PRESTOPPING: HOW DOES EARLY STOPPING HELP GENERALIZATION AGAINST LABEL NOISE?

Anonymous authors
Paper under double-blind review

ABSTRACT

Noisy labels are very common in real-world training data, which lead to poor generalization on test data because of overfitting to the noisy labels. In this paper, we claim that such overfitting can be avoided by “early stopping” training a deep neural network before the noisy labels are severely memorized. Then, we resume training the early stopped network using a “maximal safe set,” which maintains a collection of almost certainly true-labeled samples at each epoch since the early stop point. Putting them all together, our novel two-phase training method, called Prestopping, realizes noise-free training under any type of label noise for practical use. Extensive experiments using four image benchmark data sets verify that our method significantly outperforms four state-of-the-art methods in test error by 0.4–8.2 percent points under existence of real-world noise.

1 INTRODUCTION

By virtue of massive labeled data, deep neural networks (DNNs) have achieved a remarkable success in numerous machine learning tasks, such as image classification (Krizhevsky et al., 2012) and object detection (Redmon et al., 2016). However, owing to their high capacity to memorize any label noise, the generalization performance of DNNs drastically falls down when noisy labels are contained in the training data (Jiang et al., 2018; Han et al., 2018; Song et al., 2019). In particular, Zhang et al. (2017) have shown that a standard convolutional neural network (CNN) can easily fit the entire training data with any ratio of noisy labels and eventually leads to very poor generalization on the test data. Thus, it is challenging to train a DNN robustly even when noisy labels exist in the training data.

A popular approach to dealing with noisy labels is “sample selection” that selects true-labeled samples from the noisy training data (Jiang et al., 2018; Ren et al., 2018; Han et al., 2018; Yu et al., 2019; Song et al., 2019). Here, \((1-\tau)\times 100\%\) of small-loss training samples are treated as true-labeled ones and then used to update a DNN robustly, where \(\tau \in [0, 1]\) is a noise rate. This loss-based separation is well known to be justified by the memorization effect (Arpit et al., 2017) that DNNs tend to learn easy patterns first and then gradually memorize all samples. In practice, Han et al. (2018) empirically proved that training on such small-loss samples yields a much better generalization performance under artificial noise scenarios.

Despite its great success, Song et al. (2019) have recently argued that the performance of the loss-based separation becomes considerably worse depending on the type of label noise. For instance,
In symmetric noise (Figure 1(a)), the loss-based approach well separates true-labeled samples from false-labeled ones because many true-labeled ones exhibit smaller loss than false-labeled ones. On the other hand, in pair and real-world noises (Figures 1(b) and 1(c)), many false-labeled samples are misclassified as true-labeled ones because the two distributions overlap closely; this overlap confirms that the loss-based separation still accumulates severe label noise from many misclassified cases, especially in real-world noise or pair noise which is regarded as more realistic than symmetric noise [Ren et al., 2018; Yu et al., 2019]. This limitation definitely calls for a new approach that supports any type of label noise for practical use.

In this regard, as shown in Figure 2(a), we thoroughly investigated the memorization effect of a DNN on the two types of noises and found two interesting properties as follows:

- **A noise type affects the memorization rate for false-labeled samples:** The memorization rate for false-labeled samples is faster with pair noise than with symmetric noise. That is, the red portion in Figure 2(a) starts to appear earlier in pair noise than in symmetric noise. This observation supports the significant overlap of true-labeled and false-labeled samples in Figure 1(b). Thus, the loss-based separation performs well only if the false-labeled samples are scarcely learned at an early stage of training, as in symmetric noise.

- **There is a period where the network accumulates the label noise severely:** Regardless of the noise type, the memorization of false-labeled samples significantly increases at a late stage of training. That is, the red portion in Figure 2(a) increases rapidly after the dashed line, in which we call the error-prone period. (See Section 3.2.1 for the details of estimating the error-prone period). We note that the training in that period brings no benefit. The generalization performance of “Default” deteriorates sharply, as shown in Figure 2(c).

Based on these findings, we contend that eliminating this error-prone period should make a profound impact on robust optimization.

In this paper, we propose a novel approach, called **Prestopping**, that achieves noise-free training based on the early stopping mechanism. Because there is no benefit from the error-prone period, **Prestopping** early stops training before that period begins. This early stopping effectively prevents a network from overfitting to false-labeled samples, and the samples memorized until that point are added to a maximal safe set because they are true-labeled (i.e., blue in Figure 2(a)) with high precision. Then, **Prestopping** resumes training the early stopped network only using the maximal safe set in support of noise-free training. Notably, our proposed merger of “early stopping” and “learning from the maximal safe set” indeed eliminates the error-prone period from the training process, as shown in Figure 2(b). As a result, the generalization performance of a DNN remarkably improves in both noise types, as shown in Figure 2(c).
As for the notion of network memorization, a sample \( x \) is defined to be memorized by a network if the majority of its recent predictions at time \( t \) coincide with the given label, as in Definition 3.1.

**Definition 3.1. (Memorized Sample)** Let \( \hat{y}_t = \Phi(x; \theta_t) \) be the predicted label of a sample \( x \) at time \( t \) and \( H^q_t(q) = \{\hat{y}_t, \hat{y}_{t+1}, \ldots, \hat{y}_{t+q}\} \) be the history of the sample \( x \) that stores the predicted labels of the recent \( q \) times, where \( \Phi \) is a neural network. Next, \( P(y|x, t; q) \) is formulated such that it provides

\[ P(y|x, t; q) = \frac{1}{|H^q_t(q)|} \sum_{x \in H^q_t(q)} \mathbb{1}(y = \hat{y}_t) \]

where \( \mathbb{1}(y = \hat{y}_t) \) is the indicator function that equals 1 if \( y = \hat{y}_t \) and 0 otherwise. This formulation is known as the **decoupling** strategy.

**REFERENCES**

1. Chang et al. (2017)
2. Han et al. (2018)
3. Malach & Shalev-Shwartz (2017)
4. Reed et al. (2015)
5. Song et al. (2019)
Under review as a conference paper at ICLR 2020

Figure 3: Early stop point estimated by ideal and heuristic methods when training DenseNet (L=40, k=12) on CIFAR-100 with two types of synthetic noises of 40%: (a) and (b) show the stop point derived by the ground-truth labels and the clean validation set, respectively.

The key idea of Prestopping is learning from a maximal safe set with an early stopped network. Thus, the two components of “early stopping” and “learning from the maximal safe set” respectively raise the questions about (Q1) when is the best point to early stop the training process? and (Q2) what is the maximal safe set to enable noise-free training during the remaining period?

3.2 Two Main Components and Phases

The key idea of Prestopping is learning from a maximal safe set with an early stopped network. Thus, the two components of “early stopping” and “learning from the maximal safe set” respectively raise the questions about (Q1) when is the best point to early stop the training process? and (Q2) what is the maximal safe set to enable noise-free training during the remaining period?

3.2.1 Question 1: Best Point to Early Stop

It is desirable to stop the training process at the point when the network (i) not only accumulates little noise from the false-labeled samples, (ii) but also acquires sufficient information from the true-labeled ones. Intuitively speaking, as indicated by the dashed line in Figure 3(a), the best stop point is the moment when label precision and label recall in Definition 3.2 cross with each other because it is the best trade-off between the two metrics. The period beyond this point is what we call the error-prone period because label precision starts decreasing rapidly.

If the ground truth \( y^* \) of a noisy label \( \tilde{y} \) is known, the best stop point can be easily calculated by these metrics.

**Definition 3.2. (Label Metrics)** Let \( \mathcal{M}_t \subseteq \mathcal{D} \) be a set of memorized samples at time \( t \). Then, label precision (LP) and recall (LR) \((\text{Han et al., 2018})\) are formulated as Eq. (4).

\[
\text{LP} = \frac{|(x, \tilde{y}) \in \mathcal{M}_t : \tilde{y} = y^*|}{|\mathcal{M}_t|}, \quad \text{LR} = \frac{|(x, \tilde{y}) \in \mathcal{M}_t : \tilde{y} = y^*|}{|(x, \tilde{y}) \in \mathcal{D} : \tilde{y} = y^*|} \quad (4)
\]

However, it is not straightforward to find the exact best stop point without the ground-truth labels. Hence, we present two practical heuristics of approximating the best stop point with minimal supervision. These two heuristics require either a small clean validation set or a noise rate \( \tau \) for the minimal supervision, where they are widely regarded as available in many studies \( \text{Veit et al., 2017; Ren et al., 2018; Han et al., 2018; Yu et al., 2019; Song et al., 2019} \).

- **Validation Heuristic:** If a clean validation set is given, we stop training the network when the validation error is the lowest. It is reasonable to expect that the lowest validation error is achieved near the cross point; after that point (i.e., in the error-prone period), the validation error likely increases because the network will be overfitted to the false-labeled samples. As shown in Figure 3(b), the estimated stop point is fairly close to the best stop point in both noise types.

\[\text{The Iverson bracket } [P] \text{ returns 1 if } P \text{ is true; 0 otherwise.}\]
Algorithm 1 describes the overall procedure of Prestopping with the validation heuristic, which is self-explanatory. First, the network is trained on the noisy training data \( \bar{D} \) in the default manner (Lines 3–6). During this first phase, the validation data \( V \) is used to evaluate the best point for the early stop, and the network parameter is saved at the time of the lowest validation error (Lines 7–8). Then, during the second phase, Prestopping continues to train the early stopped network for the remaining learning period (Lines 10–12). Here, the maximal safe set \( S_t \) at the current moment is retrieved, and each sample \( x \in S_t \cap B_t \) is used to update the network parameter. The mini-batch samples not included in \( S_t \) are no longer used in pursuit of robust learning (Lines 14–15).

- **Noise-Rate Heuristic:** If a noise rate \( \tau \) is known, we stop training the network when the training error reaches \( \tau \times 100\% \). If we assume that all true-labeled samples of \( (1-\tau) \times 100\% \) are memorized before any false-labeled samples of \( \tau \times 100\% \) (Arpit et al., 2017), this point indicates the cross point of label precision and label recall with their values to be all 1. This heuristic tends to perform worse than the validation heuristic because the assumption does not hold perfectly. We further discuss its quality in Appendix A.

3.2.2 **Question 2: Criterion of a Maximal Safe Set**

Because the network is early stopped at the (estimated) best point, the set of memorized samples at that time is quantitatively sufficient and qualitatively less noisy. That is, it can be used as a safe and effective training set to resume the training of the early stopped network without accumulating the label noise. Based on this intuition, we define a maximal safe set in Definition 3.3, which is initially derived from the memorized samples at the early stop point and gradually increased as well as purified along with the network’s learning progress. In each mini-batch \( B_t \), the network parameter \( \theta_t \) is updated using the current maximal safe set \( S_t \) as in Eq. (5), and subsequently a more refined maximal safe set \( S_{t+1} \) is derived by the updated network.

**Definition 3.3. (Maximal Safe Set)** Let \( t_{\text{stop}} \) be the early stop point. A maximal safe set \( S_t \) at time \( t \) is defined to be the set of the memorized samples of the network \( \Phi(x; \theta_t) \) when \( t \geq t_{\text{stop}} \). The network \( \Phi(x; \theta_t) \) at \( t \geq t_{\text{stop}} \) is the early stopped network, i.e., \( S_{t_{\text{stop}}} = M_{t_{\text{stop}}} \), and the network \( \Phi(x; \theta_{t+1}) \) at \( t > t_{\text{stop}} \) is obtained by Eq. (5).

\[
\theta_{t+1} = \theta_t - \alpha \nabla \left( \frac{1}{|B_t|} \sum_{x \in B_t} L(x, \hat{y}; \theta_t) \right) \tag{5}
\]

\[
B_t' = \{ x | x \in S_t \cap B_t \}
\]

Figure 4 shows the main advantage of learning from the maximal safe set during the remaining period. Even when the noise rate is quite high (e.g., 40%), this learning paradigm exploits most of true-labeled samples, in considering that the label recall of the maximal safe set was maintained over 0.81 in symmetric noise and over 0.84 in pair noise after the 80th epoch. Further, by excluding unsafe samples that might accumulate the label noise, the label precision of the maximal safe set increases rapidly at the beginning of the remaining period; it was maintained over 0.99 in symmetric noise and over 0.94 even in pair noise after the 80th epoch, which could not be realized by the small-loss separation (Han et al., 2018; Song et al., 2019).

4 **Algorithm Description**

4.1 **Main Algorithm: Prestopping with Validation Heuristic**

Algorithm 1 describes the overall procedure of Prestopping with the validation heuristic, which is self-explanatory. First, the network is trained on the noisy training data \( \bar{D} \) in the default manner (Lines 3–6). During this first phase, the validation data \( V \) is used to evaluate the best point for the early stop, and the network parameter is saved at the time of the lowest validation error (Lines 7–8). Then, during the second phase, Prestopping continues to train the early stopped network for the remaining learning period (Lines 10–12). Here, the maximal safe set \( S_t \) at the current moment is retrieved, and each sample \( x \in S_t \cap B_t \) is used to update the network parameter. The mini-batch samples not included in \( S_t \) are no longer used in pursuit of robust learning (Lines 14–15).
Algorithm 1 Prestopping with Validation Heuristic

INPUT: $\mathcal{D}$: data, $\mathcal{V}$: clean validation data, epochs: total number of epochs, $q$: history length
OUTPUT: $\theta_t$: network parameters

1: $t \leftarrow 1$; $\theta_t \leftarrow$ Initialize the network parameter;
2: $\theta_{\text{stop}} \leftarrow \theta_t$ /* The parameter of the stopped network */
3: for $i = 1$ to epochs do /* Phase I: Learning from a noisy training data set */
4: for $j = 1$ to $|\mathcal{D}|/|\mathcal{B}_t|$ do
5: Draw a mini-batch $\mathcal{B}_t$ from $\mathcal{D}$;
6: $\theta_{t+1} = \theta_t - \alpha \nabla \left( \frac{1}{|\mathcal{B}_t|} \sum_{x \in \mathcal{B}_t} \mathcal{L}(x, \hat{y}; \theta_t) \right)$; /* Update by Eq. (1) */
7: $\text{val}_\text{err} \leftarrow \text{Get\_Validation\_Error}(\mathcal{V}, \theta_t)$; /* A validation error at time $t$ */
8: if isMin($\text{val}_\text{err}$) then $\theta_{\text{stop}} \leftarrow \theta_t$; /* Save the network when $\text{val}_\text{err}$ is the lowest */
9: $t \leftarrow t + 1$;
10: $\theta_t \leftarrow \theta_{\text{stop}}$; /* Load the network stopped at $t_{\text{stop}}$ */
11: for $i = \text{stop\_epoch}$ to epochs do /* Phase II: Learning from a maximal safe set */
12: for $j = 1$ to $|\mathcal{D}|/|\mathcal{B}_t|$ do
13: Draw a mini-batch $\mathcal{B}_t$ from $\mathcal{D}$;
14: $\mathcal{S}_t \leftarrow \{x|\text{argmax}_y P(y|x,t;q) = \hat{y}\}$; /* A maximal safe set in Definition 3.3 */
15: $\theta_{t+1} = \theta_t - \alpha \nabla \left( \frac{1}{|\mathcal{S}_t \cap \mathcal{B}_t|} \sum_{x \in \mathcal{S}_t \cap \mathcal{B}_t} \mathcal{L}(x, \hat{y}; \theta_t) \right)$; /* Update by Eq. (5) */
16: $t \leftarrow t + 1$;
17: return $\theta_t, \mathcal{S}_t$;

4.2 Collaboration with Sample Refurbishment: Prestopping+

To further improve the performance of Prestopping, we introduce Prestopping+ that employs the concept of selectively refurbishing false-labeled samples in SELFIE (Song et al. 2019). In detail, the final maximal safe set $\mathcal{S}_{\text{end}}$ is retrieved from the first run of Prestopping, and then it is used as the true-labeled set for the next run of SELFIE with initializing the network parameter. Following the update principle of SELFIE, the modified gradient update rule in Eq. (6) is used to train the network. For each sample $x$ in the mini-batch $\mathcal{B}_t$, the given label $\hat{y}$ of $x$ is selectively refurbished into $y^{\text{refurb}}$ if it can be corrected with high precision. However, the mini-batch samples included in $\mathcal{S}_{\text{end}}$ are omitted from the refurbishment because their label is already highly credible. Then, the mini-batch samples in the refurbished set $\mathcal{R}_t$ are provided to update the network together with those in the final maximal safe set $\mathcal{S}_{\text{end}}$. Refer to [Song et al. 2019]’s work for more details.

5 Evaluation

Data Sets: To verify the superiority of Prestopping, we performed an image classification task on four benchmark data sets: CIFAR-10 (10 classes)$^4$ and CIFAR-100 (100 classes)$^4$, a subset of 80 million categorial images [Krizhevsky et al. 2014]; ANIMAL-10N (10 classes)$^6$, a real-world noisy data of human-labeled online images for confusing animals [Song et al. 2019]; FOOD-101N (101 classes)$^6$, a real-world noisy data of crawled food images annotated by their search keywords in the FOOD-101 taxonomy [Bassard et al. 2014; Lee et al. 2018]. We did not apply any data augmentation.

Noise Injection: As all labels in the CIFAR data sets are clean, we artificially corrupted the labels in these data sets using typical methods for the evaluation of synthetic noises [Ren et al. 2018; Han et al. 2018; Yu et al. 2019]. For k classes, we applied the label transition matrix $T$: (i) symmetric noise: $T_{ij} = \frac{1}{k-1}$ and (ii) pair noise: $T_{ij} = \tau \land \forall k \neq i, k \neq j, T_{ik} = 0$, where $T_{ij}$ is the probability of the true label $i$ being flipped to the corrupted label $j$ and $\tau$ is the noise rate. For the pair noise, the corrupted label $j$ was set to be the next label of the true label $i$ following the recent work [Yu et al. 2019].

https://www.cs.toronto.edu/~kriz/cifar.html
https://dm.kaist.ac.kr/datasets/animal-10n
https://kuanghui.research.att/food-101N
To evaluate the robustness on varying noise rates from light noise to heavy noise, we tested five noise rates $\tau \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$. In contrast, we did not inject any label noise into ANIMAL-10N and FOOD-101N because they contain real label noise estimated at 8.0% and 18.4% respectively (Lee et al. 2018; Song et al. 2019).

**Clean Validation Data:** Recall that a clean validation set is needed for the validation heuristic in Section 3.2.1. As for ANIMAL-10N and Food-101N, we did exploit their own clean validation data; 5,000 and 3,824 images, respectively, were included in their validation set. However, as no validation data exists for the CIFAR data sets, we constructed a small clean validation set by randomly selecting 1,000 images from the original training data of 50,000 images. Please note that the noise injection process was applied to only the rest 49,000 training images.

**Networks and Hyperparameters:** For the classification task, we trained DenseNet ($L=40, k=12$) and VGG-19 with a momentum optimizer. Specifically, we used a momentum of 0.9, a batch size of 128, a dropout of 0.1, and batch normalization. Prestopping has only one unique hyperparameter, the history length $q$, and it was set to be 10, which was the best value found by the grid search (see Appendix B.1 for details). The hyperparameters used in the compared algorithms were favorably set to be the best values presented in the original papers. As for the training schedule, we trained the network for 120 epochs and used an initial learning rate 0.1, which was divided by 5 at 50% and 75% of the total number of epochs.

**Algorithms:** We compared Prestopping and Prestopping+ with not only a baseline algorithm but also the four state-of-the-art robust training algorithms: Default trains the network without any processing for the label noise; Active Bias re-weights the backward loss of training samples based on their prediction variance; Co-teaching selects a certain number of small-loss samples to train the network based on the co-training mechanism (Blum & Mitchell 1998); Co-teaching+ is similar to Co-teaching, but its small-loss samples are selected from the disagreement set; SELFIE selectively refurbishes noisy samples and exploits them together with the small-loss samples. All the algorithms were implemented using TensorFlow 1.8.0 and executed using 16 NVIDIA Titan Volta GPUs. For reproducibility, we provide the source code at [https://bit.ly/2l3g9Jx](https://bit.ly/2l3g9Jx). In support of reliable evaluation, we repeated every task thrice and reported the average and standard error of the best test errors, which are the common measures of robustness to noisy labels (Chang et al. 2017; Jiang et al. 2018; Ren et al. 2018; Song et al. 2019).

### 5.1 Performance Comparison

Figures 5 and 6 show the test error of the seven training methods using two CNNs on two data sets with varying pair and symmetric noise rates, respectively. The results of Prestopping and Prestopping+ in Section 4 were obtained using the validation heuristic. See Appendix A for the results of the noise-rate heuristic. In order to highlight the improvement of Prestopping and Prestopping+ over the other methods, their lines are dark colored. Figure 7 shows the test error on two real-world noisy data sets with different noise rates. The best test errors are detailed in Tables 1 and 2 of Appendix C.

#### 5.1.1 Result with Pair Noise (Figure 5)

The performance trend in the two network architectures were similar with each other. In general, either Prestopping or Prestopping+ achieved the lowest test error in a wide range of noise rates on both CIFAR data sets. With help of the refurbished samples, Prestopping+ achieved a slightly better performance than Prestopping in CIFAR-10. However, an opposite trend was observed in CIFAR-100; this phenomenon was due to a large number of falsely corrected labels under the pair noise, especially when the number of classes is large (see Appendix B.2 for details). Although SELFIE achieved relatively lower test error among the existing methods, the test error of SELFIE was still worse than that of Prestopping. Co-teaching did not work well because many false-labeled samples were misclassified as clean ones; Co-teaching+ was shown to be even worse than Co-teaching despite it being an improvement of Co-teaching (see Section 5.3 for details). The test error of Active Bias was not comparable to that of Prestopping. The performance improvement of ours over the others increased as the label noise became heavier. In particular, at a heavy noise rate of 40%, Prestopping or Prestopping+ significantly reduced the absolute test error by 2.2 ppm–18.1 ppm compared with the other robust methods.

In FOOD-101N, we used a subset of the entire training data marked with whether the label is correct or not.
well separated from false-labeled ones by their small-loss criteria; hence, the dominance between Co-teaching and SELFIE was reversed so that the former slightly outperformed the latter.

5.1.3 Result with Real-World Noise (Figure 7)
Both Prestopping and Prestopping+ maintained their dominance over the other methods under real-world label noise as well. Prestopping+ achieved the lowest test error when the number of classes is small (e.g., ANIMAL-10N), while Prestopping was the best when the number of classes is large (e.g., FOOD-101N) owing to the difficulty in label correction of Prestopping+. (Thus, practitioners are recommended to choose between Prestopping and Prestopping+ depending on the number of classes in hand.) Specifically, they improved the absolute test error by $0.4pp$–$4.6pp$ and $0.5pp$–$8.2pp$ in ANIMAL-10N and FOOD-101N, respectively. Therefore, we believe that the advantage of our methods is unquestionable even in the real-world scenarios.

5.2 Convergence Analysis (Figure 8)
To verify that the error-prone period is eliminated, we plot the convergence of test error for the seven training methods. The test error of Default, Active Bias, and Co-teaching+ first reached their lowest values before the error-prone period, and then increased rapidly in the error-prone period. On the other hand, the remaining methods kept reducing the test error even in that period. Notably, Prestopping+
achieved the lowest test error at the end, owing to the successful merger of the advanced sample selection in Prestopping and the sample refurbishment in SELFIE. In addition, the performance of Prestopping was shown to be almost the same as and comparable to that of Prestopping+ in pair and symmetric noises, respectively. The test error of SELFIE also converged faster than those of the other existing methods, but it was still inferior to Prestopping+. Thus, these results confirm that our two methods effectively overcome the error-prone period.

5.3 Anatomy of Co-teaching+ (Figure 9)

Although Co-teaching+ is the latest method, its performance was worse than expected, as shown in Section 5.1. Thus, we looked into Co-teaching+ in more detail. A poor performance of Co-teaching+ was attributed to the fast consensus of the label predictions for true-labeled samples, especially when training a complex network. In other words, because two complex networks in Co-teaching+ start making the same predictions for true-labeled samples too early, these samples are excluded too early. As shown in Figures 9(a) and 9(b), the disagreement ratio with regard to the true-labeled samples dropped faster with a complex network than with a simple network, in considering that the ratio drastically decreased to 49.8% during the first 5 epochs. Accordingly, it is evident that Co-teaching+ with a complex network causes a narrow exploration of the true-labeled samples, and the selection accuracy of Co-teaching+ naturally degraded from 60.4% to 44.8% for that reason, as shown in Figure 9(c). Therefore, we conclude that Co-teaching+ may not suit a complex network.

6 Conclusion

In this paper, we proposed a novel two-phase training strategy for the noisy training data, which we call Prestopping. The first phase, “early stopping,” retrieves an initial set of true-labeled samples as many as possible, and the second phase, “learning from a maximal safe set,” completes the rest training process only using the true-labeled samples with high precision. Prestopping can be easily applied to many real-world cases because it additionally requires only either a small clean validation set or a noise rate. Furthermore, we combined this novel strategy with sample refurbishment to develop Prestopping+. Through extensive experiments using various real-world and simulated noisy data sets, we verified that either Prestopping or Prestopping+ achieved the lowest test error among the seven compared methods, thus significantly improving the robustness to diverse types of label noise. Overall, we believe that our work of dividing the training process into two phases by early stopping is a new direction for robust training and can trigger a lot of subsequent studies.
REFERENCES


Eran Malach and Shai Shalev-Shwartz. Decoupling “when to update” from “how to update”. In NeurIPS, pp. 960–970, 2017.


A  Prestopping with Noise-Rate Heuristic

Algorithm 2 describes the overall procedure of Prestopping with the noise-rate heuristic, which is also self-explanatory. Compared with Algorithm 1 only the way of determining the best stop point in Lines 7–9 has changed.

Algorithm 2 Prestopping with Noise-Rate Heuristic

INPUT: $\tilde{D}$: data, epochs: total number of epochs, $q$: history length, $r$: noise rate
OUT: $\theta_t$: network parameters

1. $t \leftarrow 1$; $\theta_1 \leftarrow$ Initialize the network parameter;
2. $\theta_{stop} \leftarrow \theta_1$; /* The parameter of the stopped network */
3. for $i = 1$ to epochs do /* Phase I: Learning from a noisy training data set */
   4. for $j = 1$ to $|\tilde{D}|/|B_i|$ do
      5. Draw a mini-batch $B_i$ from $\tilde{D}$;
      6. $\theta_{t+1} = \theta_t - \alpha \nabla \left( \frac{1}{|B_i|} \sum_{x \in B_i} L(x, \tilde{y}; \theta_t) \right)$; /* Update by Eq. (1) */
      7. train_err $\leftarrow$ Get_Training_Error($\tilde{D}$, $\theta_t$); /* A training error at time t */
      8. if train_err $\leq \tau$ then /* Save the network when train_err $\leq \tau$ */
         9. $\theta_{stop} \leftarrow \theta_t$; break;
   10. $t \leftarrow t + 1$;
   11. $\theta_t \leftarrow \theta_{stop}$; /* Load the network stopped at $t_{stop}$ */

12. for $i = stop_{epoch}$ to epochs do /* Phase II: Learning from a maximal safe set */
   13. for $j = 1$ to $|\tilde{D}|/|B_i|$ do
      14. Draw a mini-batch $B_i$ from $\tilde{D}$;
      15. $S_i \leftarrow \{x | \text{argmax}_y P(y|x; t, q) = \tilde{y} \}$; /* A maximal safe set in Definition 3.3 */
      16. $\theta_{t+1} = \theta_t - \alpha \nabla \left( \frac{1}{|S_i|} \sum_{x \in S_i} L(x, \tilde{y}; \theta_t) \right)$; /* Update by Eq. (5) */
      17. $t \leftarrow t + 1$;
   18. return $\theta_t, S_i$;

A.1 Result with Synthetic Noise (Figure 10)

To verify the performance of Prestopping and Prestopping+ with the noise-rate heuristic in Section 5.2.1 we trained a VGG-19 network on two simulated noisy data sets with the same configuration as in Section 5. Figure 10 shows the test error of our two methods using the noise-rate heuristic as well as those of the other five training methods. Again, the test error of either Prestopping or Prestopping+ was the lowest at most error rates with any noise type. The trend of the noise-rate heuristic here was almost the same as that of the validation heuristic in Section 5.1. Especially when the noise rate was 40%, Prestopping and Prestopping+ significantly improved the test error by 5.1pp–17.0pp in the pair noise (Figure 10(a)) and 0.3pp–17.3pp in the symmetric noise (Figure 10(b)) compared with the other robust methods.

![Figure 10: Best test errors using VGG-19 on two simulated noisy data sets with varying noise rates.](image)

A.2 Comparison with Validation Heuristic (Figure 11)

Figure 11 shows the difference in test error caused by the two heuristics. Overall, the performance with the noise-rate heuristic was worse than that with validation heuristic, even though the worse one outperformed the other training methods as shown in Figure 10. As the assumption of the noise-rate heuristic does not hold perfectly, a lower performance of the noise-rate heuristic looks reasonable.
However, we expect that the performance with this heuristic can be improved by stopping a little earlier than the estimated point, and we leave this extension as the future work.

B SUPPLEMENTARY EVALUATION

B.1 HYPERPARAMETER SELECTION (FIGURE 12)

Prestopping requires one additional hyperparameter, the history length $q$ in Definition 3.1. For hyperparameter tuning, we trained DenseNet ($L=40$, $k=12$) on CIFAR-10 and CIFAR-100 with a noise rate of 40%, and the history length $q$ was chosen in a grid $\{1, 5, 10, 15, 20\}$. Figure 12 shows the test error of Prestopping obtained by the grid search on two noisy CIFAR data sets. Regardless of the noise type, the lowest test error was typically achieved when the value of $q$ was 10 in both data sets. Therefore, the history length $q$ was set to be 10 in all experiments.

B.2 IMPACT OF NUMBER OF CLASSES ON Prestopping+ (FIGURE 13)

We investigate the impact of the number of classes on Prestopping+ under the pair noise of 40%. Figure 13 shows the ratio of refurbished samples used for training (hatched bar) with the refurbishing accuracy (solid line) on two data sets with 10 and 100 classes. When the number of classes is small (e.g., 10), Prestopping+ refurbished most of false-labeled samples very accurately; the refurbishing accuracy was consistently over 91.6% during the entire training period. On the other hand, when the number of classes is large (e.g., 100), the refurbishing accuracy drastically dropped from 92.0% to 76.7% as more samples were refurbished. Thus, as shown in Figure 13(b), collaboration with sample refurbishment is not always synergistic because such a significant drop in the refurbishing accuracy could induce many falsely corrected samples, especially when there are many classes.
C COMPLETE RESULTS ON BEST TEST ERROR

Table 1 shows the test error of seven training methods using two CNNs on two *simulated* noisy data sets with varying noise rates. Table 2 shows the test error of seven training methods using two CNNs on two *real-world* noisy data sets with different noise rates.

**Table 1:** The best test errors (%) of seven training methods on two types of *synthetic* noises with varying noise rates (0%, 10%, 20%, 30%, and 40%) in Figures 5 and 6.

<table>
<thead>
<tr>
<th>Noise Type</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Set</td>
<td>VGG</td>
<td>VGG</td>
<td>VGG</td>
<td>VGG</td>
</tr>
<tr>
<td>Default</td>
<td>7.20±0.05</td>
<td>11.4±0.14</td>
<td>27.3±0.35</td>
<td>38.9±0.07</td>
</tr>
<tr>
<td>Active Bias</td>
<td>7.45±0.36</td>
<td>11.4±0.08</td>
<td>26.7±0.48</td>
<td>39.9±0.09</td>
</tr>
<tr>
<td>Co-teaching</td>
<td>7.16±0.01</td>
<td>7.1±0.14</td>
<td>27.4±0.57</td>
<td>37.7±0.04</td>
</tr>
<tr>
<td>Co-teaching+</td>
<td>10.7±0.27</td>
<td>13.6±0.28</td>
<td>30.1±0.24</td>
<td>43.6±0.09</td>
</tr>
<tr>
<td>SELFIE</td>
<td>7.20±0.06</td>
<td>11.0±0.07</td>
<td>27.0±0.41</td>
<td>38.4±0.12</td>
</tr>
<tr>
<td>Prestopping</td>
<td>7.32±0.06</td>
<td>11.4±0.15</td>
<td>27.2±0.11</td>
<td>38.2±0.17</td>
</tr>
<tr>
<td>Prestopping+</td>
<td>8.02±0.11</td>
<td>10.9±0.10</td>
<td>26.7±0.18</td>
<td>37.7±0.16</td>
</tr>
</tbody>
</table>

(a) The best test errors under pair and symmetric noises of 0% (τ = 0).
Table 2: The best test errors (%) on real-world noises in Figure 7.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>ANIMAL-10N</th>
<th>FOOD-101N</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DenseNet</td>
<td>VGG</td>
</tr>
<tr>
<td>Default</td>
<td>18.2±0.15</td>
<td>20.8±0.36</td>
</tr>
<tr>
<td>Active Bias</td>
<td>17.6±0.13</td>
<td>20.0±0.30</td>
</tr>
<tr>
<td>Co-teaching</td>
<td>16.9±0.14</td>
<td>19.9±0.01</td>
</tr>
<tr>
<td>Co-teaching+</td>
<td>19.8±0.11</td>
<td>22.3±0.61</td>
</tr>
<tr>
<td>SELFIE</td>
<td>16.4±0.21</td>
<td>18.5±0.14</td>
</tr>
<tr>
<td>Prestopping</td>
<td>15.7±0.18</td>
<td>18.1±0.14</td>
</tr>
<tr>
<td>Prestopping+</td>
<td>15.5±0.12</td>
<td>17.8±0.07</td>
</tr>
</tbody>
</table>

D Case Study: Noisy Labels in Original CIFAR-100

One interesting observation is a noticeable improvement of Prestopping+ even when the noise rate was 0%, as shown in Table 4(a). It was turned out that Prestopping+ sometimes refurbished the labels of the false-labeled samples which were originally contained in the CIFAR data sets. Figure 14 shows a few successful refurbishment cases. For example, an image falsely annotated as a “Boy” was refurbished as a “Baby” (Figure 14(a)), and an image falsely annotated as a “Mouse” was refurbished as a “Hamster” (Figure 14(c)). Thus, this sophisticated label correction of Prestopping+ helps overcome the residual label noise in well-known benchmark data sets, which are misconceived to be clean, and ultimately further improves the generalization performance of a network.

Figure 14: Refurbishing of false-labeled samples originally contained in CIFAR-100. The subcaption represents “original label” → “refurbished label” recognized by Prestopping+.