradients. For each mini-batch of size N, we randomly choose $N/2$ labeled images and $N/2$ unlabeled ones.

In the second stage, our semi-supervised learning of the network continues from the one trained in Stage 1. We find that the values of the cross-entropy term are in general much larger than the regularization term. Therefore, the balance cost α is introduced and its value is determined according to the model's performance on the validation set. Unlike the ramp-up and ramp-down balance cost used in [23], we fix α in the whole course of training for the ease of tuning on the large dataset. However, the ramp-up and ramp-down cost are used on the smaller MNIST and CIFAR-10 datasets.

3.3. Balancing different classes in the training

It is worth mentioning that we balance different classes in the training in our experiments because this simple and well-known trick gives rise to surprisingly large gains. By overly sampling the classes which have smaller number of samples than the other classes, we make the training set balanced across all classes. We notice that for the images crawled from the Web, the long-tailed distribution of different classes is a common case. One can easily retrieve thousands of images of a popular query and yet only a few for less interesting queries or rare classes. This inevitably influences even the well benchmarked datasets. For example, for the classes in ImageNet [42], a search of the keyword "Goose" returns a lot of results (and some of them are noisy). If we use the keyword "Egretta Garzetta", the results are very clean but there is only a small number of images.

4. Experiments

We run extensive experiments to evaluate the proposed two-stage and semi-supervised approach on both small-scale MNIST [24] and CIFAR-10 [21] and a large-scale benchmark of natural images, Cloting1M [59]. The results indicate that the two-stage method significantly outperforms the competing baselines when the noisy rate is high, and is comparable to the existing methods when the training set is contaminated by small noisy rates or zero noise.

4.1. Datasets

We test our method on three datasets: MNIST [24], CIFAR-10 [21], and Clothing1M [59]. MNIST is a dataset of handwritten digits. It has 60,000 training images and a test set of 10,000 image. Each image has 28x28 pixels. CIFAR-10 consists of 60,000 32x32 tiny images of real objects. Among them, 10,000 are left out as the test set. The Clothing1M dataset is about the same scale as ImageNet but with only 14 clothing classes. According to the estimation by the authors, about 39% of the labels in Clothing1M are incorrect because the dataset is automatically crawled by computer and has not been fully screened by human annotators. Nonetheless, for the research purpose, it does offer a clean test set of 22K images and a small extra clean set of 50K images that can be used in the training phase.

We follow the experiment setups in [37] to experiment with the three datasets. In particular, we add noise to the labels of the MNIST in the following way. Let $A \rightarrow_p B$ denote that the label of class A is changed to class B with probability p for any data point of class A (for simplicity, p is omitted in the discussion below). This artificially creates a noisy version of the MNIST. We change the labels following the paths $2 \rightarrow 7$, $3 \rightarrow 8$, $5 \rightarrow 6$, and $7 \rightarrow 1$. Similarly, we add noise to the labels of CIFAR-10 by TRUCK $\rightarrow$ AUTOMOBILE, BIRD $\rightarrow$ AIRPLANE, DEER $\rightarrow$ HORSE, CAT $\leftrightarrow$ DOG. When the noisy rate is $p = 0.6$, an example transition matrix of the labels is shown in Figure 3 for CIFAR-10. Both symmetric and asymmetric noises are tested in [37], and we experiment with both as well.

For the images in the CIFAR-10 and Clothing1M datasets, we subtract the per-pixel mean from each of them before sending them to the network. In the training, we flip the images and randomly crop 32x32 and 224x224 regions from the CIFAR10 and Clothing1M images, respectively.

$$\begin{bmatrix}
1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\
0. & 1. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\
0. & 0. & .4 & 0. & 0. & 0. & 0. & .6 & 0. & 0. \\
0. & 0. & 0. & .4 & 0. & 0. & 0. & 0. & .6 & 0. \\
0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. & 0. \\
0. & 0. & 0. & 0. & 0. & .4 & .6 & 0. & 0. & 0. \\
0. & 0. & 0. & 0. & 0. & .6 & .4 & 0. & 0. & 0. \\
0. & .6 & 0. & 0. & 0. & 0. & 0. & .4 & 0. & 0. \\
0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1. & 0. \\
0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 1.
\end{bmatrix}$$

Figure 3. An example of the label transition matrix for artificially adding noise to CIFAR-10.

4.2. Experiments on CIFAR-10 and MINIST

For all the experiments on CIFAR-10 and MNIST, we leave 10% out of the training set for validation and use the rest to train our neural networks. During the training, we decrease the learning rate by the ratio of 0.5 after the validation accuracy saturates for up to 10 epochs. We use the SGD optimizer for CIFAR-10 and AdaGrad for MNIST. The momentum is set to be 0.9, the initial learning rate is 0.1, and the minimal learning rate is set to be 1e-7. For all the baseline models, we use the cross-entropy loss as the objective function. We train our neural network for 40 epochs on MNIST and 120 epochs on CIFAR-10.

For CIFAR-10, we use a 14-layers ResNet for both the baseline model and the semi-supervised learning model. Specifically for the semi-supervised learning model, we add a Gaussian noise layer on the top of the model and dropout between two convolutional layers within the residual units to achieve the output difference for the unsupervised loss. For MNIST, we implement a fully connected network with two dense hidden layers of size 128 and dropout. Again, we add Gaussian noise layer for the semi-supervised learning model. Both networks follow the same architectures as in [37].

Qualitative results of Stage 1. In Figure 4, we provide some samples of the labeled and unlabeled images of different categories in CIFAR-10. They are the output of the Stage 1 and are the input to the semi-supervised learning approach in Stage 2. We can find that the labeled images are more typical and representative of their corresponding classes than the unlabeled ones. Their foregrounds also stand out more clearly from the background. These verify that Stage 1 works as we expected. It is able to mine the images whose labels are correct and, more importantly, remove the labels of the hard cases whose labels could be not only wrong but difficult to correct.

Quantitative results of Stage 1. Table 2 shows the percentages of the images with correct labels, incorrect labels, and no labels, respectively, of the CIFAR-10 training set before and after Stage 1. We can see that a much smaller percentage (16.75%) of images are labeled correctly after we apply Stage 1 than the percentage (70%) in the original noisy training set. Although this seems like an undesirable situation for the conventional fully supervised methods for learning from noisy labels [18, 33], it actually helps our semi-supervised approach because Stage 1 also reduces the percentage of incorrectly labeled images from 30% to only 0.2%, which effectively gets rid of the misleading supervision of the neural network.

For comparison, we have also included the percentages of correct/incorrect labels of [37]. It explicitly infers the label transition matrix (cf. Figure 3) and uses it to re-weigh the data terms in the loss function. Essentially, it transforms the one-hot labels of an image to a vector of continuous values. We recover the label of the image by either sampling a label from the vector or taking the $\arg\max$ of the vector. The results of both operations are shown in Table 2. Unfortunately, the correction by [37] actually makes the situation worse as more images are incorrectly labeled after the correction. We conjecture that other methods that try to model the label noise could suffer the same issue which then hurt the performance of the neural networks.

Quantitative results of Stage 2. Finally, we apply our semi-supervised method using the processed training set of Stage 1. The corresponding results are shown in Table 3. In addition to the baseline results reported by [37] (the row of cross-entropy and by our own implementation (the row of cross-entropy, i.e., directly training the neural network using the noisy set), we also include the results of the improved baseline, i.e., balancing different classes and refined validation set using the approach proposed in Stage 1 in the training stage. For the competing method, we include both versions of [37], which are the best published methods by the time we submit the paper, as far as we know, on the three datasets studied in this paper. However, we exclude the results of the groundtruth transformation matrix in [37] because such a ground truth is usually unknown in practice.

From Table 3, we can see that our method outperforms the baselines and the existing approach [37] to a large margin when the noisy rate is as high as $p = 0.6$. When the noisy rate is smaller ($p = 0$ and $p = 0.2$), our approach performs better or about the same as [37]. These results verify our modeling hypothesis that, instead of attempting to correct all the noisy labels, the alternative direction can be more effective by ignoring some of the labels in exchange for a small and yet pure labeled set. The more noisy the labels are, the harder to correct the labels or to infer the weights in the loss as done in the existing methods of learning from noisy labels, and the more advantageous our two-stage framework is.

Figure 4. Some labeled and unlabeled images of CIFAR-10 after Stage 1.

Table 2. Some statistics of the correct and incorrect labels by different methods on CIFAR-10 (noisy rate $p = 0.6$)

	Images with correct labels	Images with incorrect labels	Images with no labels
original training set	70.0%	30.0%	0.0%
after [37]'s correction ($\arg\max$)	54.1%	45.9%	0.0%
after [37]'s correction (sampling)	68.1%	31.9%	0.0%
after Stage 1 of our method	16.8%	0.2%	81.0%

4.3. Experiments on Clothing1M

The Clothing1M dataset provides a small clean training set of 50K images. The total number of images with noisy labels is about 1M. We experiment both with and without the direct usage of that clean set for the clothing classification task, and reconcile our experiment setup with that used by [37].

We use the 50-layer ResNet [16] pre-trained on ImageNet [42] as our base neural network classifier. The balancing cost is $\alpha = 100$. Recall that we overly sample the labeled images in the semi-supervised training process so that each mini-batch has the same numbers of labeled and unlabeled examples. We call every pass over all the labeled images a sub-epoch, and in total we run the experiments for 100 sub-epochs.The ADAM optimizer with a learning rate of 3e-4 is used. We divide the learning rate by half when the validation accuracy does not improve for 10 consecutive sub-epochs. We also apply early stopping when the validation accuracy does not improve for 20 consecutive sub-epochs.

Table 4 compares our method with several existing ones. We mainly compare to the loss re-weighting method [37] (rows #5 and #6), which estimates backward and forward label transition matrices and achieves state-of-the-art results on the Clothing1M. In addition, we also include three earlier methods whose results are reported in [37] on the Clothing1M: a one-stage semi-supervised approach using pseudo

Table 3. Comparison results on CIFAR-10 and MINIST

Methods	CIFAR-10 14-layer ResNet				MNIST fully connected			
	$p = 0$	sy.$p = 0.2$	asy.$p = 0.2$	asy.$p = 0.6$	$p = 0$	sy.$p = 0.2$	asy.$p = 0.2$	asy.$p = 0.6$
cross-entropy [37]	87.8	83.7	85.0	57.6	97.9± 0.0	96.9± 0.1	97.5± 0.0	53± 0.6
unhinged (BN) [57]	86.9	84.1	83.8	52.1	97.6± 0.0	96.9± 0.1	97.0± 0.1	71.2 ± 1.0
sigmoid (BN) [12]	76.0	66.6	71.8	57.0	97.2± 0.1	93.1± 0.1	96.7± 0.1	71.4± 1.3
savage [30]	80.1	77.4	76.0	50.5	97.3± 0.0	96.9± 0.0	97.0± 0.1	51.3± 0.4
bootstrap soft [40]	87.7	84.3	84.6	57.8	97.9± 0.0	96.9± 0.0	97.5± 0.0	53.0± 0.4
bootstrap hard [40]	87.3	83.6	84.7	58.3	97.9± 0.0	96.8± 0.0	97.4± 0.0	55.0± 1.3
backward [37]	87.7	80.4	83.8	66.7	97.9± 0.0	96.9± 0.0	96.7± 0.1	67.4± 1.5
forward [37]	87.4	83.4	**87.0**	74.8	97.9± 0.0	96.9± 0.0	97.7± 0.0	64.9± 4.4
cross-entropy	87.9	82.4	85.5	56.2	98.0± 0.1	97.1± 0.1	97.6± 0.2	52.9± 0.6
improved baseline	87.8	83.6	85.2	74.1	98.0± 0.1	97.1± 0.1	97.7± 0.1	**76.7± 1.6**
ours	**88.0**	**84.5**	85.6	**75.8**	**98.2± 0.1**	**97.7± 0.4**	**97.8± 0.1**	**83.4± 1.3**

Table 4. Comparison results on the Clothing1M dataset [59].

#	model	loss / method	initialization	training set	accuracy (reported)	accuracy (our impl.)
1	AlexNet	pseudo-label [25]	#9	1M, 50K	73.04	–
2	AlexNet	bottom-up [47]	#9	1M, 50K	76.22	–
3	AlexNet	label noise model [59]	#9	1M, 50K	78.24	–
4	50-ResNet	cross-entropy	ImageNet	1M	68.94	69.03
5	50-ResNet	backward [37]	ImageNet	1M	69.13	–
6	50-ResNet	forward [37]	ImageNet	1M	69.84	–
7	50-ResNet	**ours**	ImageNet	1M	–	77.34
8	50-ResNet	**ours**	ImageNet	1M, 50K	–	**79.38**
9	AlexNet	cross-entropy	ImageNet	50K	72.63	–
10	50-ResNet	cross-entropy	ImageNet	50K	75.19	74.84
11	50-ResNet	cross-entropy	#6	50K	80.38	–
12	50-ResNet	cross-entropy	#7	50K	–	80.44
13	50-ResNet	cross-entropy	#8	50K	–	**80.53**

labels [25] (row #1), a bottom-up training method for deep neural networks with noisy labels [47] (row #2), and a label noise model [59] (row #3). Finally, we report the results of the baseline neural networks that are trained using the small clean set (50K) and the large noisy set (1M), respectively, with the vanilla cross-entropy loss (rows #4, #9, and #10). From the table, we can see that our two-stage method significantly outperforms all competing approaches. In particular, our 77.34% accuracy is much better than the 69.84% accuracy by the forward label transition [37].

Finally, we fine-tune our neural network classifiers (rows #7 and #8) over the small clean set with a very small learning rate. It is interesting to see that this increases the accuracy with noticeable margins. The same applies to the neural network obtained in [37] (cf. from row #6 to row #11). This verifies the necessity of human annotations, and, on the other hand, it also indicates the need of effective algorithms that can learn from the noisy labels. Given a new visual recognition task, it seems like a reasonable strategy to learn from the noisy training set first and then fine-tune the model with a small manually labeled clean set.

5. Conclusion and discussion

In this paper, we propose a semi-supervised two-stage approach for learning from noisy labels. We devise two techniques to mine the noisy set in the first stage depending on whether or not there is clean data available to the learning agent. After that, we train a deep neural network using a semi-supervised learning method. In our approach, we do not need to know any prior knowledge or estimate any distribution of the noisy labels.

Our approach outperforms the existing methods especially when the noisy rate is high. This confirms our modeling intuition that the network could be misled by the incorrect labels, which are inevitable by the existing label correction methods for learning from the noisy labels. In contrast, we avoid this explicit effort of label correction by completely ignoring the labels of some data points thanks to the use of the semi-supervised learning strategy. Other realization of our two-stage framework will be explored in the future work.

Acknowledgement. This work was supported in part by NSF-1741431.

References

[1] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.

[2] F. A. Breve, L. Zhao, and M. G. Quiles. Semi-supervised learning from imperfect data through particle cooperation and competition. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8. IEEE, 2010.

[3] C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. *Journal of artificial intelligence research*, 11:131–167, 1999.

[4] X. Chen and A. Gupta. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1431–1439, 2015.

[5] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851, 2012.

[6] S. Diersen, E.-J. Lee, D. Spears, P. Chen, and L. Wang. Classification of seismic windows using artificial neural networks. *Procedia computer science*, 2011.

[7] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[8] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1915–1929, 2013.

[9] R. Fergus, Y. Weiss, and A. Torralba. Semi-supervised learning in gigantic image collections. In *Advances in neural information processing systems*, pages 522–530, 2009.

[10] B. Frénay and M. Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2014.

[11] C. Gan, C. Sun, L. Duan, and B. Gong. Webly-supervised video recognition by mutually voting for relevant web images and web video frames. In *European Conference on Computer Vision*, pages 849–866. Springer, 2016.

[12] A. Ghosh, N. Manwani, and P. Sastry. Making risk minimization tolerant to label noise. *Neurocomputing*, 160:93–107, 2015.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[14] P. Guo, H. Huang, Q. Chen, L. Wang, E.-J. Lee, and P. Chen. A model-driven partitioning and auto-tuning integrated framework for sparse matrix-vector multiplication on gpus. In *Proceedings of the 2011 TeraGrid Conference*. ACM, 2011.

[15] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] H. Huang, J. M. Dennis, L. Wang, and P. Chen. A scalable parallel lsqr algorithm for solving large-scale linear system for tomographic problems: a case study in seismic tomography. *Procedia Computer Science*, 18:581–590, 2013.

[18] G. H. John. Robust decision trees: Removing outliers from databases. In *KDD*, pages 174–179, 1995.

[19] L. Joseph, T. W. Gyorkos, and L. Coupal. Bayesian estimation of disease prevalence and the parameters of diagnostic tests in the absence of a gold standard. *American journal of epidemiology*, 141(3):263–272, 1995.

[20] A. Joulin, L. van der Maaten, A. Jabri, and N. Vasilache. Learning visual features from large weakly supervised data. In *European Conference on Computer Vision*, pages 67–84. Springer, 2016.

[21] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[23] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[25] D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013.

[26] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and J. Li. Learning from noisy labels with distillation. *arXiv preprint arXiv:1703.02391*, 2017.

[27] T. Liu and D. Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2016.

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[29] N. Manwani and P. Sastry. Noise tolerance under risk minimization. *IEEE transactions on cybernetics*, 43(3):1146–1151, 2013.

[30] H. Masnadi-Shirazi and N. Vasconcelos. On the design of loss functions for classification: theory, robustness to outliers, and savageboost. In *Advances in neural information processing systems*, pages 1049–1056, 2009.

[31] A. Miranda, L. Garcia, A. Carvalho, and A. Lorena. Use of classification algorithms in noise detection and elimination. *Hybrid Artificial Intelligence Systems*, pages 417–424, 2009.

[32] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *arXiv preprint arXiv:1704.03976*, 2017.

[33] F. Muhlenbach, S. Lallich, and D. A. Zighed. Identifying and handling mislabelled instances. *Journal of Intelligent Information Systems*, 22(1):89–109, 2004.

[34] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. In *Advances in neural information processing systems*, pages 1196–1204, 2013.

[35] A. Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

[36] G. Patrini, F. Nielsen, R. Nock, and M. Carioni. Loss factorization, weakly supervised learning and label noise robustness. In *International Conference on Machine Learning*, pages 708–717, 2016.

[37] G. Patrini, A. Rozza, A. Menon, R. Nock, and L. Qu. Making neural networks robust to label noise: a loss correction approach. *arXiv preprint arXiv:1609.03683*, 2016.

[38] H. Ramaswamy, C. Scott, and A. Tewari. Mixture proportion estimation via kernel embeddings of distributions. In *International Conference on Machine Learning*, pages 2052–2060, 2016.

[39] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.

[40] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[41] D. Rolnick, A. Veit, S. Belongie, and N. Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017.

[42] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[43] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.

[44] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[45] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[46] G. Stempfel and L. Ralaivola. Learning svms from sloppily labeled data. *Artificial Neural Networks–ICANN 2009*, pages 884–893, 2009.

[47] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*, 2014.

[48] J.-w. Sun, F.-y. Zhao, C.-j. Wang, and S.-f. Chen. Identifying and correcting mislabeled training instances. In *Future generation communication and networking (FGCN 2007)*, volume 1, pages 244–250. IEEE, 2007.

[49] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284, 2017.

[50] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[51] A. Tarvainen and H. Valpola. Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017.

[52] C. M. Teng. Evaluating noise correction. In *Pacific Rim International Conference on Artificial Intelligence*, pages 188–198. Springer, 2000.

[53] C.-M. Teng. A comparison of noise handling techniques. In *FLAIRS Conference*, pages 269–273, 2001.

[54] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[55] A. Vahdat. Toward robustness against label noise in training deep discriminative neural networks. *arXiv preprint arXiv:1706.00038*, 2017.

[56] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604, 2015.

[57] B. Van Rooyen, A. Menon, and R. C. Williamson. Learning with symmetric label noise: The importance of being unhinged. In *Advances in Neural Information Processing Systems*, pages 10–18, 2015.

[58] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. Belongie. Learning from noisy large-scale datasets with minimal supervision. *arXiv preprint arXiv:1701.01619*, 2017.

[59] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2691–2699, 2015.

[60] X. Zhu. Semi-supervised learning literature survey. 2005.

[61] X. Zhu and A. B. Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

[62] B. Zhuang, L. Liu, Y. Li, C. Shen, and I. Reid. Attend in groups: a weakly-supervised deep learning framework for learning from web data. *arXiv preprint arXiv:1611.09960*, 2016.