Iterative reweighting learning for open-set label noise via clean samples

Master Thesis

Jiarong Fan

Ecole Centrale de Pékin, 2020

Abstract. In recent years, great progress has been made to deal with abnormal examples in Deep Neural Networks (DNNs). However, along with the demand of more massive datasets, experiments show always a performance degradation due to label noise, which can be easily overfitted by DNNs. In this work, we propose a new iterative reweighting learning method of using small number of clean samples to deal with label noise. We use clean samples to fine-tune a trained model, and by leveraging the difference of representative features between models before and after fine-tune, we build class-based discriminators to reweight samples. Experiments on different benchmark datasets demonstrate our model's improvements over state-of-the-art methods and, the analysis of fine-tune with clean data may also be used in other tasks.

Keywords: Label noise, representation learning, fine-tune, iterative framework

1 Introduction

Supervised learning with labelled instance has been very successful in machine learning tasks, esp. in deep convolutional neural networks (CNN). Under many situations, the labels of images are more easily corrupted due to some inevitable factors, like limit knowledge, expert-required work, and many other factors. This stochastic process that pollutes labels is called label noise [7]. Besides, many works [7, 24] have given clear evidence that containing mislabelled instances could be harmful to both the accuracy and training time of a model. As a result, a collection of clean datasets is usually necessary to obtain a robust and more accurate model. However, with the deep learning networks becoming more complex, the size of dataset required is also in fast-growing, which is time-consuming and money-consuming, not to mention to guaranty the accuracy of label.

To cope with this imperfection, many approaches have been proposed. We categorize these methods into two groups: (1) *Only noisy data* approach and (2) *Noisy data with few clean data* approach. Method in the first category uses only the corrupted dataset to train a model and aims at relabelling or removing suspect samples in training stage, by utilization of correction methods such as loss correction [10, 28] or adding extra correction layers [8, 32]. Another relabelling method is to build annotator models by considering the generation of label noise as crowd-sourcing [19, 22, 33, 37, 38] where each annotator could be modelled by

giving a certain distribution of labels. The *Noisy data with few clean data* approach addresses this issue with the help of a small number of clean samples to supervise the training process by learning mappings between noisy and clean annotations [17, 35, 39]. This approach has shown accuracy improvement in many noisy datasets with the help of clean data, and theoretically we also believe that a small number of verified labels brings more information that can be learned by a model.

Recently, a more complex yet common scenario is introduced [36] where mislabelled instance could not belong to any class in existed dataset, and this scenario is referred to as the open-set noisy label problem[1]. Obviously, this extends the existed problem and disables many methods. Within the first *Only noisy data* approach, recent methods take assumptions on noise or annotator to relabel noisy samples. Theses approach, however, can't work under open-set noise scenario since the correction of label is always false. Other methods [18, 36] propose to reweight noisy data in loss function. The reweighting methods work for both open-set and closed-set problem, but the existed reweighting methods rely on the hidden representations of noisy samples, which may not be accurate to be passed to a reweighting module. Methods in *Noisy data with few clean data* approach focus on building relationship between verified label and noisy label to correct noise, thus the current methods could not work well under open-set noise scenario as well. In this work, to get into a more complex yet real situation, we adopt the open-set problem. In face of this problem, the existed *Only noisy data* approach is influenced by original noisy samples and there is still no mature framework to bring a small clean dataset for open-set noise problem.

In this paper, we present a new label noise cleansing method that is simple and efficient and show that our way of using small number of verified instances could be more helpful to the accuracy of a model than mixing the verified instances directly in the training set. Actually, there exists two difficulties to build a reweighting method with few clean data: (1) As mentioned above, the current reweighting method works on the feature space of data, which contains usually noise. If the feature representations are not clean enough, the reweighting module would misidentify the corrupted samples. (2) Except for building mappings between clean labels with noisy labels to relabel data, the commonly used approach of combining clean and noisy data is to first pre-train a network using the large noisy dataset and then fine-tune it with the small verified dataset. However, this approach does not fully leverage the information contained in the clean set [35]. Without any label noise assumption, how to fully leverage a small set of clean data is also a big problem.

To deal with the two difficulties, our method is in two parts: (1) To augment the quality of extracted feature, inspired by the idea of Siamese Network [36], we add clean data only to rectify the centre of class feature in a dimension-reduced space. (2) Inspired by the commonly used fine-tuning and the fact that

[1] If the noisy samples possess a true class that is not present in the training data, the label noise is of open-set type [36], and otherwise closed-set type such as symmetric and asymmetric noise type [7].

DNNs tend to learn simple patterns first under label noise [1], we investigate the variation of feature space with respect to clean samples and noisy samples, and show that the two datasets present different variations. A weight generative framework is set up to make use of the difference to reweight data. Experiments show that the proposed approach outperforms the state-of-the-art methods for open-set noise type and provides a novel perspective on model fine-tuning.

Our main contributions are as follows:

- Following the open-set noisy label problem, we are able to make use of a small number of verified samples to improve the quality of representative feature and the accuracy of model.
- We innovatively investigate the effects of fine-tune process in face of label noise in representation learning. We then make use of these effects to distinguish clean and noisy data.
- We propose a weight generative method to generate weights for each sample. Compared with previous methods, our method shows improvement for open-set label noise.

2 Related work

To reduce the impact of mislabelled samples in a dataset, there are several popular ideas, and we divide them into two categories: *Only noisy data* approach and *Noisy data with few clean data* approach. The first approach uses only the current noisy dataset whereas the second exploits the information in a small number of clean samples. When it comes to the open-set problem, most of the methods could not be used since they make different assumptions and relabel corrupted data. To deal with this newly raised problem, our work leverages many recent researches in several different areas, summarized below.

Only noisy data approach. Some previous methods like [3] seek for models which are robust to the presence of label noise, some other methods aim at cleansing data: either re-correcting existed label under closed-set label noise assumption or removing mislabelled samples. To prove the efficiency of re-correcting existed label, many works adopt manually produced symmetric and asymmetric label noise among the original dataset. Junnan et al. [21] divides dataset into two parts and trains two models that perform simultaneously label co-refinement and label co-guessing to re-correct labels. Some new methods [32, 41] exploit the ability of network and relabel samples iteratively by model predictions. Adding loss correction [10, 28] and adaptation layer to estimates the noise transition matrix [8] are also well studied. Other methods aim at removing noisy samples from dataset, such as detecting mislabelled samples by classification confidence [31]. A very efficient outlier detection factor called local outlier factor (LOF) [5] is also proposed to detect open-set label noise [36]. Like outlier detection, Sluban et al. [29] use a high agreement of several classification filters to detect and to remove mislabelled data. Under open-set noise assumption, where we can't relabel sample to an existed class, we propose a joint discriminator to discriminate mislabelled ones and update their weights in loss function.

Noisy data with few clean data approach. To make labels for all data is sometimes impossible, esp. for the dataset which contains millions of samples. But it is often easy to obtain a small number of clean labels. Hu et al. [17] use supervision of clean data to avoid overfitting, similar work has also been done by Veit et al [35]. Modelling label noise [40] and pre-training [11, 26] with a small clean dataset are also fully researched. Our proposed method belongs to this category. Inspired by the similar goal of making full use of little clean data, we design a framework that amplifies the information between clean data and noisy data. However, existed methods in this category only try to relabel data with the help of clean samples. Different from previous methods, we simply fine-tune a trained model with clean data, and we can then use the modification of model to give a lower weight to mislabelled samples.

Representation learning. Actually, the different noisy label learning methods mentioned above act on either the images themselves or their reduced dimension features which belong to their representations. Bengio et al. [4] mentions the challenge and possible criteria for representation learning, as well as the importance of representation learning in deep learning area: for a machine learning problem, data conversion is necessary in most cases. However, the traditional data processing method can be a long haul compared with some current representation learning models by DNN. To deal with label noise, embedding CNN layers to extract features are used for following label cleaning procedures [20, 35], similar works of exploiting samples' feature space are also researched by others [2, 8, 25, 43]. Guo et al. [9] measures the complexity of data using its distribution density in feature space to reduce the impact of noisy labels. To learn more useful representations of data, Siamese Network [6] and Triplet Network [12] are proposed. Siamese Network is used to extract features followed by LOF layer to reweight open-set noisy samples [36], but the extracted features are also potentially polluted. Inspired by it and Triplet Network, we propose a new contrastive loss for Triplet Network in using some clean samples to maintain the centre of a class of data.

3 Methodology

In this section, we present details of the proposed data cleaning network. Traditionally, fine-tuning is to continue training a network that has already been trained to make it perform another similar task. However, we propose a new utilization of the fine-tune process. We compare the variation of clean data and noisy data during fine-tuning. As a result, we design a novel framework that leverages the difference to identity noisy data.

3.1 Overview

As shown in Fig. 1, the whole framework of our proposed model is rather simple. We propose a new loss function for triplet neural network to improve the quality of feature after the Triplet Network. The orange part is the stem CNN classifier.

The blue part is a feature decoder which is used to give classification results. The purple part is a feature discriminator with scalar output that will serve as sample weight. As black dotted line demonstrates, we first train the stem model with designed triplet loss to get a model f. We then fine-tune the stem model to $\widetilde{f}$ with weight-added small amount of clean data, i.e. we continue training, but at the same time, the clean data's weight is increased in the classification loss function. We build negative samples from the output of f and positive samples from $\widetilde{f}$ to train a purple discriminator for each class of sample. The output of the discriminator should be a weight between 0 and 1 which is regularized by sigmoid function. This weight is then used as degree of confidence to retrain the classifier. With iterations of the training classifier and the training discriminator, we will get a more accurate final model.

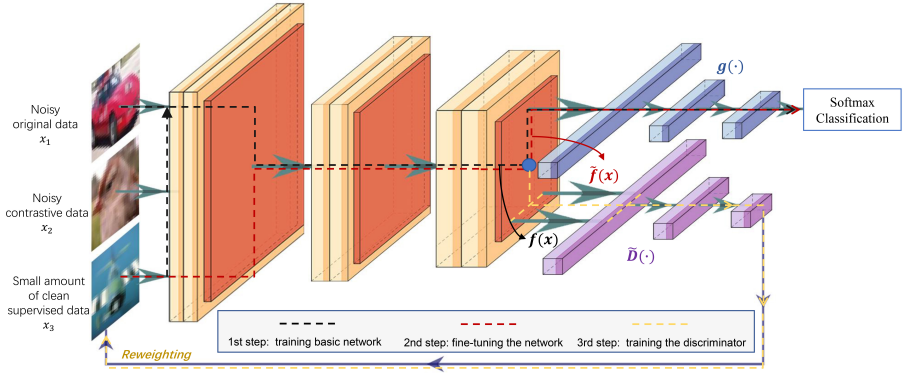


Fig. 1. The Overview of proposed model. The orange part is a stem feature extractor, with feature decoder the blue part. The purple part is a discriminator that gives degree of confidence to each sample. The three steps make up an iteration of training. The stem feature extractor and the feature decoder will serve as the final classifier. When training this model, we first train $g \circ f$ on a triplet loss from data x_1, x_2, x_3 to get stem model f, then we fine-tune $g \circ f$ with clean samples to get $\widetilde{f}$. At last, we regard outputs of f as negative samples and that of $\widetilde{f}$ as positive samples to train a discriminator $\mathcal{D}$, which will update the weights for noisy samples.

3.2 Triplet loss design

A Triplet Network is allowed to learn by comparisons of instances rather than label-instance pair [13]. But this original Triplet Network is only set to train the similarity and dissimilarity between samples and the model is based on clean labels, while under our assumption the given labels are potentially polluted. To take full advantage of some number of clean samples, we propose the following base network and loss function.

The network is composed of a triplet encoder f and a decoder g. The nature of a triplet encoder is a neural network whose weights are shared with another two networks. The nature of the decoder is a classifier that classifies the dimension-reduced feature from the triplet encoder. For a set of noisy samples $\mathcal{H} = \{x_i^j\}$, $i \in [|1, n|]$, $j \in [|1, m|]$, where n is the number of samples and m is the number of class, x_i^j represents the i^{th} sample which is of j^{th} class. For a set of clean samples $\mathcal{C} = \{\bar{x}_i^j\}, i \in [|1, b|], j \in [|1, m|]$, b is the number of clean samples. With two random inputs x_i^j, x_k^q, we match a third instance $\bar{x}_p^j$ so that it's the same class with the first input. Denote $d(x, y)$ a distance function that evaluates how far it is between x and y in their space. The contrastive loss function is defined as below:

$$CtLoss(x_i^j, x_k^q, \bar{x}_p^j) = (2\sigma_{jq} - 1)d(f(x_i^j), f(x_k^q)) + K(x_i^j, \bar{x}_p^j) \tag{1}$$

$$K(x_i^j, \bar{x}_p^j) = \lambda \cdot \max\{(d(f(x_i^j), f(\bar{x}_i^j)) - \varepsilon), 0\} \tag{2}$$

Where σ_{jq} is 1 iff $j = q$, else 0. λ measures how much the clean samples involve in the loss function. To avoid overfitting between clean sample and original sample, we add a $\varepsilon > 0$. In this way, when these two samples are close enough, we don't need to push them closer anymore. It's also worth mentioning that this contrastive loss can be applied after any hidden layer.

The output of decoder g is a SoftMax classification, we add a cross entropy loss recorded as $CeLoss$ on the contrastive loss so that the decoder could work without extra training:

$$Loss(x_i^j, x_k^q, \bar{x}_p^j) = \gamma \cdot CtLoss(x_i^j, x_k^q, \bar{x}_p^j) + CeLoss(g \circ f(x_i^j), j) \tag{3}$$

γ is a regularization term of contrastive loss. In this way, the composed triplet loss enables the stem network to generate the features for each sample, while the decoder can be served as the final classifier once the training is completed.

3.3 Two-phases weight generative network

As mentioned above, we add clean data to rectify the triplet loss. The use of verified data in contrastive loss is to adjust the centre of representation in dimension-reduced space. To make full use of these data, we propose this two-phases weight generation network. To begin with, we train an initial model with triplet loss, and we denote the stem network as f. Then we fine-tune f to $\widetilde{f}$ so that $\widetilde{f}$ contains more clean information than f, by using cross entropy loss function with no gradients in the feature classifier g. To utilize the difference between f and $\widetilde{f}$, a discriminator is set to memorize the gap. For some input $\{x_i^q\}, i \in [|1, k|]$, we train a discriminator $\mathcal{D}_q$ that minimizes:

$$Loss(\mathcal{D}_q) = -E_{x \in \{x_i^q\}}[log(1 - \mathcal{D}_q(f(x))) + log(\mathcal{D}_q(\widetilde{f}(x)))] \tag{4}$$

where $E_{x \in \{x_i^q\}}$ represents the mean for the current batch. The loss function comes from the binary cross entropy (BCE) loss, where the label is 0 for $f(x)$ and 1 for $\widetilde{f}(x)$. When training the discriminator, the f and $\widetilde{f}$ remains unchanged. Once the training epoch is finished, the same discriminator will be used to generate an initial weight for each sample in training set: for a known batch of sample $\mathcal{B}_q = \{x_i^q\}, i \in [|1, b|]$, we calculate their weights $\mathcal{W}_q$ by:

$$\mathcal{W}_q(x_i^q) = \mathcal{D}_q(\widetilde{f}(x_i^q)), i \in [|1, b|] \tag{5}$$

The output of discriminator follows a sigmoid function, so the weight is always between 0 and 1. By now, the first iteration is done, each weight is assigned for its corresponding sample, from the second iteration, the previous weight is set as initial value and we repeat the same step:

1. Fine-tune f to $\widetilde{f}$ with clean data and the new weights for the whole training data. It is worth noting that we don't just use the clean data, but add extra weights for it in SoftMax loss function compared with basic training data.

2. Repeat the discriminator/generator step to refresh the weight for each data.

The whole algorithm is shown in Algorithm. 1.

Algorithm 1 Framework of weight generative network.

Input: The set of original samples $\{x_i^j\}$; The small set of clean samples $\{\bar{x}_i^j\}$; The Triplet Network f; The feature decoder g; The discriminator $\mathcal{D}_j$; The original weight for each sample $\{w_i\}$;

Output: The classifier and the weight for each sample.

```
Train the classifier g ∘ f with both {x_i^j} and {x̄_i^j} by triplet loss;
repeat
    Fine-tune f to f̃ with {x_i^j} and weights-added {x̄_i^j}
    Train discriminator with f and f̃ using {x_i^j} with weight {w_i}
    for j ∈ [|1, m|] do
        for each batch of {x_i^j} do
            Generate new weights for the current batch and update {w_i}
        end for
    end for
until The change of {w_i} < ε
```

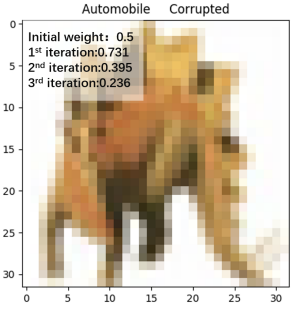
3.4 Data and experiments setup

Since we follow the subject of open-set label noise problem, we use the same method of corruption with Wang et al. [36], corrupting the original data with unrelated samples. Results of their work show that the source of noisy samples has little influence on final model. Similarly, we use only CIFAR-100 to add open set label noise for CIFAR-10 with proportion 0%, 20%, 40%, 60%, 80% and to corrupt MNIST with proportion 40%. A proportion of original data is

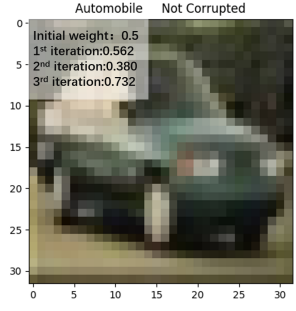
replaced by other irrelevant data from CIFAR-100. For the triplet encoder, we use VGG16 architecture for benchmark dataset CIFAR-10 and VGG11 architecture for dataset MNIST. Batch normalization layer is also used after each convolutional layer in the triplet encoder. The discriminator for both is a 3-layer FC architecture, with Leaky ReLU as activation function and sigmoid at last. The encoder/decoder is trained using SGD with momentum 0.9, and weight decay $5e^{-}4$. The learning rate is 0.1 before 150 epochs, 0.01 between 150 and 300 epochs, and 0.001 afterwards. The models will not stop until the accuracy on validation set no longer changes for 150 epochs.

4 Experiment results and ablation study

Our basic idea is quite simple. Take CIFAR-10 dataset for example, our idea is shown in Fig. 2 where two samples labelled as automobile are selected. On the left, one sample obviously not belong to automobile class is shown, we can see that after the three first iterations, the weight is reduced. For the clean sample of automobile, its weight becomes larger during the weights generating process.



(a) Corrupted automobile sample



(b) Uncorrupted automobile sample

Fig. 2. Samples of automobile and weights generation for first 3 iterations

4.1 Results analysis

We select 3 state-of-art models fit for open-set problem according to Wang et al. [36] and two other suitable and efficient SOTA models [21, 42]. We experimentally test these 5 competitors and a baseline of cross entropy. We also add 5% of train set as clean data to fine-tune all the stem classifier of these 5 models. The DivideMix and MetaCleaner models [21, 42] use separately wide-ResNet28 and 18-layer ResNet for CIFAR-10, here to make it fair, we adopt VGG16 as the stem network for each SOTA model. Results show that our method outperforms the open-set problem methods. The Forward and Backward methods [27], the DivideMix[21] are originally used for closed-set problem, where noisy labels are relabelled within the existed class to get better results. Our model is designed

to better fit the open-set problem, which is an extension of closed-set problem. To get a full-scale comparison, we also introduce symmetric closed-set noisy labels to CIFAR-10, because the symmetric noisy labels are shown to be more harmful to the results [34]. However, since our model cannot relabel samples automatically, it is thus not as good as others in closed-set problem, and we test only two different closed-set settings to show that our method works better than pure cross entropy loss. The comparisons are shown below in Table 1.

Table 1. Accuracy comparison between single DNN cross entropy loss and triplet DNN loss function

Open-set noise	CIFAR-10	CIFAR-10	CIFAR-10	CIFAR-10	CIFAR-10	MNIST
Noise ratio	0%	20%	40%	60%	80%	40%
Cross entropy	95.87%	85.01%	81.92%	70.46%	54.55%	83.72%
Forward [27]	-	86.98%	83.03%	74.79%	69.82%	85.96%
Backward [27]	-	82.91%	71.80%	67.53%	65.44%	79.77%
Wang et al. [36]	-	87.52%	87.29%	83.18%	71.49%	90.46%
DivideMix [21]	-	88.63%	85.99%	72.09%	60.01%	85.12%
MetaCleaner [42]	-	89.27%	87.38%	83.10%	68.10%	93.82%
Single DNN+2-step	-	90.06%	88.00%	84.71%	74.62%	94.91%
Triplet DNN	91.65%	83.57%	83.39%	71.83%	66.28%	82.59%
Triplet DNN+2-step	-	**91.60**%	**90.04**%	**86.90**%	**74.69**%	**95.30**%
Closed-set noise	CIFAR-10	CIFAR-10	CIFAR-10	CIFAR-10	CIFAR-10	MNIST
Noise ratio	0%	20%	40%	60%	80%	40%
Cross entropy	95.87%	83.82%	79.35%	-	-	-
DivideMix [21]	-	**95.73**%	**95.11**%	-	-	-
MetaCleaner [42]	-	93.00%	91.88%	-	-	-
Triplet DNN+2-step	-	89.97%	89.65%	-	-	-

4.2 Ablation study

In this section, we would like to review some experimental arguments for the choice of different setup mentioned above. To help readers understand why these will work, we propose first some assumptions that we are going to prove by experimental approaches.

Assumption 1 *The triplet loss proposed is beneficial for generating better representative features for following label noise cleansing procedure.*

Assumption 2 *The fine-tune process influences the feature space of clean data more than that of noisy data.*

Triplet neural network As mentioned above, a triplet neural network is used in experiments. Several experiments are conducted to prove Assumption .1, i.e. the effectiveness of triplet neural network.

(1) Adding extra constraints to a DNN sometimes reduces its performance, but in case of open-set label noise, the accuracy on test set is slightly higher than that in a single DNN when noise proportion is bigger than 40%, as shown in the Table 2.

Table 2. Accuracy comparison between single DNN cross entropy loss and triplet DNN loss function

Dataset noise	CIFAR-10 0%	CIFAR-10 20%	CIFAR-10 40%	CIFAR-10 60%
Single DNN loss	**95.87**%	**85.01**%	81.92%	70.46%
Triplet DNN loss	91.65%	83.57%	**83.39**%	**71.83**%

We can see that the triplet loss function doesn't reduce the model's performance, although it can be explained by the fact that we increase the weight of clean data in training process by adding the term $K\left(x_i^j, \bar{x}_p^j\right)$ in loss function. In this way, we are able to avoid excessive deviations of noisy data in their low-dimension representative feature and keep the most important information.

(2) In the case of training f under 40% label noise, we choose 50 samples of each class in CIFAR-10 test set to compare the model trained by single DNN and that by triplet DNN. By using the t-SNE method to visualize the 500 samples in their low-dimension representative space in Fig .3 and Fig .4, we can see that under triplet loss the stem network gather better the low-dimension feature than single loss function. Because in the two-phases weight learning step, the discriminator tries to learn the pattern of a class of samples. Intuitively and experimentally, the aggregation of these features helps the weight generation step. Similar results are also shown in work of Wang et al. [36], where the condensed features fit better the outlier detection methods.

Fig. 3. Representations(t-SNE 2D embeddings) of 10 classes of 500 CIFAR-10 samples trained by Single DNN

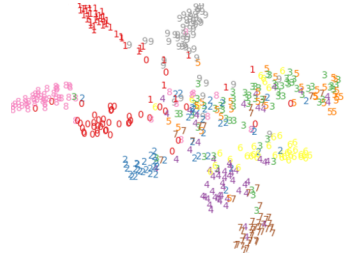


Fig. 4. Representations(t-SNE 2D embeddings) of 10 classes of 500 CIFAR-10 samples trained by Triplet DNN

Fine-tuning with clean data In this part, some experimental results are to be reviewed to prove the Assumption .2. In general case, if we have more useful

data, the model can get more information and have a better performance [30]. Besides, in work of Ma et al. [23], a two-stage learning scenario is proposed in terms of Local Intrinsic Dimensionality (LID) [14–16] which is the dimension of a submanifold that contains an element x and fits the sample distribution around x. In their work, the subspace dimensionality of training and test examples is affected by label noise, and the training process acts like two totally different learning styles.

In case of label noise, to figure out how this model has changed after fine tuning f to $\widetilde{f}$, we set up two experiments:

(1) The most intuitive way is to use a distance function $d(x, y)$ and observe the distance between f and $\widetilde{f}$. Denote $\{x_T\}$ the clean data in train set, $\{x_F\}$ the noisy data in train set. Experimentally, we select all data from the first class, and we calculate the L2-distance $d(f, \widetilde{f})$. Separately, the distance histograms of the two sets are drawn in Fig .5 and Fig .6 to show the difference. After fine-tuning, the features of clean data tend to have a greater change than noisy data. This difference between correct data and mislabelled data is then used for training discriminator.

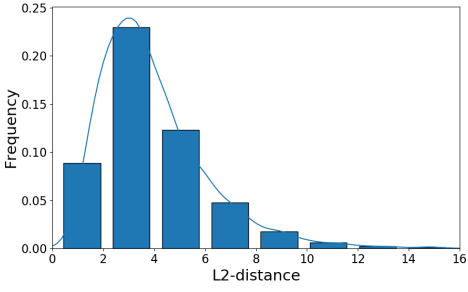


Fig. 5. Histogram of $d(f, \widetilde{f})$ for $\{x_T\}$

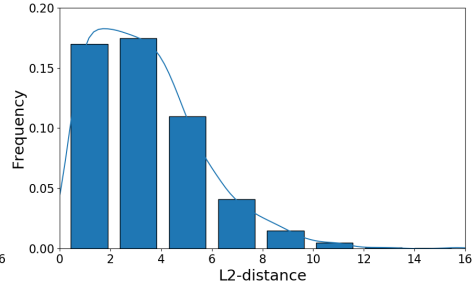


Fig. 6. Histogram of $d(f, \widetilde{f})$ for $\{x_F\}$

(2) Local Intrinsic Dimensionality (LID) is an expansion-based measure of intrinsic dimensionality of the underlying data subspace [14–16]. It reveals the dimension of the submanifold containing x that would best fit the data distribution in the vicinity of x. Ma et al. [23] provides an efficient way to estimate LID through batch sampling and, by comparing the training process under clean data with that under noisy data, proposes a two-stage of learning that DNN follows in the presence of label noise: a) The early stage is *dimensionality compression* where LID score decreases continually. b) The later stage is *dimensionality expansion* where the learning process overfits the data and thus the LID score increases. For more explanation and proof, we refer to their paper [23].

Using LID for reference, we compute the estimation of LID the same way in work of Ma et al. [23] through batch sampling during the fine-tune stage to see how the new added clean data act in training process. The LID score is used to figure out which kind of impact that fine-tune process makes on the whole

model. As shown in Fig. 7, a comparison between using fine-tune and without fine-tune is made. The experiment is done on CIFAR-10 with 40% label noise. The blue line and the red one begin from the same checkpoint where the model is already in *dimensionality expansion stage*, so the LID score rises for both from the beginning. The vertical black line indicates the intervention of fine-tune process. The learning parameters are the same. The LID score is estimated through batch sampling every 1/5 epoch. We infer from the figure that the fine-tune process delays the dimensionality expansion stage, and most importantly, weakens the effect of label noise according to Ma et al. [23]. Thus, the fine-tuned model contains more clean information that we are going to exploit.

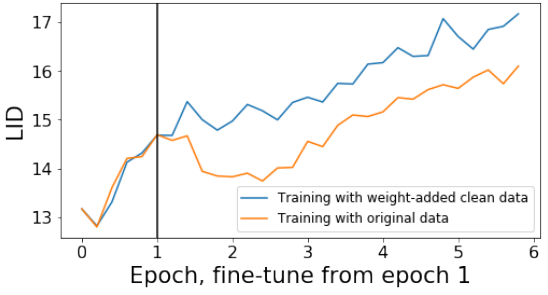


Fig. 7. Influence of fine-tune process in terms of Local Intrinsic Dimensionality

To summarize these two experiments, there exists a clear difference between fine-tuned model $\widetilde{f}$ and original model f. A discriminator is thus proposed to utilize the difference.

Convergence of two-phases weight generative network Based on the two assumptions mentioned above, we design a discriminator to generate weights for each sample, and here we simply discuss whether our two-phases framework leads to a good convergence.

With input positive features and negative features, the discriminator learns the pattern of "good" feature. As discussed in previous section, if a sample belongs to the true label category, it's variation after fine-tuning is larger than noisy sample. Therefore, the clean sample contributes a bigger loss. The discriminator tries to separate $f(x_T)$ with $\widetilde{f}(x_T)$, and become less sensitive to the same difference of x_F. Let's consider a 2-D distribution of samples in Fig. 8. The orange 'X' represents feature of noise sample by original model, and the blue ones represent clean samples of the same model. After fine-tuning, the features of noise samples have little variations while that of clean samples moves a lot. We know nothing about if a feature is from noisy sample or not, but we know if it's from the fine-tuned model. Therefore, we can build models to separate them, and here, a linear regression discriminator (the red line, in fact a hyperplane

since the feature is in 2-D) is supposed to fit them. If we set the predictions of the linear model to be 0 on left and 1 on right, the output of the discriminator can also separate the noise samples with clean samples, and the discriminator should be functioned on the output of $\widetilde{f}$ instead of f to get correct weights. This is how it works in our case, instead we just apply a NN discriminator in consideration of the complexity of sample features. Since the goal is to train a good classifier of dataset, the weights are then applied on the loss function of each sample batch.

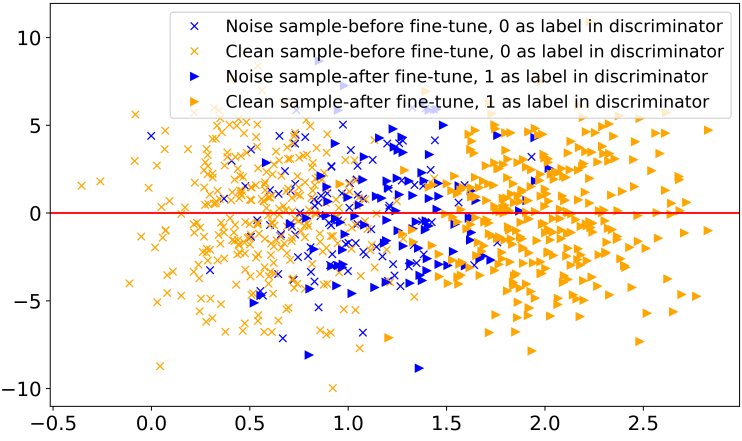


Fig. 8. A possible example of two types of features

5 Conclusions

In this paper, we first investigate the effects of Triplet Network in face of label noise, esp. when there are few number of clean data to take use of. Because most of label cleansing methods rely on the dimension reduced representative features of original data rather than the raw images, the quality of features is crucial for the following steps. We then study the impacts of fine-tune process on clean samples' feature as well as noisy samples'. The results show that fine-tune process reduces the Local Intrinsic Dimensionality to avoid overfitting and that the clean samples are easier to be affected than noisy samples. In view of this characteristic, we then construct positive samples from model with fine-tune and negative samples from model without fine-tune. A discriminator is trained to enlarge the difference between these two types of feature. According to the results of discriminator, a weight generative approach is proposed to generate weights for each sample belonging to every class. The weights are added to loss function to train the feature extractor and feature decoder. In this way, an

iterative training process is executed to produce the final weights for data and a more precise classifier. For future research, we hope that the usage of fine-tune could also be applied in other fields, such as downsizing datasets or separating hard/soft instances within a dataset.

References

1. Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., et al.: A closer look at memorization in deep networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 233–242. JMLR. org (2017)
2. Azadi, S., Feng, J., Jegelka, S., Darrell, T.: Auxiliary image regularization for deep cnns with noisy labels. arXiv preprint arXiv:1511.07069 (2015)
3. Beigman, E., Klebanov, B.B.: Learning with annotation noise. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1. pp. 280–287. Association for Computational Linguistics (2009)
4. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1798–1828 (2013)
5. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data. pp. 93–104 (2000)
6. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a" siamese" time delay neural network. In: Advances in neural information processing systems. pp. 737–744 (1994)
7. Frenay, B., Verleysen, M.: Classification in the presence of label noise: A survey. IEEE Transactions on Neural Networks and Learning Systems **25**(5), 845–869 (May 2014). https://doi.org/10.1109/TNNLS.2013.2292894
8. Goldberger, J., Ben-Reuven, E.: Training deep neural-networks using a noise adaptation layer (2016)
9. Guo, S., Huang, W., Zhang, H., Zhuang, C., Dong, D., Scott, M.R., Huang, D.: Curriculumnet: Weakly supervised learning from large-scale web images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 135–150 (2018)
10. Han, B., Yao, J., Niu, G., Zhou, M., Tsang, I.W.H., Zhang, Y., Sugiyama, M.: Masking: A new perspective of noisy supervision. In: NeurIPS (2018)
11. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
12. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: International Workshop on Similarity-Based Pattern Recognition. pp. 84–92. Springer (2015)
13. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: International Workshop on Similarity-Based Pattern Recognition. pp. 84–92. Springer (2015)
14. Houle, M.E.: Dimensionality, discriminability, density and distance distributions. In: 2013 IEEE 13th International Conference on Data Mining Workshops. pp. 468–473. IEEE (2013)
15. Houle, M.E.: Local intrinsic dimensionality i: an extreme-value-theoretic foundation for similarity applications. In: International Conference on Similarity Search and Applications. pp. 64–79. Springer (2017)
16. Houle, M.E.: Local intrinsic dimensionality ii: multivariate analysis and distributional support. In: International Conference on Similarity Search and Applications. pp. 80–95. Springer (2017)
17. Hu, M., Han, H., Shan, S., Chen, X.: Weakly supervised image classification through noise regularization. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)

18. Jiang, L., Zhou, Z., Leung, T., Li, L., Fei-Fei, L.M.: Learning data-driven curriculum for very deep neural networks on corrupted labels. In: International Conference on Machine Learning. pp. 2309–2318 (2018)
19. Khetan, A., Lipton, Z.C., Anandkumar, A.: Learning from noisy singly-labeled data. arXiv preprint arXiv:1712.04577 (2017)
20. Lee, K.H., He, X., Zhang, L., Yang, L.: Cleannet: Transfer learning for scalable image classifier training with label noise. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5447–5456 (2018)
21. Li, J., Socher, R., Hoi, S.: Dividemix: Learning with noisy labels as semi-supervised learning (2020)
22. Liu, Q., Peng, J., Ihler, A.T.: Variational inference for crowdsourcing. In: Advances in neural information processing systems. pp. 692–700 (2012)
23. Ma, X., Wang, Y., Houle, M.E., Zhou, S., Erfani, S.M., Xia, S.T., Wijewickrema, S., Bailey, J.: Dimensionality-driven learning with noisy labels. arXiv preprint arXiv:1806.02612 (2018)
24. Nettleton, D.F., Orriols-Puig, A., Fornells, A.: A study of the effect of different types of noise on the precision of supervised learning techniques. Artificial intelligence review **33**(4), 275–306 (2010)
25. Niu, L., Tang, Q., Veeraraghavan, A., Sabharwal, A.: Learning from noisy web data with category-level supervision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7689–7698 (2018)
26. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1717–1724 (2014)
27. Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., Qu, L.: Making deep neural networks robust to label noise: A loss correction approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1944–1952 (2017)
28. Patrini, G., Rozza, A., Menon, A., Nock, R., Qu, L.: Making neural networks robust to label noise: a loss correction approach. stat **1050**, 13 (2016)
29. Sluban, B., Gamberger, D., Lavra, N.: Advances in class noise detection. In: Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence. pp. 1105–1106. IOS Press (2010)
30. Sun, C., Shrivastava, A., Singh, S., Gupta, A.: Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE international conference on computer vision. pp. 843–852 (2017)
31. Sun, J.w., Zhao, F.y., Wang, C.j., Chen, S.f.: Identifying and correcting mislabeled training instances. In: Future generation communication and networking (FGCN 2007). vol. 1, pp. 244–250. IEEE (2007)
32. Tanaka, D., Ikami, D., Yamasaki, T., Aizawa, K.: Joint optimization framework for learning with noisy labels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5552–5560 (2018)
33. Tanno, R., Saeedi, A., Sankaranarayanan, S., Alexander, D.C., Silberman, N.: Learning from noisy labels by regularized estimation of annotator confusion. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
34. Vahdat, A.: Toward robustness against label noise in training deep discriminative neural networks. In: Advances in Neural Information Processing Systems. pp. 5596–5605 (2017)

35. Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., Belongie, S.: Learning from noisy large-scale datasets with minimal supervision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 839–847 (2017)
36. Wang, Y., Liu, W., Ma, X., Bailey, J., Zha, H., Song, L., Xia, S.T.: Iterative learning with open-set noisy labels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8688–8696 (2018)
37. Welinder, P., Branson, S., Perona, P., Belongie, S.J.: The multidimensional wisdom of crowds. In: Advances in neural information processing systems. pp. 2424–2432 (2010)
38. Whitehill, J., Wu, T.f., Bergsma, J., Movellan, J.R., Ruvolo, P.L.: Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In: Advances in neural information processing systems. pp. 2035–2043 (2009)
39. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2691–2699 (2015)
40. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2691–2699 (2015)
41. Yi, K., Wu, J.: Probabilistic end-to-end noise correction for learning with noisy labels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7017–7025 (2019)
42. Zhang, W., Wang, Y., Qiao, Y.: Metacleaner: Learning to hallucinate clean representations for noisy-labeled visual recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7373–7382 (2019)
43. Zhou, H., Sun, J., Yacoob, Y., Jacobs, D.W.: Label denoising adversarial network (ldan) for inverse lighting of faces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6238–6247 (2018)