Training Better Deep Neural Networks with Uncertainty Mining Net

Yang Sun [†‡2]
ys2899@columbia.edu
Abhishek Kolagunda [†1]
abhishek.kolagunda@ibm.com
Steven Eliuk[1]
steven.eliuk@ibm.com
Xiaolong Wang[⋆1]
visionxiaolong@gmail.com

[1] IBM
California, US
[2] ByteDance Ltd.
Beijing, China

Abstract

In this work, we consider the problem of training deep neural networks on partially labeled data with label noise. That is, semi-supervised training of deep neural networks with noisily labeled data. As far as we know, this is a scarcely studied topic. We present a novel end-to-end deep generative framework for improving classifier performance when dealing with such data challenges. We call it Uncertainty Mining Net (UMN). We utilize all the available data (labeled and unlabeled) to train the classifier via a semi-supervised generative framework. During training, UMN estimates the uncertainty of the labels to focus on clean data for learning. More precisely, UMN applies a novel sample-wise label uncertainty estimation scheme. Extensive experiments and comparisons against state-of-the-art methods on several popular benchmark datasets demonstrate that UMN can reduce the impact of label noise and significantly improve classifier performance.

1 Introduction

Deep Learning (DL) models for classification, to learn powerful representations, usually require a large amount of training data. However, for many real world problems, it is not always possible to obtain sufficiently large, well annotated training data. What one usually gets is limited training data with corrupt labels which affect the model performance. Although acquiring large data is not hard, considering the information explosion on the internet, accurate labeling is usually an expensive and error-prone task which involves human interaction, especially experts with knowledge in the specific field. In many enterprise use-cases, one has to train DL models using limited training data with corrupted data labels. It becomes very challenging to apply the current popular deep learning frameworks to solve this problem.

In this paper, we propose a framework for semi-supervised learning in the presence of label noise. Our major contributions can be summarized into the following aspects:

[†]These authors contributed equally.
[‡]The author contributed to this work while employed in IBM, California, USA.
[⋆]Corresponding author.

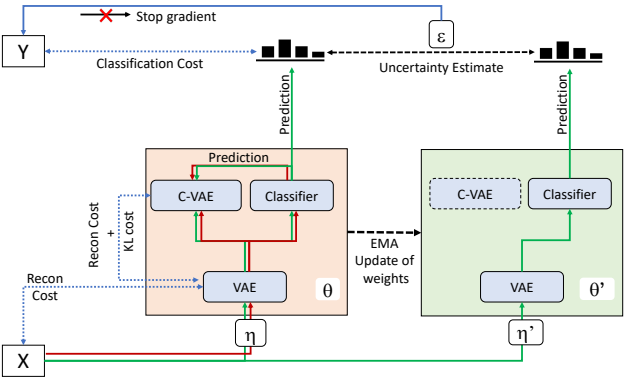


Figure 1: An overview of UMN framework: X and Y are the observed inputs and potentially corrupt labels. θ and θ' are semi-supervised generative models. η and η' are noise functions that perturb the input X. The red and green colored arrows represent the flow of unlabeled and labeled data through the models respectively. θ is updated via a stochastic gradient descent approach so as to minimize the classification and VAE losses, and θ' is the Exponential Moving Average (EMA) of θ. The sample-wise uncertainty (ε) in the predictions of θ and θ' are used to re-weigh the gradients that are back propagated to θ via the classification loss. The general architecture for θ is adapted from [8]. VAE indicates Variational AutoEncoder and C-VAE represents Conditional-VAE. Note that in θ', when making predictions (on labeled data) only the weights from encoder of the VAE and the classifier are used. For simplicity, the KL loss for the predictions on unlabeled data with respect to an uniform prior is not shown.

- Though the problem under study is common in many real world scenarios, the topic is scarce in the literature. We propose one of the first complete solutions for semi-supervised learning with noisy labels.
- We propose a generative process that models sample-wise label noise and use it to minimize the effects of label noise in training classifiers. We validate the approach through improved experimental results.
- We also propose an approach to estimate per sample label noise using the iterate moving average model. We have provided theoretical analysis/explanation for the same.
- Finally, we develop an end-to-end deep learning framework (Figure. 1) to train on partially labeled datasets with noisy labels.

2 Related Works

In recent times, the problem of training deep neural networks with noisy labeled data has started to draw attention. Natarajan et al.[15] propose the unbiased estimator of the surrogate loss function and calculate theoretical bounds for empirical risk optimization. Modeling the consistency as the regularization for deep neural network is another route for robust

learning with noisy labeled data. Reed et al.[17] design a robust loss to model the prediction consistency. In [13], the authors propose a knowledge distillation framework where they train an auxiliary model on a small set of clean data samples and linearly combine its predictions with the observed labels to form the new targets to train. Menon et al. model the corruption process of a dataset by learning the class-probability estimator [14]. Another way of regularization for training on noisy labels is to estimate the class conditional corruption ratio. Goldberger et al.[2] propose a softmax layer along with the classifier to predict the class conditional corruption ratio. In [6], they propose MentorNet which learns a data driven dynamic curriculum using a meta learning scheme. In [20] the authors present a joint optimization approach that alternates between optimizing the model parameters and the sample labels during training.

Semi-supervised learning with few labeled samples is also a well studied area. Typical related works include [3, 8, 21]. Kingma et al. [8] propose a stacked deep generative semi-supervised model by training on partially labeled datasets. They first train a variational auto-encoder on the labeled data, then stack on top of another conditional variational auto-encoder and continue training in semi-supervised fashion to get a robust classifier. In [21], the authors train a classifier in semi-supervised way using temporal ensemble approach. They train a student network and maintain a teacher network as the exponential moving average of the student. They use classification loss for labeled data and enforce consistency loss between teacher and student for unlabeled data. Semi-supervised learning approaches have also been employed to deal with noisy labels in fully labeled datasets [1, 9, 12]. This typically involves splitting the given labeled dataset into labeled (mostly clean) and unlabaled (mostly noisy) datasets and applying a semi-supervised learning approach.

There have been limited works dealing with bi-quality data [4], [11]. Bi-quality data is defined in [4] as datasets with few labeled samples where the labels are potentially corrupted. In [11], the authors discuss to use the deep generative semi-supervised model to model noisy labels. They assume that the categorical corruption rate is known which does not hold in many real scenarios. In comparison to the related works, we present a new framework for training a robust classifier with a partially labeled dataset containing label noise. We assume that pre-knowledge of label corruption rate and/or a separate clean dataset are not available during model training. This is more close to the real scenarios.

3 Method

The goal of our work is to train a robust classifier model using a dataset that has a limited set of labeled samples and has label noise. Our proposed framework consists of two major components. First, is a stacked generative model for semi-supervised learning whose generative process includes a model for sample-wise label noise. The *true* label of a sample is treated as a latent variable in the generative process. Second, is a sample-wise label uncertainty estimation scheme using the exponential moving average model. The difference between the predictions of the classifier being trained and its iterate moving average is used to estimate per-sample label uncertainty during training. The two components are combined to form an end-to-end robust learning framework that reduces the influence of potentially mislabeled samples on the classifier during the training process. The framework does not require any prior knowledge of the label corruption rates or a cleanly labeled subset for pre-training. Instead, the framework estimates sample-wise label uncertainty during model training. In the following subsections, we will provide details of the proposed framework.

3.1 Semi-Supervised Learning with Label Noise

In [8] a generative process for semi-supervised learning was proposed. The approach stacked a generative model for the data conditioned on latent class and feature variables ($M2$) over a generative model for the data with a continuous latent variable ($M1$). They also showed that, for semi-supervised learning, stacked $M1+M2$ yielded better results than using $M2$ alone or $M1$ followed by a classifier. Instead of the two separate generative models, authors in [11] proposed an unified, generalized generative process for semi-supervised learning as

$$y, z_b \sim M(y),\ \ p(z_b), z_a \sim p_\theta(z_a|z_b, y),\ x \sim p_\theta(x|z_a). \tag{1}$$

Where, M is the multinomial distribution and $p(z_b)$ is a *Normal* distribution. y indicates the true label. The latent variable z_a is jointly generated from z_b and y. z_b is the latent representation of the input. To deal with label noise, in [11] a model for mislabeling was added to the above generative process: $\hat{y} \sim p_\theta(\hat{y}\,|\,y)$. Where, y is treated as a latent variable. $\hat{y}$ is used to denote the observed labels. This term, however, modeled only categorical (class conditional) label noise. We propose to instead model sample-wise label noise as

$$\hat{y} \sim p_\theta(\hat{y}\,|\,y, z_a). \tag{2}$$

The term $p(\hat{y}\,|\,y, z_a)$ represents the likelihood of a sample's observed label $\hat{y}$ being correct given the latent label y and the sample itself. Since this term models sample-wise label correctness probability, without loss of generality, it can be used for both symmetric and asymmetric noise models.

From (1) and (2) the posterior distribution can be factorized as

$$q(z_a, z_b, y|x, \hat{y}) = q(z_a|x)\, q(y|z_a)\, q(z_b|z_a, y). \tag{3}$$

Note that the encoder q has no dependence on observed label $\hat{y}$. Following which the Evidence Lower Bound (*ELBO*) can be derived as shown in (4) (details in the supplementary materials). Where $q_\phi(y|z_a)$, $q_\phi(z_a|x)$, $q_\phi(z_b|z_a, y)$, $p_\theta(z_a|z_b, y)$, $p_\theta(x|z_a)$ indicate the classi-

$$\begin{aligned} ELBO = &- \sum_{x\in\{L,U\}} E_{q_\phi(z_a|x)} \sum_{y\in C} q_\phi(y|z_a) q_\phi(z_b|z_a, y) p(z_b) - \sum_{x\in\{L,U\}} E_{q_\phi(z_a|x)} \sum_{y\in C} q_\phi(y|z_a) p(y) \\ &- \sum_{x\in\{L,U\}} E_{q_\phi(z_a|x)} \sum_{y\in C} q_\phi(y|z_a) E_{q_\phi(z_b|z_a,y)} \big(\log p_\theta(z_a|z_b, y) - \log q_\phi(z_a|x)\big) \\ &+ \sum_{x\in\{L,U\}} E_{q_\phi(z_a|x)} \log p_\theta(x|z_a) + \sum_{x\in\{L\}} E_{q_\phi(z_a|x)} \sum_{y_k\in C} q_\phi(y_k|z_a) \log p_\theta(\hat{y}|y_k, z_a). \end{aligned} \tag{4}$$

fier, encoder, conditional encoder, conditional decoder and decoder functions respectively. L, U, C are sets of labeled data, unlabeled data and labels respectively. The last term is the labeled loss term and is the supervised portion of (4). To deal with label noise, this term uses the label uncertainty function $p_\theta(\hat{y}|y_k, z_a)$ that modulates sample weights during the gradient back-propagation. The detailed calculation of $p_\theta(\hat{y}|y_k, z_a)$ is given in the following section.

3.2 Uncertainty Estimation

We model the point-wise label correctness probability using a point-wise label uncertainty estimate $\varepsilon(z_a)$.

$$p(\hat{y}\,|\,y,z_a) = \begin{cases} 1-\varepsilon(z_a), & \text{if } \hat{y}=y \\ \varepsilon(z_a)/(C-1). & \text{otherwise} \end{cases} \tag{5}$$

The labeled loss term can then be formulated as

$$E_{q_\phi(z_a|x)}\big[f(\varepsilon(z_a))q_\phi(y|z_a)\big]. \tag{6}$$

where $f(\varepsilon(x_a)) = \log[\frac{(C-1)(1-\varepsilon(x_a))}{\varepsilon(x_a)}]$. That is, samples contribute relatively more to the gradient back-propagation when the estimated corruption rate is small which indicates that there is a high probability that the given label is correct. By focusing on these reliable targets during training, we can train a more robust classifier. From this, one can see that the key is to approximate $\varepsilon(z_a)$ precisely, and that it cannot be known apriori. We estimate $\varepsilon(z_a)$, during the training process, using the differences in predictions from the updating classifier model and its moving average model.

In comparison, the label correctness in [11] is modeled using a constant predefined category wise uncertainty ε. This assigns the same weight for all samples belonging to the same category, i.e., all labeled terms contribute equally to the gradient back propagated regardless of their label correctness. In most real scenarios the class conditional corruption rate cannot be known beforehand and it is not accurate to apply the same categorical corruption rate to all data samples of a given class.

Exponential moving average of the model weights is calculated as:

$$\theta'_t = \gamma\theta'_{t-1} + (1-\gamma)\theta_t, \tag{7}$$

where θ is the set of weight parameters of the classifier model and θ' is its exponential moving average. γ controls the smoothness of model updates. We name them *Learner* and *Guider* models respectively. t is the step index of the iterative optimization. Since this iterate average gives optimal bound for convergence rate [16] and can be less sensitive to the noisy updates, we adopt it as the *Guider* model to estimate the label uncertainty.

We estimate the label uncertainty via the absolute difference in the predicted probability of the *Learner* and the *Guider* for the observed class as following

$$\varepsilon(z_a) = |\,g'(z_a,\hat{y}) - g(z_a,\hat{y})\,|, \tag{8}$$

where g' and g are the classifier's output of the *Learner* and *Guider* respectively for the given sample z_a and observed class $\hat{y}$. As the *Learner* learns from the noisy labeled data, the *Guider* model is used to approximate label uncertainty and weigh the training samples based on the uncertainty to limit the contributions of unreliable samples to the gradients' back-propagation. It has to be noted that the value of $\varepsilon(z_a)$ (eq. 8) in our approach is not a true uncertainty measure, it is a weight factor that gives an empirical measure of relative label uncertainty. Our experiments also validate that estimating relative label uncertainty is sufficient to train robust classifiers.

An explanation of why our approach for estimating label uncertainty works is as follows. Near convergence, θ oscillates about θ' (supporting theorem and proof are in the supplementary material). Given that the minima of loss for clean samples are more tightly clustered together than that of noisy samples, θ' will converge closer to the minima associated with

clean samples. Then for each sample, the absolute difference between the predictions of θ' and θ is proportional to the gradient norm of the loss with respect to θ'. Larger the gradient higher the chance that the minimizer of the loss for a sample is far from θ' and hence more likely to be a noisy sample. This is different from filtering out samples based on prediction confidence, which does not take into account that the *minimum* of loss for some noisily labeled samples might be lower than that of the clean samples. Our approach builds an outlier rejection mechanism based on the gradient norm of the sample-wise loss. Although we use the absolute difference, there is also an exploration space for investigating other distance metrics.

UMN provides a flexible framework. Depending on the task, the encoder and decoder can be implemented using various deep learning architectures, e.g., AlexNet, GoogLeNet, ResNet, etc. Though we only show experiments on image classification problems, UMN can be applied to similar problems in other domains.

4 Experiments and Results

Our experiments are designed to evaluate whether UMN is an effective approach to learn a good model with limited annotated noisy training data. We compare UMN to popular approaches of supervised learning, semi-supervised learning and robust learning which deal with noisy labels. These methods represent some of the state-of-the-art approaches for model learning with noisy data. Further, we evaluate the performance of applying the uncertainty estimation module to identify corrupted labels.

We experiment on a variety of image classification problems with varying degrees of label corruption rates. We use three popular datasets including **MNIST**, **SVHN** and **CIFAR-10**. For a comprehensive evaluation, we set up five different uniform labels corruption rates including $[10\%, 20\%, 30\%, 40\%, 50\%]$.

4.1 Implementation Details

In the experiments, supervised deep learning framework means all the labeled data are directly used for training. We compare against a CNN architecture with 13 convolutional layers (detailed architecture in the supplementary material) and ResNet-101 architecture [5] representing supervised learning approaches. To compare against semi-supervised learning approaches, we select two popular works - Mean-Teacher (MT) [21] and Mislabeled-VAE (M-VAE) [11]. Langevin et al. [11] only briefly discuss their idea without showing much details of the model architecture and experimental results on popular benchmarks. In this work, we implement the idea in [11] and compare with UMN on all three datasets. In our experiments, we use the same encoder and decoder architectures for UMN and M-VAE.

To compare against robust learning approaches, we choose recent works including MentorNet [6], Joint Optimization framework (JO) [20] and Reweight [18]. For a fair comparison with these approaches, we follow all the original setting in these papers for our experiments. For each corruption ratio, we ran 5 experiments by randomly choosing samples to corrupt each time, and report the mean error rate.

As illustrated in Fig. 1, our framework (UMN) is composed of two encoders and two decoders (VAE and C-VAE) along with a classifier. For the experiments on MNIST, we adopt a multi-layer perceptron (MLP) architecture similar to the one proposed in [8], for the VAE, C-VAE and the classifier models. The encoders, decoders and the classifier of our

model use a single hidden layer with ReLU activation and an output layer without activations. For SVHN and CIFAR-10, we apply a 13-layer convolutional neural network (ConvNet) to build our framework (detailed architecture in the supplementary material).

We use the Adam optimizer [7] with an initial learning rate of 0.001, $\beta_1 = 0.90$ and $\beta_2 = 0.99$. The decay for the moving average model is set as described in [21]. Each experiment has 5 runs and each run has 350 epochs. In each epoch, UMN utilizes all the labeled and unlabeled training data. For the supervised learning frameworks, we only use the labeled data. We implemented the whole framework using TensorFlow and ran experiments on a single NVIDIA Tesla P100 GPU.

4.2 Datasets and Results

MNIST

MNIST is a widely used dataset in machine learning community. It includes $60,000$ images with size 28×28 pixels. In our experiments, we use $50,000$ images for training and $10,000$ images for testing. 100 labeled data samples are used in the experiments. In semi-supervised training, each mini-batch has 5 labeled samples and 95 unlabeled examples. In our compared supervised model, we apply a Multi-Layer Perceptron (MLP) classifier with one hidden layer of 784 units with ReLU activation. All the results are listed in Table 1.

From the results, we can find that UMN performs much better than other approaches. Besides UMN, M-VAE [11] also outperforms other benchmarks. One possible reason may be due to probablistic modeling of the uncertainty. However, M-VAE depends on the pre-defined label corruption ratio which can't be obtained in most real scenarios. Note that we use the ground truth corruption rates for the value of ε in M-VAE.

We did not run Reweight [18] on the MNIST dataset since the experiments an implementation of their approach on noisy MNIST data is not available and our implementation of the approach would not be a fair comparison without having the right parameter settings.

SVHN

We also experiment with the Street View House Numbers (SVHN) datasets. This dataset includes $73,257$ RGB images of 32×32 resolution belonging to ten different classes. Following the same experimental setting as in [21], we use 500 labeled data samples from SVHN where 50 data samples per category and we use the rest of the images for unlabeled data in the semi-supervised learning setting. We randomly corrupt the sample labels within each category uniformly with our defined corruption rates. In our experiment, each batch includes 5% labeled data.

The comparison results are shown in Table 1. As listed in this table, we can find that UMN behaves the best in general except for the case when the corruption ratio is 10%. However, we do not observe this phenomena in MNIST and CIFAR-10. One possible reason may be that the consistency loss term which minimizes the deviation of the *student* from the *teacher* network helps MT model alleviate the effects of noisy labels when the label corruption ratio is small. The performance of MT drops significantly as we ingest more mislabeled data. From the results, we also observe that other two robust learning approaches (MentorNet and Reweight) achieve better performance than supervised only but much lower than these Semi-supervised learning methods. Possibly because these methods do not utilize the unlabeled data.

As illustrated in Fig. 2, over-fitting is observed in the M-VAE at higher label corruption ratios. As we discuss previously, one major reason is because of the pre-defined ε which is

Table 1: Comparison of results on different benchmarks. S-1 represents the supervised learning with 13 conv layers and S-2 represents supervised learning with ResNet-101. MN and RW indicate the MentorNet and Reweight approach respectively. Error rate percentage % is used as the measurement unit.

Dataset	Corr. ratio	Approaches							
		S-1	S-2	MT	M-VAE	MN	RW	JO	**UMN(ours)**
MNIST	10%	29.4	29.5	6.6	4.3	25.6	-	10.1	$\mathbf{2.5 \pm 0.14}$
	20%	36.2	36.6	11.7	**2.7**	39.7	-	18.1	2.8 ± 0.37
	30%	41.1	43.0	14.6	7.4	45.8	-	26.7	$\mathbf{5.3 \pm 3.9}$
	40%	51.3	49.0	17.9	9.9	57.7	-	29.5	$\mathbf{5.8 \pm 3.8}$
	50%	57.5	59.4	34.4	28.0	65.3	-	55.3	$\mathbf{18.0 \pm 12.0}$
SVHN	10%	34.9	32.0	**18.0**	26.1	29.1	27.3	26.6	23.9 ± 0.2
	20%	46.3	46.0	45.5	35.2	42.1	39.0	33.3	$\mathbf{30.5 \pm 0.3}$
	30%	61.4	58.9	53.0	37.2	53.8	51.9	42.3	$\mathbf{30.9 \pm 0.6}$
	40%	61.8	60.5	63.9	40.1	59.1	53.7	56.6	$\mathbf{37.1 \pm 0.9}$
	50%	65.1	63.3	65.9	48.3	62.0	57.9	65.1	$\mathbf{43.2 \pm 1.4}$
CIFAR -10	10%	43.2	41.3	39.2	41.2	41.2	39.4	40.3	$\mathbf{37.8 \pm 0.5}$
	20%	46.3	44.7	42.1	43.9	42.4	41.9	42.5	$\mathbf{39.8 \pm 0.8}$
	30%	52.9	50.0	47.8	51.4	51.6	49.9	46.7	$\mathbf{43.5 \pm 0.5}$
	40%	57.1	56.9	52.2	52.1	54.1	53.9	48.7	$\mathbf{43.5 \pm 1.2}$
	50%	63.3	61.3	65.4	56.3	57.1	58.3	55.4	$\mathbf{51.9 \pm 1.7}$

used as the constant value during the training. In contrast, UMN applies the sample-wise uncertainty estimation. Furthermore, the training of UMN also converges faster with better performance.

We also conduct experiments to evaluate the model performance with different labeled data scales in the SVHN dataset. We compare UMN with a popular supervised robust learning approach MentorNet (MN) [6], for which the test errors are illustrated in Table 2. In UMN, the training data also covers these unlabeled data which can't be used by MN. UMN shows better performance when training data has less than 1000 labeled data samples per class. MN gets better performance as the size of the labeled data increases, this also coincides with the findings in [19] about the robustness of DNNs to label noise when there is enough labeled data. But, when there is limited annotated data available, like in many real use-cases, UMN can be very useful for robust training of DNN classifiers.

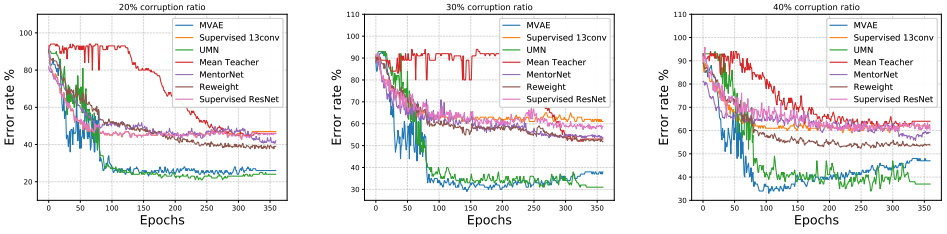
In addition, we evaluate the performance of UMN with higher corruption rates and compare it with MentorNet [6]. UMN achieves error rates of 49.1%, 53.1% and 54.5% which is significantly lower than 67.1%, 70.0% and 71.2% achieved by MentorNet for label corruption ratios of 60%, 70% and 80% respectively.

Table 2: Test errors rates with varying number of labels for SVHN

Num labels	1K	2K	3K	10K	20K
UMN	23.9	18.1	16.3	14.8	13.2
MentorNet	30.5	27.9	25.1	18.3	11.2

CIFAR-10

Next, we run the comparisions on CIFAR-10 dataset [10]. This dataset contains 32×32



20% corruption ratio 30% corruption ratio 40% corruption ratio

Figure 2: Illustration of error rates' comparisons between UMN and other benchmarks on SVHN during training stage under different corruption ratios (20%, 30% and 40%). UMN is less prone to overfitting on mislabeled data at higher corruption ratio.

pixels RGB images belonging to ten different classes. To have a fair comparison, we follow the same experimental setting as used in [21]. We randomly select 100 data samples from each category. In total, 4,000 labeled data samples are included in the training set. The rest of the training data are used for unlabeled data. The results are summarized in Table 1. From the listed results, we can see that UMN achieves the best performance compared to all other approaches.

From the experimental results, one can see that M-VAE is, in most cases, the second best behind UMN. Though there are similarities in how mislabeling is modeled in our generative processes, the use of sample-wise label uncertainty estimation in UMN is shown to be a better approach for robust training of DNN classifiers.

4.3 Identifying samples with corrupt labels

We conduct experiments using the MNIST-digit dataset to quantify how well UMN can identify samples whose labels are corrupted. We say, a sample's label is corrupted if there is a disagreement between the predictions of the *Learner* and the *Guider* models. That is, if the estimated $\varepsilon > 0$ (Eq. 8) for a given sample. Table 3 summarizes the accuracy in identifying samples with corrupt labels. The sample-wise estimate of ε obtained in the final epoch of training is used in the analysis. From the listed results, we can see that UMN has demonstrated very promising potential for filtering out noisy data. It could be easily adopted to other related works as the data pre-processing (cleaning) step.

Table 3: Performance of identifying samples with corrupt labels.

Corruption rate	10%	20%	30%	40%	50%
Recall	1.0	0.85	0.71	0.85	0.8
Precision	0.32	0.55	0.71	0.79	0.7
F1 score	0.48	0.67	0.71	0.82	0.75

5 Conclusion

In this work, we target a new problem of training a DNN classifier on partially labeled data with label noise and propose a novel framework (UMN). UMN is an end-to-end semi-

supervised deep generative framework. It can help train a better classification model with limited, noisily annotated training data. In addition, UMN can be used to explicitly approximate the label uncertainty for a given potentially mislabeled data sample. Compared to previous works, UMN does not need a subset of cleanly labeled data or any pre-knowledge of the label corruption model and rate for robust training of classifiers. UMN directly provides sample-wise label uncertainty estimation via the prediction differences between the classifier being trained and its moving average. Experimental results demonstrate the superiority of UMN over state-of-the-arts on popular benchmarks.

References

[1] Yifan Ding, Liqiang Wang, Deliang Fan, and Boqing Gong. A semi-supervised two-stage approach to learning from noisy labels. In *WACV*. IEEE, 2018.

[2] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. 2016.

[3] Jonathan Gordon and José Miguel Hernández-Lobato. Bayesian semisupervised learning with deep generative models. *arXiv preprint arXiv:1706.09751*, 2017.

[4] Ryuichiro Hataya and Hideki Nakayama. Unifying semi-supervised and robust learning by mixup. In *ICLR The 2nd Learning from Limited Labeled Data (LLD) Workshop*, 2019.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[6] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. *arXiv preprint arXiv:1712.05055*, 2017.

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[8] Diederik P. Kingma, Danilo Jimenez Rezende, Shakir Mohamed, and Max Welling. Semi-supervised learning with deep generative models. *CoRR*, abs/1406.5298, 2014. URL http://arxiv.org/abs/1406.5298.

[9] Kyeongbo Kong, Junggi Lee, Youngchul Kwak, Minsung Kang, Seong Gyun Kim, and Woo-Jin Song. Recycling: Semi-supervised learning with noisy labels in deep neural networks. *IEEE Access*, 7:66998–67005, 2019.

[10] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[11] Maxime Langevin, Edouard Mehlman, Jeffrey Regier, Romain Lopez, Michael I. Jordan, and Nir Yosef. A deep generative model for semi-supervised classification with noisy labels. *CoRR*, abs/1809.05957, 2018. URL http://arxiv.org/abs/1809.05957.

[12] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

[13] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *ICCV*, 2017.

[14] Aditya Menon, Brendan Van Rooyen, Cheng Soon Ong, and Bob Williamson. Learning from corrupted binary labels via class-probability estimation. In Francis Bach and David Blei, editors, *ICML*, Lille, France, 07–09 Jul 2015. URL http://proceedings.mlr.press/v37/menon15.html.

[15] Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *NIPS*, 2013. URL http://papers.nips.cc/paper/5073-learning-with-noisy-labels.

[16] B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM J. Control Optim.*, 30(4):838–855, July 1992. ISSN 0363-0129. doi: 10.1137/0330046. URL http://dx.doi.org/10.1137/0330046.

[17] Scott E. Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *ICLR*, 2015. URL http://arxiv.org/abs/1412.6596.

[18] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018.

[19] David Rolnick, Andreas Veit, Serge Belongie, and Nir Shavit. Deep learning is robust to massive label noise. *arXiv preprint arXiv:1705.10694*, 2017.

[20] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *CVPR*, 2018.

[21] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NIPS*. Curran Associates, Inc., 2017.