CLC : Noisy Label Correction via Curriculum Learning

Jaeyoon Lee
Department of Electronic Engineering
Hanyang University
Seoul, Korea
zxc1421@hanyang.ac.kr

Hyuntak Lim
Department of Electronic Engineering
Hanyang University
Seoul, Korea
lim3944@hanyang.ac.kr

Ki-Seok Chung
Department of Electronic Engineering
Hanyang University
Seoul, Korea
kchung@hanyang.ac.kr

***Abstract*—Deep neural networks reveal their usefulness through learning from large amounts of data. However, unless the data is correctly labeled, it may be very difficult to properly train a neural network. Labeling the large set of data is a time-consuming and labor-intensive task. To overcome the risk of mislabeling, several methods that are robust against the label noise have been proposed. In this paper, we propose an effective label correction method called *Curriculum Label Correction* (CLC). With reference to the loss distribution from self-supervised learning, CLC identifies and corrects noisy labels utilizing curriculum learning. Our experimental results verify that CLC shows outstanding performance especially in a harshly noisy condition, 91.06% test accuracy on CIFAR-10 at a noise rate of 0.8. Code is available at https://github.com/LJY-HY/CLC.**

***Index Terms*—Noisy Label, Curriculum Learning, Self-Supervision**

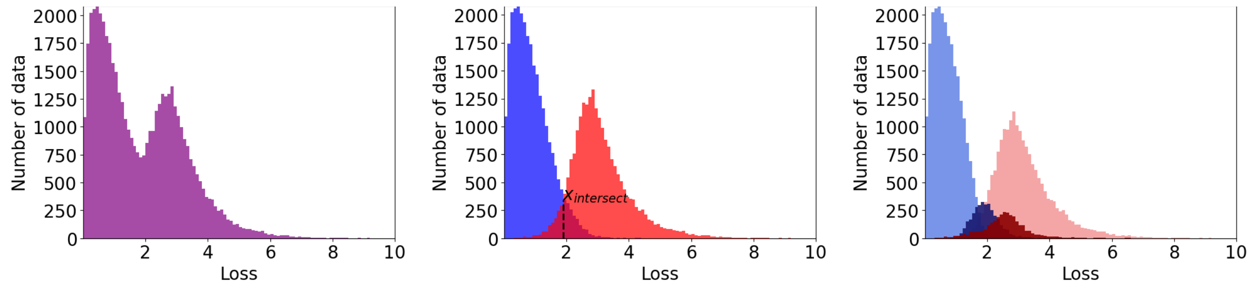
I. INTRODUCTION

The availability of a huge amount of data is one of the deep learning breakthroughs. However, unless the data is properly labeled, it is very challenging to properly train a neural network. Labeling huge amount of training data is a time-consuming and labor-intensive task. In particular, many publicly-available train datasets contain lots of mislabeled data, often referred to as noisily-labeled data. Without making efforts to make the training method robust against such noisily-labeled data, it is almost impossible for deep learning models to achieve consistently high accuracy. The authors of [1] claim that when the ratio of mislabeled data (noise rate) increases, the performance of a neural network often drops drastically let alone the training loss value takes longer to converge. Several methods that are robust against the label noise have been proposed. Many studies focused on disentangling the distribution between correctly-labeled data (*clean samples*) and noisily-labeled data (*noisy samples*). However, they often failed to show good performance largely owing to the concern that the threshold to distinguish noisy samples from clean samples was inappropriately set, resulting in many false negatives and false positives. Some papers have mitigated this adversity by re-weighting data samples [2], [3], but these approaches either require external meta-data or have to rely on heuristics.

In this paper, instead of disentangling the distribution explicitly, we propose a novel approach to correct noisy labels based on the following two observations:

- In general, neural networks are capable of distinguishing clean samples from noisy ones well in relatively-early stages of the training [3]. In Figure 1(b), the loss distribution of the CNN trained by self-supervised learning with a noise rate 0.5 after ten epochs also shows a similar aspect. This distribution demonstrates that the clean data have significantly lower loss values than the noisy ones. Specifically, there are two peaks where the left peak shows the loss distribution of clean data and the right one shows that of noisy data.
- When the correct label is derived from self-supervised trained neural network, the confidence level is typically high. Otherwise, the confidence level is relatively low. The high confidence level typically implies that the loss value may be either very small or very large. Therefore, when a correct label is derived with a high confidence level, (*easy*), the loss value may be either very small or very large. On the other hand, if the correct label is not derived (*hard*), the confidence level should be low, and therefore, the loss values are distributed around the center of the loss distribution.

These two observations suggest that there are two important perspectives in correcting noisy labels; the level of confidence (*easy* or *hard*) and the loss distribution. In other words, the level of confidence and the size of the training loss value are mutually independent of each other. In this paper, we claim that utilizing the confidence level should be useful to correct the noisy labels. To take the easiness and the hardness into the consideration, the proposed method employs a method called curriculum learning where the training starts with easy data and deals with hard data later by gradually increasing the level of difficulty. Specifically, we correct the noisy labels from the easiest data to the hardest ones similarly to curriculum learning. Correcting the noisy label according to the difficulty of the data lowers the risk of mislabeling. It also makes the network robust to the remaining label noise as the training progresses. On top of this attempt, to improve the accuracy of the label correction, we pay attention to the loss characteristic that the easy data should have two extreme loss values (either very small or very large) while the hard data have loss



(a) Total Distribution (b) Noisiness (c) Noisiness + Easiness

Fig. 1. Loss distribution of CIFAR-10 with a noise rate of 0.5 at the 1^{st} training epoch. (a) Total loss distribution of dataset(purple). (b) Loss distribution based on whether data is clean(blue) or noisy(red). The point where the two distributions intersect is marked as $x_{intersect}$. (c) Fine-grained loss distribution not only based on noisiness but also on easiness; Clean and easy data(pale blue) has the smallest loss average. Clean and hard data(dark blue) has the second smallest loss average. Noisy and hard data(dark red) has relatively bigger loss average. Noisy and easy data(pale red) has the largest loss on average.

values of the medium size as demonstrated in Figure 1(c). In this paper, we claim that by considering both the confidence level and the loss distribution, noisy labels can be effectively corrected.

The main contributions of this paper are summarized as follows:

- We apply a curriculum learning to correct the presumably-noisy labels. Some studies also utilized curriculum learning [2], [4] to solve noisy-label problem. They regarded correctly-labeled data as easy data while noisily-labeled data as hard. In contrast, we adopt curriculum learning differently as the method to determine the order of correcting the noisy labels.
- The proposed method significantly outperforms the compared methods when the noise rate is high, 0.6 or higher.
- The proposed method requires neither additional data for meta-learning nor additional model for robust training against noisy labeling.

The rest of this paper is organized as follows. In Section II, related studies are presented briefly. Section III introduces the motivation of this study with some background information. In Section V, the overall process of our method is described. In Section VI, experimental setting is described, and the performance of our method is compared with other existing works. Section VII concludes this paper.

II. Related Work

There are three types of methods for the robust training with noisy labels. The first approach assumes a small subset of training data has clean labels and semi-supervised learning with weak supervision is conducted. Hendrycks et al. [5] proposed a loss correction method by utilizing a small set of trusted labels to train a classifier with noisy labels. Zheng et al. [6] proposed a method where a meta-model corrects noisy labels while the main model is trained by data with clean labels.

The second type is to develop a noise-tolerant algorithm without any other data with clean labels. Kim et al. [7] proposed an indirect learning method called Negative Learning to decrease the amount of incorrect information. Zhang et al. [8] attempted to train a neural network on convex combinations of pairs of examples and their labels. Foret et al. [9] utilized the sharpness information of the loss value to improve generalization performance.

The last type is to identify noisily-labeled data and improve performance by either fixing the labels or excluding such data from training. Reed et al. [10] dealt with noisy and incomplete labeling by augmenting the prediction objective with a notion of consistency. Arazo et al. [3] proposed a beta mixture model to estimate the probability that the data is incorrectly labeled and to correct the loss value by relying on the network prediction. Xiao et al. [11] modeled the relationship among images, class labels, and label noises with a probabilistic graphical model and integrated it into an end-to-end deep learning system. Jiang et al. [2], [4] proposed a method of re-weighting data guided by a MentorNet, which informs the data is noisy or not. Each assigned weight to the data is considered when calculating the loss value for training a StudentNet.

Our proposed method can be regarded as one of the last-type approaches. We improve the network performance by correcting the label of the noisy data. Previous methods [2]–[4], [11] focused on separating clean data from noisy ones based on the observation that loss values diverge in proportion to the noisiness. However, our proposed method leverages not only the noisiness but also the easiness of the data. Thereby, the risk of erroneous modification of the data is reduced, resulting in a better performance.

III. Background

A. Self-Supervision

Generally, supervised learning has the advantage of achieving high performance. However, labeling a large amount of training data is error-prone and time-consuming. Therefore, the risk of mislabeling is pretty high. Self-supervised learning has emerged as a solution to overcome such mislabeling. Self-supervised learning typically defines a set of specific pretext tasks, which can be solved without any labels, such as context prediction [12], solving jigsaw puzzles [13], and contrastive learning [14]. By training the network to solve

TABLE I

COMPARISON OF THE TEST ACCURACY(%) OF CIFAR-10 BETWEEN THREE DIFFERENT NEURAL NETWORK SETTINGS. THE ENCODERS AND THE CLASSIFIERS OF EACH SETTING ARE TRAINED WITH THE DATA STATE(NOISY/CLEAN) SHOWN IN THE TABLE.

Dataset			CIFAR-10 Noise Rate(%)			
Settings	Encoder	Classifier	0.0	0.4	0.6	0.8
Setting 1	Noisy	Noisy	95.65	82.12	76.40	65.55
Setting 2	Noisy	Clean	95.65	86.30	81.94	73.99
Setting 3	Clean	Noisy	95.65	95.44	95.55	95.30
SimCLR	Self-Supervised	Noisy	93.00	89.53	86.14	70.29

these tasks, the network encoder can extract image representation appropriately and perform downstream tasks such as image classification. Among many different self-supervision training methods, we use the contrastive learning [14] to train the encoder.

In our method, unlike other studies that train the encoder and the classifier simultaneously, the encoder is trained with self-supervised learning first, and then the classifier is trained later, which is an incompatible training procedure with other approaches.

B. Curriculum Learning

Curriculum Learning (CL) was proposed by Bengio et al. in 2009 [15]. CL formalizes the cognitive process of humans and animals that learns more accessible aspects of a task first and then gradually increases the difficulty level. By utilizing CL, better generalization performance and faster optimization can be achieved. Some studies have adopted this CL paradigm to solve the noisy-labeling problem. In [2], clean samples are regarded as *easy* while noisy samples are regarded as *hard*. As the training progresses, the level of difficulty that the neural network can tolerate increases. In this paper, the concept of curriculum learning is applied differently. As mentioned earlier, in the proposed method, curriculum learning is applied to determine the order of correcting the noisy labels. The easiness is determined by the criterion that whether the network generates a correct label with a high confidence level. By correcting the label of easy data first, the proportion of clean data gradually increases. Therefore, the performance of the network is improved in proportion to the increasing ratio of clean data.

IV. SELF-SUPERVISED LEARNING ON NOISY LABEL

Unlike other methods that train their encoder and classifier concurrently, CLC trains a neural network in two steps. The encoder in CLC is trained first without a classifier using the self-supervised learning method of SimCLR [14]. Then, supervised learning in conjunction with the proposed label correcting algorithm is applied to train a classifier. To verify the effectiveness of the self-supervised learning method for training the encoder, we constructed three experimental settings with an additional setting (SimCLR) as follows:

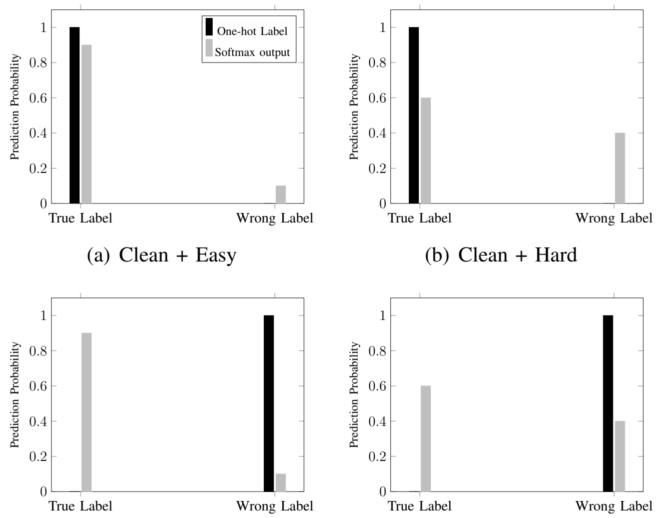
- *Setting 1* : Both encoder and classifier are trained with noisy data concurrently.
- *Setting 2* : With an encoder trained with noisy data, a classifier is retrained with clean data after initialization. The encoder is frozen while retraining the classifier.
- *Setting 3* : With an encoder trained with clean data, a classifier is retrained with noisy data after initialization. The encoder is frozen while retraining the classifier.
- *SimCLR* : An encoder is trained with self-supervised learning first. A classifier is trained with noisy data in conjunction with the pre-processed encoder.

Supervised learning with a cross-entropy loss function is applied for *Setting 1,2* and *3*. Self-supervised learning is applied to train the encoder of *SimCLR*. Noisy samples are used when training the noisy part of the networks. The labels of samples are changed randomly with the noise rate probability.

Table I summarizes the performance comparison results. The performance results of the neural network of *Setting 1* show that the noisy samples degrades the accuracy considerably as the accuracy of 95.65% with no noisy samples drops to 65.55% with a noise rate of 0.8%. The neural networks of *Setting 2* and *3*, which are partially trained with clean data, also suffer from some accuracy drop owing to the noisy data, but the degrees of the accuracy drop are much less than that of *Setting 1*. The degree of the accuracy drop of *Setting 2* was more severe than that of *Setting 3*, from which we can infer that the performance of a neural network depends more on an encoder rather than a classifier. Unlike *Setting 2* and *3*, however, no access is allowed to the clean data in the noisy label task. Therefore, considering the importance of training the encoder properly, we examine the performance of the neural network in which an encoder is trained with self-supervised learning. Compared to *Setting 3*, the results of *SimCLR* suggest that replacing an encoder with a self-supervised encoder should be effective to mitigate the accuracy drop.

V. CURRICULUM LABEL CORRECTION

In this paper, we propose a label correction method called Curriculum Label Correction (CLC). By referring to the loss distribution, CLC determines whether the data label is noisy or clean. That is, the data with a large loss value is very likely



(a) Clean + Easy (b) Clean + Hard

(c) Noisy + Easy (d) Noisy + Hard

Fig. 2. Distribution differences between the output of a network and the one-hot label of data for four different cases: Clean+Easy, Clean+Hard, Noisy+Easy, Noisy+Hard. The output of the network is colored in light gray. It is calculated via softmax with a sum of 1. The label of the data is colored in black. It is represented as a one-hot vector and has a value of 1 for only one label.

to be noisy. Then, as mentioned above, easy data are corrected first, and hard ones are corrected later as in CL.

Given a set of data samples $D = \{x, y\}^M$, the loss value of mini-batch $D_m = \{x, y\}^m$ is calculated as:

$$l_k^m(x) = -\frac{1}{m}\sum_{i=1}^{m} y_i \log Softmax(P_k(x_i)) \tag{1}$$

where M and m denote the size of the dataset and that of the mini-batch, respectively, and l_k^m is derived from calculating the cross-entropy loss of the softmax output of a neural network P_k at training epoch k. The loss distribution for mini-batch D_m implies that the loss of clean data tends to be small and the loss value of noisy data tends to be big. Therefore, as shown in Figure 1(b), we assume that the loss distribution of both clean and noisy data follow a Gaussian distribution and each distribution has its mean and standard deviation μ_1, σ_1 and μ_2, σ_2, respectively:

$$\begin{aligned} f_{clean}(x) &= \frac{1-\alpha}{\sigma_1\sqrt{2\pi}} \exp\Big(-\frac{1}{2}\Big(\frac{x-\mu_1}{\sigma_1}\Big)^2\Big) \\ f_{noisy}(x) &= \frac{\alpha}{\sigma_2\sqrt{2\pi}} \exp\Big(-\frac{1}{2}\Big(\frac{x-\mu_2}{\sigma_2}\Big)^2\Big) \end{aligned} \tag{2}$$

where α denotes the noise rate. Therefore, we can approximate the ratio of noisy to clean data $g(x)$ at point x as follows:

$$g(x) = \frac{\alpha}{1-\alpha}\frac{\sigma_1}{\sigma_2} \exp\Big(-\frac{(x-\mu_2)^2}{2\sigma_2^2} + \frac{(x-\mu_1)^2}{2\sigma_1^2}\Big) \tag{3}$$

and the derivative of $g(x)$ with respect to x is:

$$\frac{dg}{dx} = g(x)\Big(-\frac{x-\mu_2}{\sigma_2^2} + \frac{x-\mu_1}{\sigma_1^2}\Big) \tag{4}$$

For simplicity, we assume that standard deviations σ_1 and σ_2 are equal so that $g(x)$ is dependent only to α and x. Here, we can see that when the ratio of noisy samples to clean samples becomes larger than one, it grows exponentially after point $x_{intersect}$, where two distributions intersect as depicted in Figure 1(b). Therefore, it should be noted that the false-positive rate should decrease as the threshold, the minimum value to be judged to be noisy, is set higher.

In CLC, the data is classified into one of the four categories, not just the *clean* and *noisy* data, but also the combinations of *clean/noisy* and *easy/hard*. As mentioned above, *easy/hard* denotes the level of confidence of the output of a network, *easy* means the confidence level is high, and *hard* does the opposite. When a clean sample x_{clean} passes through the network P_k, it will have a relatively lower loss value than that of noisy data x_{noisy}. Among clean samples, the easy one, as shown in Figure 2(a), would have a loss value close to 0. Hard data, on the other hand, would result in a bigger loss value than the easy data as shown in Figure 2(b). In case of noisy data x_{noisy}, the tendency of the loss values is reversed. As the confidence of the data increases, the loss value grows bigger as shown in Figure 2(c), and as it decreases, the loss value gets smaller as shown in Figure 2(d). It is mainly because the one-hot label is assigned to the wrong one. These tendencies eventually result in the loss distribution as shown in Figure 1(c). Figure 1(c) illustrates the loss distribution of the four data categories: Clean and easy data (pale blue) ; clean and hard data (dark blue) ; noisy and easy data (pale red) ; noisy and hard data (dark red).

As mentioned above, the data with a low loss value has a high probability of being clean. Therefore, no additional processing is needed. In contrast, the initial label y_i of the data with a high loss value is supposed to be replaced with a new label. In CLC, the new label is the class of which softmax value is highest, and it turns out that this label-correction method achieves an extremely low false-positive rate. In practice, label correction is conducted for every top τ percentile of mini-batch D_m at every mini-batch training step. As the noisy labels are corrected as the mini-batch training progresses, $g(x)$, the ratio of noisy samples to clean ones computed by Equation 3, decreases as the noise rate α decreases. However, at the same time, the false-positive ratio also increases after the correction. To deal with the adverse effects of the false positives, we adjust the threshold τ to keep the false-positive ratio sufficiently low. We decrease the value of τ in proportion to the square of the completion percentage while training the network.

Algorithm 1 describes the mini-batch training procedure in CLC: (Step 1) decrease label-correction percentile τ by the amount that is proportional to the square of the training completion percentage, (Step 2) compute the loss value of mini-batch D_m, (Step 3), set threshold loss l_{th} to be located at top τ-percentile of the loss values set $\{l(x_i, y_i)\}$, (Step 4-6) correct the labels of data with the loss values greater than l_{th}, and (Step 8) compute the loss value of the mini-batch D_m with corrected labels.

TABLE II
COMPARISON OF TEST ACCURACY(%) ON CIFAR-10 AT VARIOUS SYMMETRIC NOISE RATES. 'NR' MEANS THE PERFORMANCE FIGURE IS NOT REPORTED IN THE RELATED PAPER.

Method	CIFAR-10(%)				CIFAR-100(%)			
	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
Cross-Entropy	84.05	71.38	49.45	25.13	56.33	42.95	24.68	8.98
M-Correction	94.00	92.80	90.30	74.10	73.70	70.10	59.50	39.50
MentorNet	92.00	91.20	74.20	60.00	73.50	68.50	61.20	35.50
MSLC	93.46	91.42	87.39	69.87	72.51	68.98	60.81	24.32
MixUp	94.00	91.50	86.80	76.90	73.90	66.80	58.80	40.10
NLNL	94.23	92.43	88.32	NR	71.52	66.39	56.51	NR
MentorMix	**95.60**	**94.20**	91.30	81.00	**78.60**	**71.30**	**64.60**	41.20
CLC(ours)	92.34	92.26	**91.67**	**90.98**	68.62	66.44	**64.27**	**58.59**

Algorithm 1: Curriculum Label Correction

Input : *preprocessed model* P; *mini batch* D_m; *hyperparameter* τ
Output: *the loss of the minibatch*

```
τ_epoch ← τ × (1 − progress²)
For every (x_i, y_i) in D_m, compute l(x_i, y_i)
Set l_th(D_m) to be the (1-τ_epoch)-th percentile of the
  loss {l(x_i, y_i)}
for (x_i, y_i) do
    if l(x_i, y_i) > l_th(D_m) then
        y_i ← arg max(P(x_i))
    end
end
Stop Gradient
For every (x_i, y_i) in D_m, compute l(x_i, y_i) return
  (1/|D_m|) Σ_{i=1}^{D_m} l_i
```

VI. Experiment Results

A. Datasets and noisy label settings

We evaluated the performance of the proposed method, CLC on two image recognition datasets, CIFAR-10 and CIFAR-100 [16]. Each dataset was used for pre-training and performance evaluation. Both CIFAR-10 and CIFAR-100 contain 50,000 images for training and 10,000 images for testing. All the images have the size of 32 $\times$ 32. To generate samples with noisy labels, the following two methods were used: *symmetric* and *asymmetric*. The symmetric method corrupts the true label y to all possible classes y' with a uniform probability of $\frac{\rho}{c}$, where ρ means the noise rate and c denotes the number of classes, thereby maintaining its original label with probability $1-\rho$. Hence, the corrupted label may happen to be the original label; the original label has a probability of $1-\rho+\frac{\rho}{c}$ to stay clean. The asymmetric method flips the true label y into another random class y' with probability ρ. Once the flip is decided based on the probability ρ, the new label is randomly selected. Therefore, the number of each class is unbalanced, unlike the original dataset.

B. Implementation Details

We conducted experiments on the ResNet-50 [17] and MobileNet V2 [18] network which is composed of an encoder that extracts features from images and a fully connected (FC) layer that classifies the input. To make the dataset to be close to the real situation where we do not know whether the labeling is accurate or not, we train the whole network in two steps. We first train the encoder from scratch or load from ImageNet [19] *pre-trained* checkpoint. After training the encoder, the parameters of the encoder are fixed lest they should be updated, and the FC layer is trained with a dataset with noisy labels.

The encoder is trained from scratch on CIFAR-10 / CIFAR-100 with the SimCLR [14], which is an efficient contrastive learning method. We train the encoder by Stochastic Gradient Descent (SGD) with a momentum of 0.9, a weight decay of 0.0001, a learning rate of 0.5, a batch size of 256, and a cosine learning rate scheduler which degrades the learning rate according to the cosine function. The maximum number of iteration of cosine learning rate scheduler is equal to the total number of training iteration. Augmentation for self-supervised learning is composed of RANDOM RESIZING, RANDOM CROPPING, RANDOM HORIZONTAL FLIP, COLOR JITTERING, AND RANDOM GRAYSCALE.

To utilize the *pre-trained* encoder, we load the encoder part from the ImageNet pre-trained checkpoint. We freeze the first 75-percentile layers of the encoder, and the rest are set as trainable to perform fine-tuning. In Table II, IV, the performance of CLC, which uses a pre-trained encoder, is summarized.

A single FC layer is used as the classifier for ResNet-50. It is trained for 100 epochs by SGD with a momentum of 0.9, a weight decay of 0, a learning rate of 1.0, and a batch size of 256. Classifier for MobileNet V2 consists of a dropout layer followed by a single FC layer. It is trained for 100 epochs by SGD with momentum of 0.9, a learning rate of 0.01, dropout rate of 0.2, a weight decay of 0.01, and a batch size of 256. Augmentation for training the classifier is composed of RANDOM RESIZE, RANDOM CROPPING, and RANDOM HOR-

TABLE III
TEST ACCURACY (%) RESULTS OF ABLATION STUDY ON CIFAR-10 AND CIFAR-100.

Dataset	CIFAR-10(%)				CIFAR-100(%)			
Noise Rate	0.2	0.4	0.6	0.8	0.2	0.4	0.6	0.8
CLC-S	92.34	92.26	91.67	90.98	68.62	66.44	64.27	58.59
CLC-S w/o *quadratic*	92.55	92.21	91.70	90.85	68.22	66.72	64.19	58.62
CLC-S w/o *label correction*	91.26	89.37	85.74	71.60	66.45	63.75	59.71	51.19
CLC-S w/o *self-supervision*	71.32	66.54	17.51	10.00	43.28	34.80	3.60	8.60
CLC-P	92.62	92.30	91.80	91.06	68.71	66.99	64.38	59.67
CLC-P w/o *quadratic*	92.54	92.31	91.79	90.89	68.49	66.82	64.04	59.53
CLC-P w/o *label correction*	91.25	89.90	86.56	72.29	66.80	63.96	59.97	52.02
CLC-P w/o *fine-tuning*	60.39	56.88	50.97	35.14	30.11	22.80	14.49	7.59

IZONTAL FLIP. All experiments are conducted three times, and then the averages are computed. The experiments are carried out with PyTorch 1.6 on an NVIDIA TITAN RTX GPU.

C. Evaluation Results

1) Brief descriptions of compared methods: We compared CLC with previously-reported methods on symmetric and asymmetric noisy labels with respect to various noise rates. The compared methods are: CROSS-ENTROPY is a mere training procedure through cross-entropy loss without any additional processing; M-CORRECTION [3] approximates a Beta distribution Mixture Model (BMM) and re-weights the samples according to the BMM; MENTORNET [2] trains the main model with curriculum learning where clean data is regarded as easy while noisy data as hard. To distinguish clean samples from noisy ones, MentorNet, another separate neural network, is pre-trained using meta-data; MSLC [20] trains a model through meta-learning with noise-free metadata. The network adaptively obtains rectified soft labels; MIXUP [8] conducts the convex combinations of pairs of examples and their labels. Mixed data is used for training; NLNL [21] decreases the risk of providing incorrect information through so-called negative learning, which trains CNN by providing a complementary label that cannot be the correct label; MENTORMIX [4] applies MixUp to MentorNet; F-CORRECTION [22] estimates the confusion matrix, which informs how likely the label corruption occurs for each class label. Then, it provides an end-to-end training framework; META-LEARNING [23] proposes a noise-tolerant training algorithm that simulates actual training by generating synthetic noisy labels first.

TABLE IV
COMPARISON OF TEST ACCURACY(%) ON CIFAR-10 WITH OTHER METHODS AT 40% ASYMMETRIC NOISE.

Method	CIFAR-10(%)
	0.4
Cross-Entropy	85.00
F-Correction	87.20
M-Correction	87.40
Meta-Learning	89.20
NLNL	89.86
CLC	**90.15**

2) Performance comparison results: Table II shows the results for various symmetric noise ratios ranging from 20% to 80% for ResNet-50 in CIFAR-10 and CIFAR-100. CLC shows state-of-the-art performance at high noise rates such as 60% or 80%. However, CLC is not as good as recently-proposed methods such as MENTORMIX at 20% and 40% noise rates. This mediocre performance is mainly because CLC is based on the encoders trained with self-supervised learning that limits its peak performance to 93% and 70% for CIFAR-10 and CIFAR-100, respectively. Therefore, as the performance of the self-supervised learning method on which CLC is based improves, the performance of CLC will improve correspondingly.

CLC with MobileNet V2 shows lower performance than with ResNet-50 due to the poor generalization performance of the neural network. At noise rates of 20% and 40%, the test accuracy of CLC is 87.56% and 85.11% for CIFAR-10, 66.18% and 58.37% for CIFAR-100. Nevertheless, at 60% and 80% noise rates, CLC with MobileNet V2 performs better than some other methods. At noise rates of 60% and 80%, test accuracy of CLC for CIFAR-10 is 80.42% and 70.26%, which is better than MentorNet. For CIFAR-100, at 60% and 80% noise rates, CLC outperforms MentorNet and MSLC with 53.29% and 37.56% test accuracy, respectively.

The results on CIFAR-10 with asymmetric noise are summarized in Table IV. We used 20% and 40% of asymmetric noise because classifying certain classes with asymmetric noise larger than 50% is practically impossible. As shown in Table IV, CLC outperforms other compared methods at 40% of asymmetric noise. For asymmetric noise of 20%, the accuracy improved from 89.77% to 91.83% with CLC compared to the neural network trained with Cross-Entropy loss.

3) Ablation Study: To understand what makes CLC effective better, some of the CLC features is removed, and how much the removal will affect the performance is measured. The results summarized in Table III may be analyzed as follows:

- In this paper, we proposed the two types of CLC; CLC-P denotes the CLC method that uses a fine-tuned encoder that is pre-trained with ImageNet; CLC-S denotes the CLC method that uses an encoder that is trained from scratch via self-supervised learning on CIFAR-10 and CIFAR-100. The performance of CLC-P outperforms CLC-S in all cases, but not as noticeable.
- Decreasing the τ value by the amount that is proportional to the square of the learning completion percentage (marked as *quadratic* in Table III) does not significantly impact CLC-S. However, in case of CLC-P, although small, there is a performance improvement of about 0.15% on average.
- Label correction plays an essential role in improving the test accuracy with harsh noise rates. If the label correction is not included, the test accuracy drops about 20% for CIFAR-10 at 80% noise rate for CLC-P and CLC-S.
- Fine-tuning is essential for CLC-P, which uses an ImageNet pre-trained encoder. Without fine-tuning, even if label correction is conducted, the test accuracy drops significantly.
- CLC-S without self-supervision, which is trained in the same way as the CROSS-ENTROPY in Table II, shows a worse test accuracy than CROSS-ENTROPY even though the label correction is applied.

VII. CONCLUSION

Labeling is an important task for training deep neural networks. However, labeling is a tedious and error-prone task to result in a high risk of mislabeling. This mislabeling will severely degrade the performance. In this paper, we proposed a noise-tolerant learning algorithm termed Curriculum Label Correction (CLC). CLC adopted curriculum learning to correct the noisy label efficiently. While several existing methods set a threshold to separate the clean data from the noisy data, CLC classified samples as easy or hard ones with reference to the loss distribution to make use of curriculum learning. Compared with other existing methods, CLC showed better performance on CIFAR-10 at a high noise rate, 91.85% on 0.6 noise rate, and 91.10% on 0.8 noise rate. Finally, as the performance of self-supervised learning gets improved, we expect our method to improve accordingly as well.

ACKNOWLEDGMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2020-0-01304, Development of Self-learnable Mobile Recursive Neural Network Processor Technology)

REFERENCES

[1] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.

[2] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2304–2313.

[3] E. Arazo, D. Ortego, P. Albert, N. O'Connor, and K. McGuinness, "Unsupervised label noise modeling and loss correction," in *International Conference on Machine Learning*. PMLR, 2019, pp. 312–321.

[4] L. Jiang, D. Huang, M. Liu, and W. Yang, "Beyond synthetic noise: Deep learning on controlled noisy labels," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4804–4815.

[5] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, "Using trusted data to train deep networks on labels corrupted by severe noise," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/file/ad554d8c3b06d6b97ee76a2448bd7913-Paper.pdf

[6] G. Zheng, A. H. Awadallah, and S. Dumais, "Meta label correction for noisy label learning," in *AAAI 2021*, February 2021. [Online]. Available: https://www.microsoft.com/en-us/research/publication/meta-label-correction-for-noisy-label-learning/

[7] Y. Kim, J. Yim, J. Yun, and J. Kim, "Nlnl: Negative learning for noisy labels," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[8] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[9] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," in *International Conference on Learning Representations*, 2020.

[10] 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.

[11] 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*, 2015, pp. 2691–2699.

[12] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1422–1430.

[13] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European conference on computer vision*. Springer, 2016, pp. 69–84.

[14] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[15] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[16] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research)." [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2016. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[20] Y. Wu, J. Shu, Q. Xie, Q. Zhao, and D. Meng, "Learning to purify noisy labels via meta soft label corrector," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 10 388–10 396.

[21] Y. Kim, J. Yim, J. Yun, and J. Kim, "Nlnl: Negative learning for noisy labels," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 101–110.

[22] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1944–1952.

[23] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanhalli, "Learning to learn from noisy labeled data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5051–5059.