Robust Inference via Generative Classifiers for Handling Noisy Labels

Kimin Lee [1] **Sukmin Yun** [1] **Kibok Lee** [2] **Honglak Lee** [3 2] **Bo Li** [4] **Jinwoo Shin** [1 5]

Abstract

Large-scale datasets may contain significant proportions of noisy (incorrect) class labels, and it is well-known that modern deep neural networks (DNNs) poorly generalize from such noisy training datasets. To mitigate the issue, we propose a novel inference method, termed *Robust Generative classifier (RoG)*, applicable to any discriminative (e.g., softmax) neural classifier pre-trained on noisy datasets. In particular, we induce a generative classifier on top of hidden feature spaces of the pre-trained DNNs, for obtaining a more robust decision boundary. By estimating the parameters of generative classifier using the minimum covariance determinant estimator, we significantly improve the classification accuracy with neither re-training of the deep model nor changing its architectures. With the assumption of Gaussian distribution for features, we prove that RoG generalizes better than baselines under noisy labels. Finally, we propose the ensemble version of RoG to improve its performance by investigating the layer-wise characteristics of DNNs. Our extensive experimental results demonstrate the superiority of RoG given different learning models optimized by several training techniques to handle diverse scenarios of noisy labels.

1. Introduction

Deep neural networks (DNNs) tend to generalize well when they are trained on large-scale datasets with ground-truth label annotations. For example, DNNs have achieved state-of-the-art performance on many classification tasks, e.g., image classification (He et al., 2016), object detection (Girshick, 2015), and speech recognition (Amodei et al., 2016). However, as the scale of training dataset increases, it becomes infeasible to obtain all ground-truth class labels from domain experts. A common practice is collecting the class labels from data mining on social media (Mahajan et al., 2018) or web data (Krause et al., 2016). Machine-generated labels are often used; e.g., the Open Images Dataset V4 contains such 70 million labels for training images (Kuznetsova et al., 2018). However, they may contain incorrect labels, and recent studies have shown that modern deep architectures may generalize poorly from the noisy datasets (Zhang et al., 2017) (e.g., see the black line of Figure 1(a)).

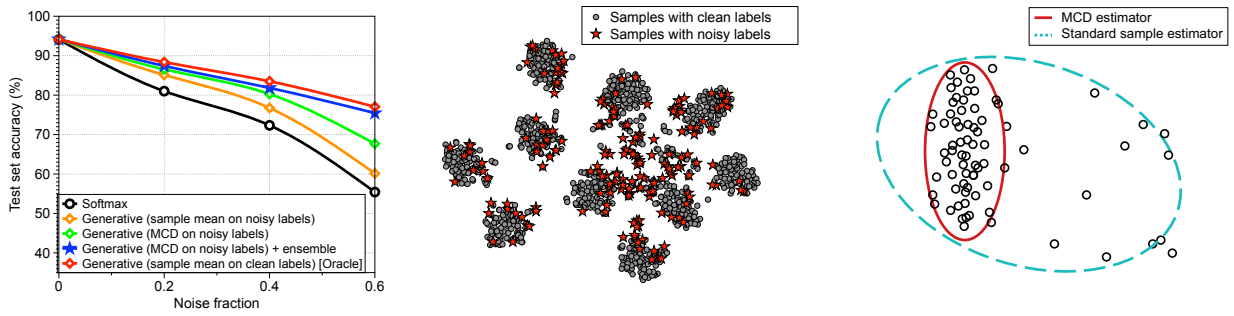
To address the poor generalization issue of DNNs with noisy labels, many training strategies have been investigated (Reed et al., 2014; Patrini et al., 2017; Ma et al., 2018; Han et al., 2018b; Hendrycks et al., 2018; Goldberger & Ben-Reuven, 2017; Jiang et al., 2018; Ren et al., 2018; Zhang & Sabuncu, 2018; Malach & Shalev-Shwartz, 2017; Han et al., 2018a). However, using such training methods may incur expensive back-and-forth costs (e.g., additional time and hyperparameter tuning) and suffer from the reproducibility issue. This motivates our approach of developing a more plausible inference method applicable to any pre-trained deep model. Hence, our direction is complementary to the prior works: one can combine ours and a prior training method for the best performance (see Tables 3, 4, & 5 in Section 4).

The key contribution of our work is to develop such an inference method, *Robust Generative classifier (RoG)*, which is applicable to any discriminative (e.g., softmax) neural classifier pre-trained on noisy datasets (without re-training). Our main idea is inducing a better posterior distribution from the pre-trained (noisy, though) feature representation by utilizing a robust generative classifier. Here, our belief is that the softmax DNNs can learn meaningful feature patterns shared by multiple training examples even under datasets with noisy labels, e.g., see (Arpit et al., 2017).

To motivate our approach, we first observe that training samples with noisy labels (red circles) are distributed like outliers when their hidden features are projected in a 2-dimensional space using t-SNE (Maaten & Hinton, 2008) (see Figure 1(b)). In other words, this phenomena implies that DNN representations even when trained with noisy labels may still exhibit *clustering properties* (i.e., the DNN learns embedding that tend to group clean examples of the same class into the clusters while pushing away the examples with corrupt labels outside these clusters). The observation inspires us to induce a generative classifier on the pre-trained hidden features since it can model joint data distributions $P(x, y)$ for input x and its label y for outlier

[1]KAIST [2]University of Michingan Ann Arbor [3]Google Brain [4]University of Illinois at Urbana Champaign [5]AItrics. Correspondence to: Kimin Lee <kiminlee@kaist.ac.kr>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.



(a) Test set accuracy comparison (b) Penultimate features by t-SNE (c) An illustration of the MCD estimator

Figure 1. Experimental results under DenseNet-100 and CIFAR-10 with uniform noise, i.e., the labels of a given proportion of training samples are flipped to other labels uniformly at random. (a) Test set accuracy of softmax and generative classifiers with various parameter estimations. (b) Visualization of features on the penultimate layer using t-SNE from training samples when the noise fraction is 20%. (c) An illustration of the MCD estimator: it is more robust against outliers by finding a subset with minimum covariance determinant.

detection and thus produce robust posterior $P(y \mid x)$ for prediction. Here, one may suggest training a deep generative classifier from scratch. However, such a fully generative approach is expensive and has been not popular for recent state-of-art classification. We instead post-process a light generative classifier only for inference.

In particular, we propose to induce the generative classifier under linear discriminant analysis (LDA) assumption and choose its parameters by the minimum covariance determinant (MCD) (Rousseeuw, 1984) estimator which calculates more robust parameters. We provide a theoretical support on the generalization property (Durrant & Kabán, 2010) of RoG based on MCD: it has the smaller errors on parameter estimations provably under some Gaussian assumptions. To improve RoG further, we observe that RoG built from low-level features can be often more effective since DNNs tend to have similar hidden features, regardless of whether they are trained with clean or noisy labels at early layers (Arpit et al., 2017; Morcos et al., 2018). Under the observations, we finally propose an ensemble version of RoG to incorporate all effects of low and high layers.

We demonstrate the effectiveness of RoG using modern neural architectures on image classification and natural language processing tasks. In all tested cases, our methods (e.g., see green and blue lines in Figure 1(a)) significantly outperform the softmax classifier, although they use the same feature representations trained by the noisy dataset. In particular, we show that RoG can be used to further improve various prior training methods (Reed et al., 2014; Patrini et al., 2017; Ma et al., 2018; Han et al., 2018b; Hendrycks et al., 2018) which are specialized to handle the noisy environment. For example, we improve the test accuracy of the state-of-the-art training method (Han et al., 2018b) on CIFAR-100 dataset with 45% noisy labels from 33.34% to 43.02%. Finally, RoG is shown to be working properly against more semantic noisy labels (generated from a machine labeler) and open-set noisy labels (Wang et al., 2018).

2. Related work

One of major directions for handling noisy labels is utilizing an estimated/corrected labels during training: Reed et al. (2014) proposed a bootstrapping method which trains deep models with new labels generated by a convex combination of the raw (noisy) labels and their predictions, and Ma et al. (2018) improved the bootstrapping method by utilizing the dimensionality of subspaces during training. Patrini et al. (2017) modified the loss and posterior distribution to eliminate the influence of noisy labels, and Hendrycks et al. (2018) improved such a loss correction method by utilizing the information from data with true class labels. Another promising direction has focused on training on selected (cleaner) samples: Jiang et al. (2018) introduced a meta-learning model, called MentorNet, and Han et al. (2018a) proposed a meta approach which can improve MentorNet. Ren et al. (2018) adaptively assigned weights to training samples based on their gradient directions. Malach & Shalev-Shwartz (2017) and Han et al. (2018b) proposed the selection methods based on an ensemble of deep models. A potential drawback of the above training methods is that they may incur expensive back-and-forth costs for training and hyperparameter tuning. On the other hand, our generative inference method is very cheap and can provide complementary benefits, i.e., ours can be easily applied to improve any of them.

Inducing a generative classifier (e.g., a mixture of Gaussian) on pre-trained deep models also has been investigated for various purposes: Hermansky et al. (2000) propose Tandem approaches which induce a generative model on top of hidden features for speech recognition. More recently, by inducing the generative model, Lee et al. (2018) introduce the Mahalanobis distance-based confidence score for novelty detection. However, their methods use naive parameter estimation under assuming perfect clean training labels, which should be highly influenced by outliers. We overcome the issue by using the MCD estimator.

3. Robust Inference via Generative Classifiers

In this section, we propose a novel inference method which obtains a robust posterior distribution from any softmax neural classifier pre-trained on datasets with noisy labels. Our idea is inducing the generative classifier given hidden features of the deep model. We show the robustness of our method in terms of high breakdown points (Hampel, 1971), and generalization error (Durrant & Kabán, 2010). We also investigate the layer-wise characteristics of generative classifiers, and introduce an ensemble of them to improve its performance.

3.1. Generative Classifier and MCD Estimator

Let $\mathbf{x}$ be an input and $y \in \{1, \cdots, C\}$ be its class label. Without loss of generality, suppose that a pre-trained softmax neural classifier is given: $P(y=c|\mathbf{x}) = \frac{\exp\left(\mathbf{w}_c^\top f(\mathbf{x}) + b_c\right)}{\sum_{c'} \exp\left(\mathbf{w}_{c'}^\top f(\mathbf{x}) + b_{c'}\right)}$, where $\mathbf{w}_c$ and b_c are the weight and the bias of the softmax classifier for class c, and $f(\cdot) \in \mathbb{R}^d$ denotes the output of the penultimate layer of DNNs. Then, without any modification on the pre-trained softmax neural classifier, we induce a generative classifier by assuming the class-conditional distribution follows the multivariate Gaussian distribution. In particular, we define C Gaussian distributions with a tied covariance $\mathbf{\Sigma}$, i.e., linear discriminant analysis (LDA) (Fisher, 1936), and a Bernoulli distribution for the class prior: $P(f(\mathbf{x})|y=c) = \mathcal{N}(f(\mathbf{x})|\mu_c, \mathbf{\Sigma}), P(y=c) = \beta_c$, where μ_c is the mean of multivariate Gaussian distribution and β_c is the normalized prior for class c. We provide an analytic justification on the LDA (i.e., tied covariance) assumption in the supplementary material. Then, based on the Bayesian rule, we induce a new posterior different from the softmax one as follows:

$$P(y=c|f(\mathbf{x})) = \frac{P(y=c)\,P(f(\mathbf{x})|y=c)}{\sum_{c'} P(y=c')\,P(f(\mathbf{x})|y=c')}$$
$$= \frac{\exp\left(\mu_c^\top \mathbf{\Sigma}^{-1} f(\mathbf{x}) - \frac{1}{2}\mu_c^\top \mathbf{\Sigma}^{-1}\mu_c + \log \beta_c\right)}{\sum_{c'} \exp\left(\mu_{c'}^\top \mathbf{\Sigma}^{-1} f(\mathbf{x}) - \frac{1}{2}\mu_{c'}^\top \mathbf{\Sigma}^{-1}\mu_{c'} + \log \beta_{c'}\right)}.$$

To estimate the parameters of the generative classifier, one can compute the sample class mean and covariance of training samples $\mathcal{X}_N = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$:

$$\bar{\mu}_c = \sum_{i:y_i=c} \frac{f(\mathbf{x}_i)}{N_c}, \quad \bar{\beta}_c = \frac{N_c}{N},$$
$$\bar{\mathbf{\Sigma}} = \sum_c \sum_{i:y_i=c} \frac{(f(\mathbf{x}_i) - \bar{\mu}_c)(f(\mathbf{x}_i) - \bar{\mu}_c)^\top}{N}, \quad (1)$$

where N_c is the number of samples labeled to be class c.

However, one can expect that the naive sample estimator (1) can be highly influenced by outliers (i.e., training samples with noisy labels). In order to improve the robustness, we propose the so-called *Robust Generative classifier (RoG)*, which utilizes the minimum covariance determinant (MCD) estimator (Rousseeuw & Driessen, 1999) to estimate its parameters. For each class c, the main idea of MCD is finding a subset $\mathcal{X}_{K_c}$ for which the determinant of the corresponding sample covariance is minimized:

$$\min_{\mathcal{X}_{K_c} \subset \mathcal{X}_{N_c}} \det\left(\widehat{\mathbf{\Sigma}}_c\right) \quad \text{subject to } |\mathcal{X}_{K_c}| = K_c, \quad (2)$$

where $\mathcal{X}_{N_c}$ is the set of training samples labeled to be class c, $\widehat{\mathbf{\Sigma}}_c$ is the sample covariance of $\mathcal{X}_{K_c}$ and $0 < K_c < N_c$ is a hyperparameter. Then, only using the samples in $\bigcup_c \mathcal{X}_{K_c}$, it estimates the parameters, i.e., $\widehat{\mu}_c, \widehat{\mathbf{\Sigma}}, \widehat{\beta}_c$, of the generative classifier, by following (1). Such a new estimator can be more robust by removing the outliers which might be widely scattered in datasets (see Figure 1(c)).

The robustness of MCD estimator has been justified in the literature: it is known to have near-optimal breakdown points (Hampel, 1971), i.e., the smallest fraction of data points that need to be replaced by arbitrary values (i.e., outliers) to fool the estimator completely. Formally, denote $\mathcal{Y}_M$ as a set obtained by replacing M data points of set $\mathcal{Y}$ by some arbitrary values. Then, for a multivariate mean estimator $\mu = \mu(\mathcal{Y})$ from $\mathcal{Y}$, the breakdown point is defined as follows (see the supplementary material for more detailed explanations including the breakdown point of covariance estimator):

$$\varepsilon^*(\mu, \mathcal{Y}) = \frac{1}{|\mathcal{Y}|} \min\left\{M \in [|\mathcal{Y}|] : \sup_{\mathcal{Y}_M} \|\mu(\mathcal{Y}) - \mu(\mathcal{Y}_M)\| = \infty\right\},$$

where the set $\{1, \ldots, n\}$ is denoted by $[n]$ for positive integer n. While the breakdown point of the naive sample estimator is 0%, the MCD estimator for the generative classifier under LDA assumption is known to attain near optimal breakdown value of $\min_c \frac{\lfloor (N_c - d + 1)/2 \rfloor}{N_c} \approx 50\%$ (Lopuhaa et al., 1991). Inspired by this fact, we choose the default value of K_c in (2) by $\lfloor (N_c + d + 1)/2 \rfloor$.

We also establish the following theoretical support that the MCD-based generative classifier (i.e., RoG) can have smaller errors on parameter estimations, compared to the naive sample estimator, under some assumptions for its analytic tractability.

Theorem 1 *Assume the followings:*

(A1) For (clean) sample $\mathbf{x}$ of correct label, the class-conditional distribution of hidden feature $f(\mathbf{x})$ of DNNs has mean μ_c and tied covariance matrix $\sigma^2\mathbf{I}$. For (outlier) sample $\mathbf{x}$ of incorrect label, the distribution of hidden feature has mean $\mu_{\texttt{out}}$ and covariance matrix $\sigma^2_{\texttt{out}}\mathbf{I}$, where $\mathbf{I} \in \mathbb{R}^{d \times d}$ is the identity matrix.

($\mathcal{A}2$) *All classes have the same number of samples (i.e., $N_c = \frac{N}{C}$), the same fraction $\delta_{\texttt{out}} < 1$ of outliers, and the sample fraction $\delta_{\texttt{mcd}} = \frac{K_c}{N_c} < 1$ of samples selected by MCD estimator.*

($\mathcal{A}3$) *The outliers are widely scattered such that $\sigma^2 < \sigma^2_{\texttt{out}}$.*

($\mathcal{A}4$) *The number of outliers is not too large such that $\delta_{\texttt{out}} < 1 - \delta_{\texttt{mcd}}$ and $\delta_{\texttt{mcd}} > \frac{d}{N_c}$.*

Let $\widehat{\mu}, \widehat{\mathbf{\Sigma}}$ and $\bar{\mu}, \bar{\mathbf{\Sigma}}$ be the outputs of the MCD and sample estimators, respectively. Then, for all $c,\ c'$, $\widehat{\mu}, \bar{\mu}, \widehat{\mathbf{\Sigma}}, \bar{\mathbf{\Sigma}}$ converge almost surely to their expectation as $N \to \infty$, and it holds that

$$\|\mu_c - \widehat{\mu}_c\|_1 \xrightarrow{\text{a.s.}} \lim_{N\to\infty} \|\mu_c - \widehat{\mu}_c\|_1 = 0,$$

$$\|\mu_c - \bar{\mu}_c\|_1 \xrightarrow{\text{a.s.}} \lim_{N\to\infty} \|\mu_c - \bar{\mu}_c\|_1 = \delta_{\texttt{out}}\|\mu_c\|_1, \tag{3}$$

$$\frac{\phi(\widehat{\mathbf{\Sigma}})\|\widehat{\mu}_c - \widehat{\mu}_{c'}\|_2}{\phi(\bar{\mathbf{\Sigma}})\|\bar{\mu}_c - \bar{\mu}_{c'}\|_2} \xrightarrow{\text{a.s.}} \lim_{N\to\infty} \frac{\phi(\widehat{\mathbf{\Sigma}})\|\widehat{\mu}_c - \widehat{\mu}_{c'}\|_2}{\phi(\bar{\mathbf{\Sigma}})\|\bar{\mu}_c - \bar{\mu}_{c'}\|_2} = \lim_{N\to\infty} \frac{1}{\left(1-\delta_{\texttt{out}}\right)^2 \phi(\bar{\mathbf{\Sigma}})} \geq 1, \tag{4}$$

where $\phi(\widehat{\mathbf{\Sigma}}) = 4\|\widehat{\mathbf{\Sigma}}^{-1}\|_2\|\widehat{\mathbf{\Sigma}}\|_2\left(1 + \|\widehat{\mathbf{\Sigma}}^{-1}\|_2\|\widehat{\mathbf{\Sigma}}\|_2\right)^{-2}$.

The proof of the above theorem is given in the supplementary material, where it is built upon the fact that the determinants can be expressed as the d-th degree polynomial of outlier ratio under the assumptions. We note that one might enforce the assumptions of the diagonal covariance matrices in $\mathcal{A}1$ to hold under an affine translation of hidden features. In addition, the assumption in $\mathcal{A}4$ holds when N_c is large enough. Nevertheless, we think most assumptions of Theorem 1 are not necessary to claim the superiority of RoG and it is an interesting future direction to explore to relax them.

The generalization error bound of a generative classifier under the assumption that the class-conditional Gaussian distributions of features is known to be bounded as follows (Durrant & Kabán, 2010):

$$P_{f(\mathbf{x})}\left(y^* \neq \arg\max_y P_{\widehat{\mu}_c,\widehat{\Sigma}}(y|f(\mathbf{x}))\right) \leq \sum_c \sum_{c'\neq c} \exp\left(-\frac{\|\widehat{\mu}_c - \widehat{\mu}_{c'}\|_2}{8\sigma^2}\cdot\phi(\widehat{\mathbf{\Sigma}})\right) + D\|\mu_c - \widehat{\mu}_c\|_1,$$

for some constant $D > 0$. Therefore, (3) and (4) together imply that utilizing the MCD estimator provides a better generalization bound, compared to the sample estimator.

3.2. Approximation Algorithm for MCD

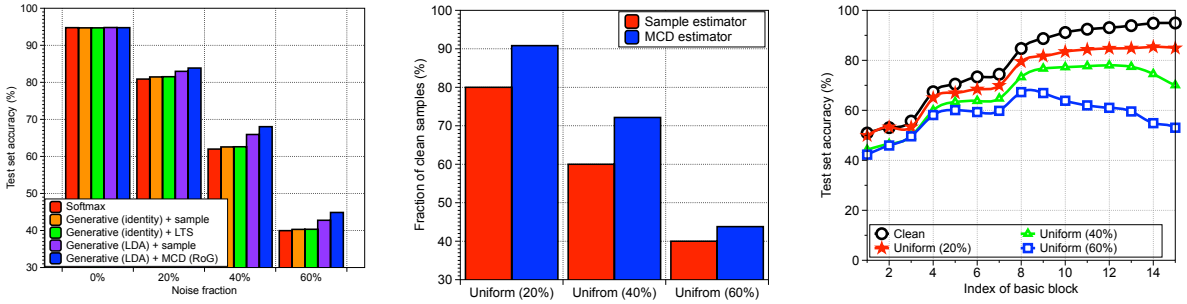
Even though the MCD estimator has several advantages, the optimization (2) is computationally intractable (i.e., NP-hard) to solve (Bernholt, 2006). To handle this issue, we aim to compute its approximate solution, by following the idea of Hubert & Van Driessen (2004). We design two step scheme as follows: (a) obtain the mean and covariance, i.e., $\widehat{\mu}_c, \widehat{\mathbf{\Sigma}}_c$, using Algorithm 1 for each class c, and (b) compute the tied covariance by $\widehat{\mathbf{\Sigma}} = \frac{\sum_c K_c \widehat{\mathbf{\Sigma}}_c}{\sum_c K_c}$. In other words, we apply the MCD estimator for each class, and combine the individual covariances into a single one due to the tied covariance assumption of LDA. Even though finding the optimal solution of the MCD estimator under a single Gaussian distribution is still intractable, Algorithm 1 can produce a local optimal solution since it monotonically decreases the determinant under any random initial subset (Rousseeuw & Driessen, 1999). We choose $I_{\max} = 2$ in our experiments since additional iterations would not improve the results significantly.

Algorithm 1 (Rousseeuw & Driessen, 1999) Approximating MCD for a single Gaussian.

Input: $\mathcal{X}_{N_c} = \{\mathbf{x}_i : i = 1, \cdots, N_c\}$ and the maximum number of iterations $I_{\max}$.
Uniformly sample initial subset $\mathcal{X}_{K_c} \subset \mathcal{X}_{N_c}$, where $|\mathcal{X}_{K_c}| = \lfloor (N_c + d + 1)/2 \rfloor$.
Compute a mean $\widehat{\mu}_c = \frac{1}{|\mathcal{X}_{K_c}|}\sum_{\mathbf{x}\in\mathcal{X}_{K_c}} f(\mathbf{x})$, and covariance $\widehat{\mathbf{\Sigma}}_c = \frac{1}{|\mathcal{X}_{K_c}|}\sum_{\mathbf{x}\in\mathcal{X}_{K_c}} (f(\mathbf{x}) - \widehat{\mu}_c)(f(\mathbf{x}) - \widehat{\mu}_c)^\top$.
for $i = 1$ **to** $I_{\max}$ **do**
  Compute the Mahalanobis distance for all $\mathbf{x} \in \mathcal{X}_{N_c}$: $\alpha(\mathbf{x}) = (f(\mathbf{x}) - \widehat{\mu}_{\mathbf{c}})^\top \widehat{\mathbf{\Sigma}}_c^{-1} (f(\mathbf{x}) - \widehat{\mu}_{\mathbf{c}})$.
  Update $\mathcal{X}_{K_c}$ such that it includes $\lfloor (N_c + d + 1)/2 \rfloor$ samples with the smallest distance $\alpha(\mathbf{x})$.
  Compute sample mean and covariance, i.e., $\widehat{\mu}_c, \widehat{\mathbf{\Sigma}}_c$, using new subset $\mathcal{X}_{K_c}$.
  Exit the loop if the determinant of covariance matrix is not decreasing anymore.
end for
Return $\widehat{\mu}_c$ and $\widehat{\mathbf{\Sigma}}_c$

3.3. Ensemble of Generative Classifiers

To further improve the performance of our method, we consider the ensemble of generative classifiers not only from the penultimate features but also from other low-level features in DNNs. Formally, given training data, we extract ℓ-th hidden features of DNNs, denoted by $f_\ell(\mathbf{x}) \in \mathbb{R}^{d_\ell}$, and compute the corresponding parameters of a generative classifier (i.e., $\widehat{\mu}_{\ell,c}$ and $\widehat{\mathbf{\Sigma}}_\ell$) using the (approximated version of) MCD estimator. Then, the final posterior distribution is obtained by the weighted sum of all posterior distributions of generative classifiers: $\sum_\ell \alpha_\ell P\left(y = c|f_\ell(\mathbf{x})\right)$, where α_ℓ is an ensemble weight at ℓ-th layer. In our experiments, we choose the weight of each layer by optimizing negative



(a) Model comparison (b) Accuracy of the MCD estimator (c) Layer-wise accuracy

Figure 2. Experimental results under ResNet-34 model and CIFAR-10 dataset. (a) Test accuracy of generative classifiers from penultimate features under various assumptions: identity covariance and tied covariance (LDA). (b) The number of clean samples among selected samples by the MCD estimator. (c) Test accuracy of generative classifiers computed at different basic blocks.

Model	Inference method	Ensemble	Clean	Uniform (20%)	Uniform (40%)	Uniform (60%)
DenseNet	Softmax	-	94.11	81.01	72.34	55.42
	Generative + sample	-	94.18	85.12	76.75	60.14
		✓	93.97	87.40	81.27	69.81
	Generative + MCD (ours)	-	94.22	86.54	80.27	67.67
		✓	94.18	**87.41**	**81.83**	**75.45**
ResNet	Softmax	-	94.76	80.88	61.98	39.96
	Generative + sample	-	94.80	82.97	65.92	42.76
		✓	94.82	83.36	68.57	46.45
	Generative + MCD (ours)	-	94.76	83.86	68.03	44.87
		✓	94.68	**84.62**	**75.28**	**54.57**

Table 1. Effects of an ensemble method. We use the CIFAR-10 dataset with various uniform noise fractions. All values are percentages and the best results are highlighted in bold if the gain is bigger than 1% compared to softmax classifier.

log-likelihood (NLL) loss over the validation set. One can expect that this natural scheme can bring an extra gain in improving the performance due to ensemble effects.

4. Experiments

In this section, we demonstrate the effectiveness of the proposed method using deep neural networks on various vision and natural language processing tasks. We provide more detailed experimental setups in the supplementary material. Code is available at `github.com/pokaxpoka/RoGNoisyLabel`.

4.1. Experimental Setup

For evaluation, we apply the proposed method to deep neural networks including DenseNet (Huang & Liu, 2017) and ResNet (He et al., 2016) for the classification tasks on CIFAR (Krizhevsky & Hinton, 2009), SVHN (Netzer et al., 2011), Twitter Part of Speech (Gimpel et al., 2011), and Reuters (Lewis et al., 2004) datasets with noisy labels. Following the setups of (Ma et al., 2018; Han et al., 2018b), we first consider two types of random noisy labels: corrupting a label to other class uniformly at random (uniform) and corrupting a label only to a specific class (flip). Our method is also evaluated on semantic noisy labels from a machine classifier and open-set noisy labels (Wang et al., 2018).

For ensembles of generative classifiers, we induce the generative classifiers from basic blocks of the last dense (or residual) block of DenseNet (or ResNet), where ensemble weights of each layer are tuned on an additional validation set, which consists of 1000 images with noisy labels. Here, when learning the weights, we use only 500 samples out of 1000, chosen by the MCD estimator to remove the outliers (see the supplementary material for more details). The size of feature maps on each convolutional layers is reduced by average pooling for computational efficiency: $\mathcal{F} \times \mathcal{H} \times \mathcal{W} \rightarrow \mathcal{F} \times 1$, where $\mathcal{F}$ is the number of channels and $\mathcal{H} \times \mathcal{W}$ is the spatial dimension.

4.2. Ablation Study

We first evaluate the performance of generative classifiers with various assumptions: identity covariance (Euclidean) and tied covariance (LDA). In the case of identity covariance, we also apply a robust estimator called the least trimmed square (LTS) estimator (Rousseeuw, 1984) which finds a K-subset with the smallest error and computes the sample mean from it, i.e., $\min_{\widehat{\mu}} \sum_{i=1}^{K} (\|\mathbf{x}_i - \widehat{\mu}\|_2^2)$. Figure 2(a) reports the test set accuracy of the softmax and

Noise type (%)	ResNet			DenseNet		
	CIFAR-10	CIFAR-100	SVHN	CIFAR-10	CIFAR-100	SVHN
	Softmax / RoG			Softmax / RoG		
Clean	94.76 / 94.68	76.81 / 76.97	95.96 / 96.09	94.11 / 94.18	**75.69** / 72.67	96.59 / 96.18
Uniform (20%)	80.88 / **84.62**	64.43 / **68.29**	83.52 / **91.67**	81.01 / **87.41**	61.72 / **64.29**	86.92 / **89.50**
Uniform (40%)	61.98 / **75.28**	48.62 / **60.76**	72.89 / **87.16**	72.34 / **81.83**	50.89 / **55.68**	81.91 / **85.71**
Uniform (60%)	39.96 / **54.57**	27.57 / **48.42**	61.23 / **80.52**	55.42 / **75.45**	38.33 / **44.12**	71.18 / **77.67**
Flip (20%)	79.65 / **88.73**	65.14 / **73.37**	85.49 / **93.00**	79.18 / **91.23**	65.37 / **69.03**	95.04 / 94.86
Flip (40%)	58.13 / **61.56**	46.61 / **66.71**	65.88 / **87.96**	56.29 / **86.42**	46.04 / **69.38**	88.83 / **93.57**

Table 2. Test accuracy (%) of different models trained on various datasets. We use the ensemble version of RoG, and the best results are highlighted in bold if the gain is bigger than 1%.

Dataset	Training method	Clean	Uniform (20%)	Uniform (40%)	Uniform (60%)
		Softmax / RoG			
CIFAR-10	Cross-entropy	94.34 / 94.20	81.95 / **84.63**	63.84 / **74.72**	62.45 / **67.47**
	Bootstrap (hard)	94.56 / 94.52	82.90 / **86.27**	75.97 / **80.72**	72.91 / **75.41**
	Bootstrap (soft)	94.46 / 94.28	80.29 / **84.82**	65.22 / **74.22**	58.55 / **66.68**
	Forward	94.53 / 94.52	85.80 / **86.84**	77.95 / **79.87**	72.56 / **74.75**
	Backward	94.39 / 94.44	77.44 / **79.16**	62.83 / **68.29**	56.64 / **66.44**
	D2L	94.55 / 94.29	88.89 / 89.00	86.68 / 87.00	76.83 / **77.92**
CIFAR-100	Cross-entropy	76.31 / 75.40	61.11 / **64.82**	45.08 / **55.90**	34.97 / **41.25**
	Bootstrap (hard)	75.65 / 75.49	61.61 / **64.81**	51.27 / **57.22**	39.04 / **43.69**
	Bootstrap (soft)	76.40 / 76.02	60.28 / **64.04**	47.66 / **56.51**	34.68 / **42.47**
	Forward	75.84 / 75.93	63.73 / **66.02**	53.03 / **57.69**	41.28 / **45.28**
	Backward	76.75 / 76.28	56.24 / **62.13**	37.70 / **50.23**	23.55 / **37.18**
	D2L	76.13 / 75.93	71.90 / 72.09	63.61 / **64.85**	9.51 / **40.57**
SVHN	Cross-entropy	96.38 / 96.41	83.45 / **91.14**	60.86 / **80.36**	38.29 / **54.99**
	Bootstrap (hard)	96.40 / 96.12	83.43 / **91.98**	74.25 / **86.83**	66.51 / **77.14**
	Bootstrap (soft)	96.51 / 96.10	86.43 / **90.84**	58.22 / **79.90**	44.52 / **62.52**
	Forward	96.36 / 96.00	88.21 / **91.99**	80.35 / **86.49**	82.16 / **84.99**
	Backward	96.43 / 96.09	87.00 / 87.11	72.02 / **73.32**	50.54 / **64.01**
	D2L	96.49 / 96.37	92.31 / **93.58**	94.46 / 94.68	92.87 / 93.25

Table 3. Test accuracy (%) of ResNet trained on various training methods which utilize a single classifier. We use the ensemble version of RoG, and the best results are highlighted in bold if the gain is bigger than 1%.

generative classifiers on features extracted from the penultimate layer using ResNet-34 trained on the CIFAR-10 dataset with the uniform noise type. First, one can observe that the generative classifiers with LDA assumption (blue and purple bars) generalize better than the softmax (red bar) and generative classifiers with identity covariance (orange and green bars) well from noisy labels. Here, we remark that they still provide a comparable classification accuracy of softmax classifier when the model is trained on clean dataset (i.e., noise = 0%). On top of that, by utilizing the MCD estimator, the classification accuracy (blue bar) is further improved compared to that employs only the naive sample estimator (purple bar). This is because the MCD estimator indeed selects the training samples with clean labels as shown in Figure 2(b). The above results justify the proposed generative classifier, in comparison with other alternatives.

Next, to confirm that the ensemble approach is indeed effective, we measure a classification accuracy of generative classifier from different basic blocks of ResNet-34. First, we found that the performances of the generative classifiers from low-level features are more stable, while the accuracy of generative classifier from penultimate layer significantly decreases as the noisy fraction increases as shown in Figure 2(c). We expect that this is because the dimension (i.e., number of channels) of low-level features is usually smaller than that of high-level features. Since the breakdown point of MCD is inversely proportional to the feature dimension, the generative classifiers from low-level features can be more robust. This also coincides with the prior observation in the literature (Morcos et al., 2018) that DNNs tend to have similar hidden features at early layers, regardless of whether they train clean or noisy labels. Since the generative classifiers from low-level features are more stable, the ensemble method significantly improves the classification accuracy as shown in Table 1. Finally, Table 2 reports the classification accuracy for all networks and datasets, where the proposed method significantly outperforms the softmax

Dataset	Noise type (%)	Cross-entropy	Decoupling	MentorNet	Co-teaching	Co-teaching + RoG
CIFAR-10	Flip (45%)	49.50	48.80	58.14	71.17	71.26
	Uniform (50%)	48.87	51.49	71.10	74.12	**76.67**
	Uniform (20%)	76.25	80.44	80.76	82.13	**84.32**
CIFAR-100	Flip (45%)	31.99	26.05	31.60	33.34	**43.18**
	Uniform (50%)	25.21	25.80	39.00	41.49	**45.42**
	Uniform (20%)	47.55	44.52	52.13	54.27	**58.16**

Table 4. Test accuracy (%) of 9-layer CNNs trained on various training methods which utilize an ensemble of classifiers or meta-learning model. We use the ensemble version of RoG and the best results are highlighted in bold if the gain is bigger than 1%.

Dataset	Training method	Softmax / RoG			
		Clean	Uniform (20%)	Uniform (40%)	Uniform (60%)
Twitter	Cross-entropy	**87.47** / 85.28	79.13 / **81.66**	66.74 / **79.37**	50.83 / **73.65**
	Forward (gold)	78.07 / **83.59**	72.97 / **81.60**	64.55 / **78.24**	51.59 / **72.33**
	GLC	83.47 / **84.68**	66.09 / **81.66**	59.72 / **79.00**	53.14 / **72.93**
Reuters	Cross-entropy	**95.88** / 94.77	87.74 / **92.83**	76.54 / **82.20**	57.49 / **64.98**
	Forward (gold)	94.57 / 94.75	88.44 / **93.24**	77.85 / **82.56**	61.01 / **66.56**
	GLC	**95.97** / 94.91	81.45 / **92.75**	73.40 / **83.82**	59.21 / **67.91**

Table 5. Test accuracy (%) of 2-layer FCNs trained on NLP datasets with uniform noise. We use the ensemble version of RoG, and the best results highlighted in bold if the gain is bigger than 1%.

classifier for all tested cases.

4.3. Compatibility and Comparison with the State-of-Art Training Methods

We compare the performance of the standard softmax classifier and RoG when they are combined with other various training methods for noisy environments, where more detailed explanations about training methods are given in the supplementary material. First, we consider the following methods that require to train only a single network: Hard/soft bootstrapping (Reed et al., 2014), forward/backward (Patrini et al., 2017), and D2L (Ma et al., 2018). Following the same experimental setup in Ma et al. (2018)[1], we use ResNet-44 and only consider the uniform noises of various levels. Table 3 shows the classification accuracy of softmax classifier and the ensemble version of RoG. Note that RoG always improves the classification accuracy compared to the softmax classifier, where the gains due to ours are more significant than those due to other special training methods.

We also consider the following methods that require to train multiple networks, i.e., an ensemble of classifiers or a meta-learning model: Decoupling (Malach & Shalev-Shwartz, 2017), MentorNet (Jiang et al., 2018) and Co-teaching (Han et al., 2018b). Following the same experimental setup of Han et al. (2018b)[2], we use a 9-layer convolutional neural network (CNN), and consider the CIFAR-10 and CIFAR-100 datasets with uniform and flip noise. In this setup, we only apply RoG to a model pre-trained by Co-teaching since it outperforms other training methods. As shown in Table 4, RoG with Co-teaching method achieves the best performance in all tested cases.

We further apply our inference method to non-convolutional neural networks on natural language processing (NLP) tasks: the text categorization on Reuters (Lewis et al., 2004), and part-of-speech (POS) tagging on Twitter POS (Gimpel et al., 2011). By following the same experimental setup of Hendrycks et al. (2018)[3], we train 2-layer fully connected networks (FCNs) using forward (gold)[4], and GLC (Hendrycks et al., 2018) methods. Note that they are designed to train a single network using a set of trusted data with golden clean labels (1% of training samples in our experiments). Hence, the setting is slightly different from what we considered so far, but we run RoG (without utilizing 1% knowledge of ground truth) to compare. Table 5 shows that, even with this unfair disadvantage, RoG can improve the performance over the baselines for these NLP datasets with noisy labels.

[1]The code is available at https://github.com/xingjunm/dimensionality-driven-learning.

[2]We used a reference implementation: https://github.com/bhanML/Co-teaching.

[3]The code is available at https://github.com/mmazeika/glc.

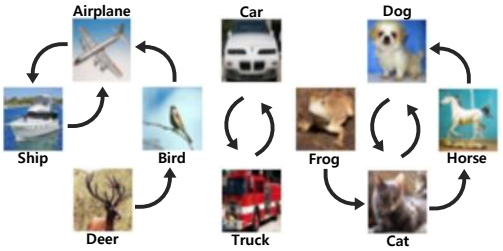
[4]We use an augmented version of forward (Patrini et al., 2017) which estimates a corruption matrix using the trusted data.

Training method	Label generator (noisy fraction) on CIFAR-10			Label generator (noisy fraction) on CIFAR-100		
	DenseNet (32%)	ResNet (38%)	VGG (34%)	DenseNet (34%)	ResNet (37%)	VGG (37%)
	Softmax / RoG			Softmax / RoG		
Cross-entropy	67.24 / **68.33**	62.26 / **64.15**	68.77 / **70.04**	50.72 / **61.14**	50.68 / **53.09**	51.08 / **53.64**
Bootstrap (hard)	67.31 / **68.40**	62.22 / **63.98**	69.11 / **70.09**	51.31 / **53.66**	50.62 / **52.62**	50.91 / **53.46**
Bootstrap (soft)	67.17 / **68.38**	62.15 / **64.03**	69.28 / **70.11**	50.57 / **54.71**	50.68 / **53.30**	51.41 / **53.76**
Forward	67.46 / **68.20**	61.96 / **64.24**	68.95 / **70.09**	50.59 / **53.91**	51.04 / **53.36**	51.05 / **53.63**
Backward	67.31 / **68.66**	62.40 / **63.45**	69.04 / **70.18**	50.54 / **54.01**	50.30 / **53.03**	51.15 / **53.50**
D2L	66.91 / **68.57**	59.10 / **60.25**	57.97 / **59.94**	5.00 / **31.67**	23.71 / **39.92**	40.97 / **45.42**

Table 6. Test accuracy (%) of DenseNet on the CIFAR-10 and CIFAR-100 datasets with semantic noisy labels. We use the ensemble version of RoG, and the best results are highlighted in bold if the gain is bigger than 1%.

Open-set data	Softmax	RoG
CIFAR-100	79.01	**83.37**
ImageNet	86.88	87.05
CIFAR-100 + ImageNet	81.58	**84.35**

Table 7. Test accuracy (%) of DenseNet on the CIFAR-10 dataset with open-set noisy labels. We use the ensemble of RoG. The best results are highlighted in bold if the gain is bigger than 1%.



(a) Confusion graph on CIFAR-10

(b) Open-set noise example

Figure 3. (a) Confusion graph from ResNet-34 trained on the CIFAR-10 dataset. (b) Examples of training samples with open-set noise for "bird" in the CIFAR-10 dataset.

4.4. Semantic and Open-Set Noisy Labels

In this section, our method is evaluated under more realistic noisy labels. First, in order to generate more semantically meaningful noisy labels, we train DenseNet-100, ResNet-34 and VGG-13 (Simonyan & Zisserman, 2015) using 5% and 20% of CIFAR-10 and CIFAR-100 training samples with clean labels, respectively. Then, we produce the labels of remaining training samples based on their predictions, and train another DenseNet-100 with the pseudo-labeled samples. Figure 3(a) shows a confusion graph for pseudo-labels, obtained from ResNet-34 trained on the 5% of CIFAR-10: each node corresponds to a class, and an edge from the node represents its most confusing class. Note that the weak classification system produces semantically noisy labels; e.g., "Cat" is confused with "Dog", but not with "Car". We remark that DenseNet and VGG also produce similar confusion graphs. Table 6 shows RoG consistently improves the performance, while the gains due to other special training methods are not very significant.

Our final benchmark is the open-set noisy scenario (Wang et al., 2018). In this case, some training images are often from the open world and not relevant to the targeted classification task at all, i.e., out-of-distribution. However, they are still labeled to certain classes. For example, as shown in Figure 3(b), noisy samples like "chair" from CIFAR-100 and "door" from (downsampled) ImageNet (Chrabaszcz et al., 2017) can be labeled as "bird" to train, even though their true labels are not contained within the set of training classes in the CIFAR-10 dataset. In our experiments, open-set noisy datasets are built by replacing some training samples in CIFAR-10 by out-of-distribution samples, while keeping the labels and the number of images per class unchanged. We train DenseNet-100 on CIFAR-10 with 60% open-set noise. As shown in Table 7, our method achieves comparable or significantly better test accuracy than the softmax classifier.

5. Conclusion

We propose a new inference method for handling noisy labels. Our main idea is inducing the generative classifier on top of fixed features from the pre-trained model. Such "deep generative classifiers" have been largely dismissed for fully-supervised classification settings as they are often substantially outperformed by discriminative deep classifiers (e.g., softmax classifiers). In contrast to this common belief, we show that it is possible to formulate a simple generative classifier that is significantly more robust to labeling noise without much sacrifice of the discriminative performance for clean labeling setting. We expect that our work would bring a refreshing perspective for other related tasks, e.g., memorization (Zhang et al., 2017), adversarial attacks (Szegedy et al., 2014), and semi-supervised learning (Oliver et al., 2018).

Acknowledgements

This work was supported by Institute for Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No.2017-0-01779, A machine learning and statistical inference framework for explainable artificial intelligence), NSF CAREER IIS-1453651, Sloan Research Fellowship, and Kwanjeong Educational Foundation Scholarship. This work is partially supported by DARPA under Grant 00009970, Wechat. We also thank Dawn Song, Dan Hendrycks, Sungsoo Ahn and Insu Han for helpful discussions.

References

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *ICML*, 2016.

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 *ICML*, 2017.

Bernholt, T. Robust estimators are hard to compute. Technical report, University of Dortmund, 2006.

Chrabaszcz, P., Loshchilov, I., and Hutter, F. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint*, 2017.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Durrant, R. J. and Kabán, A. Compressed fisher linear discriminant analysis: Classification of randomly projected data. In *ACM SIGKDD*, 2010.

Fisher, R. A. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

Garcia-Escudero, L. A. and Gordaliza, A. Robustness properties of k means and trimmed k means. *Journal of the American Statistical Association*, 94(447):956–969, 1999.

Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., Heilman, M., Yogatama, D., Flanigan, J., and Smith, N. A. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *ACL*, 2011.

Girshick, R. Fast r-cnn. In *ICCV*, 2015.

Goldberger, J. and Ben-Reuven, E. Training deep neural-networks using a noise adaptation layer. In *ICLR*, 2017.

Hampel, F. R. A general qualitative definition of robustness. *The Annals of Mathematical Statistics*, pp. 1887–1896, 1971.

Han, B., Niu, G., Yao, J., Yu, X., Xu, M., Tsang, I., and Sugiyama, M. Pumpout: A meta approach for robustly training deep neural networks with noisy labels. *arXiv preprint*, 2018a.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: robust training deep neural networks with extremely noisy labels. In *NeurIPS*, 2018b.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. Using trusted data to train deep networks on labels corrupted by severe noise. In *NeurIPS*, 2018.

Hermansky, H., Ellis, D. P., and Sharma, S. Tandem connectionist feature extraction for conventional hmm systems. In *ICASSP*, 2000.

Huang, G. and Liu, Z. Densely connected convolutional networks. In *CVPR*, 2017.

Hubert, M. and Van Driessen, K. Fast and robust discriminant analysis. *Computational Statistics & Data Analysis*, 45(2):301–320, 2004.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. Mentornet: Regularizing very deep neural networks on corrupted labels. In *ICML*, 2018.

Krause, J., Sapp, B., Howard, A., Zhou, H., Toshev, A., Duerig, T., Philbin, J., and Fei-Fei, L. The unreasonable effectiveness of noisy data for fine-grained recognition. In *ECCV*, 2016.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T., et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint*, 2018.

Lasserre, J. A., Bishop, C. M., and Minka, T. P. Principled hybrids of generative and discriminative models. In *CVPR*, 2006.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018.

Lewis, D. D., Yang, Y., Rose, T. G., and Li, F. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

Lopuhaa, H. P., Rousseeuw, P. J., et al. Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *The Annals of Statistics*, 19(1): 229–248, 1991.

Ma, X., Wang, Y., Houle, M. E., Zhou, S., Erfani, S. M., Xia, S.-T., Wijewickrema, S., and Bailey, J. Dimensionality-driven learning with noisy labels. In *ICML*, 2018.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov): 2579–2605, 2008.

Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and van der Maaten, L. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.

Malach, E. and Shalev-Shwartz, S. Decoupling "when to update" from "how to update". In *NeurIPS*, 2017.

Morcos, A. S., Raghu, M., and Bengio, S. Insights on representational similarity in neural networks with canonical correlation. In *NeurIPS*, 2018.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Ng, A. Y. and Jordan, M. I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *NeurIPS*, 2002.

Oliver, A., Odena, A., Raffel, C., Cubuk, E. D., and Goodfellow, I. J. Realistic evaluation of deep semi-supervised learning algorithms. In *NeurIPS*, 2018.

Patrini, G., Rozza, A., Menon, A. K., Nock, R., and Qu, L. Making deep neural networks robust to label noise: A loss correction approach. In *CVPR*, 2017.

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

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

Rousseeuw, P. J. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984.

Rousseeuw, P. J. and Driessen, K. V. A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, 41(3):212–223, 1999.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *ICLR*, 2014.

Wang, Y., Liu, W., Ma, X., Bailey, J., Zha, H., Song, L., and Xia, S.-T. Iterative learning with open-set noisy labels. In *CVPR*, 2018.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Zhang, Z. and Sabuncu, M. R. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, 2018.