UNIFYING SEMI-SUPERVISED AND ROBUST LEARNING BY MIXUP

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep supervised learning methods require cleanly labeled large-scale datasets, but collecting such data is difficult or sometimes impossible. There exist two popular frameworks to alleviate this problem: semi-supervised learning and robust learning to label noise. Although they relax supervised learning's restriction, these frameworks are studied independently. As a result, it is not known that which training scheme is suitable when only small cleanly-labeled data are available. In this paper, we regard learning from bi-quality data as a generalization of these studies, in which a small portion of data is cleanly labeled and the rest is corrupt. Under this setting, we compare recent algorithms for semi-supervised and robust learning. The results suggest that semi-supervised learning can outperform robust learning with noisy labels. We also propose a training strategy for mixing mixup techniques to learn from such bi-quality data effectively.

1 INTRODUCTION

Learning from imperfect data is essential for applying machine learning, especially data-hungry deep learning, to real-world problems. One approach to handle this problem is semi-supervised learning (SSL), where training data consist of a small amount of labeled data and a large amount of unlabeled data. Another way is robust learning to label noise (RLL), in which all data are labeled, but some of them are mislabeled.

SSL leverages large unlabeled data to improve performance from supervised learning on a limited number of labeled data. In the context of deep SSL, one effective method is to train neural networks to keep consistency under a small perturbation of unlabeled inputs (Miyato et al. (2018); Tarvainen & Valpola (2017); Verma et al. (2018)), which is referred as consistency regularization in Oliver et al. (2018).

In RLL setting, learners need to enhance their performance using corrupted labels and avoid performance decrease because of such data. This requirement is particularly important for deep neural networks because they have ample capacity to remember whole samples even if their labels are completely random (Arpit et al. (2017); Zhang et al. (2017a)). To handle this problem, some methods use a small amount of clean data to estimate noise transition matrix (Xiao et al. (2015); Hendrycks et al. (2018)) or to learn to select possibly correctly-labeled samples (Jiang et al. (2018); Jenni & Favaro (2018)).

While both SSL and RLL aim to alleviate the limited data problem, they are studied independently and evaluated with different benchmarks. However, if a small amount of clean data is available, they can be regarded as similar problems. Then, *can RLL outperform SSL under the same setting?* This question is our initial motivation to unify these two lines of research.

In this paper, we introduce a generalization of SSL and RLL, following the idea of trusted data (Charikar et al. (2017); Hendrycks et al. (2018)) in the literature of RLL. More precisely, we suppose some labels are guaranteed to be clean and the rest are noisy. By controlling the ratio of corrupted labels to all labels and noisiness of label corruption, the two learning frameworks can be unified.

Using the shared evaluation procedure in Oliver et al. (2018), we compare recent SSL and RLL algorithms using image classification task and find out that existing RLL methods using a small amount of clean data cannot outperform SSL under this setting. This finding suggests that such RLL algorithms cannot use noisy labels effectively.

Therefore, it is required to adaptively employ SSL and RLL in a data-driven manner. As a baseline learning algorithm, we propose to combine mixup losses for SSL (Verma et al. (2018)) and RLL (Zhang et al. (2017b)), which shows competitive results with SSL and RLL specific methods and effective use of useful information from noisy labels.

2 LEARNING FROM BI-QUALITY DATA

In this section, we describe the setting of learning from bi-quality data, a generalization of SSL and RLL. In this formulation, we suppose the given data consists of two parts: trusted data $\mathcal{D}_T$, whose labels are always correct, and untrusted data $\mathcal{D}_U$, some of those labels might be wrong. Learners are allowed to access the information whether each sample is from $\mathcal{D}_T$ or $\mathcal{D}_U$.

Let denote the ratio of trusted and untrusted data to the entire data $p = \frac{|\mathcal{D}_T|}{|\mathcal{D}_T|+|\mathcal{D}_U|}$ and $1 - p = \frac{|\mathcal{D}_U|}{|\mathcal{D}_T|+|\mathcal{D}_U|}$. We also introduce quality as

$$q = 1 - \frac{\mathcal{D}(\mathbf{p}_U(y|x)||\mathbf{p}_T(y|x))}{\mathcal{D}(\mathbf{p}(y)||\mathbf{p}_T(y|x))}, \tag{1}$$

where $\mathbf{p}_T(y|x)$ and $\mathbf{p}_U(y|x)$ are conditional probabilities of labels when inputs are given of trusted data and untrusted data, respectively, and $\mathcal{D}$ is a divergence between two probability distributions such as Kullback-Leibler divergence. In equation 1, $q \in [0, 1]$. Obviously, $q = 0$ if the labels are completely independent of the input features (that is, completely random) and $q = 1$ if the labels are clean. We suppose the quality of untrusted data q_U is in $[0, 1]$ and of trusted data q_T is 1.

We believe this setting is realistic. Suppose there are unlabeled data. Under budget constraint, one strategy to label this dataset as training data is to divide it to two parts and spend most budget on high-quality labeling to acquire D_T (e.g., using experts) and the rest on lower-quality labeling to obtain D_U (e.g., using crowdsourcing), where usually $|D_T| \ll |D_U|$.

Under this generalized framework, SSL is equivalent to a particular case where $q = 0$, that is, labels of untrusted data are entirely random and give no useful information. In SSL, such labels are usually ignored, and the data are treated as unlabeled. Meanwhile, RLL without trustable data is another special case, where $p = 0$ and $0 \lneq q < 1$. Some work of RLL uses $p > 0$ setting explicitly (Charikar et al. (2017); Li et al. (2017); Hendrycks et al. (2018)) and implicitly (Jiang et al. (2018); Jenni & Favaro (2018)).

Intuitively, when q is relatively high, we can expect some performance gain by utilizing untrusted, but somewhat informative, labels with RLL. Contrary, when q is almost zero, untrusted data is no more informative and just using standard SSL might result in better generalization. However, a critical problem in practice is that we cannot know the exact quality of untrusted data, and thus we cannot decide the best learning strategy in advance. To handle such data, we need an adaptive mechanism to properly fuse SSL and RLL in a data-driven manner.

To realize this goal, whose quality might be 0, we propose to combine techniques for SSL and RLL. Precisely, we employ the convex combination of loss functions for SSL $\mathcal{L}_{\text{semi}}$ and RLL $\mathcal{L}_{\text{robust}}$. That is, with an additional hyperparameter γ, the loss function for untrusted data is defined as

$$\mathcal{L}_U = \gamma \mathcal{L}_{\text{robust}} + (1 - \gamma)\mathcal{L}_{\text{semi}}. \tag{2}$$

For $\mathcal{L}_{\text{robust}}$ and $\mathcal{L}_{\text{semi}}$, any loss functions for SSL and RLL can be used. In this paper, as a baseline of this setting, we propose to combine *mixup* techniques for SSL (Verma et al. (2018)) and RLL (Zhang et al. (2017b)).

3 MIXMIXUP

mixup is a regularization technique, where neural network models are trained on virtual training pairs $(\tilde{x}, \tilde{y})$, where

$$\tilde{x} = \lambda x_i + (1 - \lambda)x_j, \;\; \tilde{y} = \lambda y_i + (1 - \lambda)y_j. \tag{3}$$

(x_i, y_i) and (x_j, y_j) are sampled from training data, and $\lambda \in [0, 1]$ is sampled from Beta distribution. Zhang et al. (2017b) shows this method alleviates the performance decrease under label corruption.

Algorithm 1 mixmixup: learning from bi-quality data

Prepare trusted set: $\mathcal{D}_T$, untrusted set: $\mathcal{D}_U$
Initialize neural network f
Set hyper parameters: α, β, γ
Set loss function: $\mathcal{L}(\cdot, \cdot)$: categorical cross entropy

for $k \in \{0, 1, \ldots, K-1\}$:
  Sample $\lambda_\alpha \sim \mathrm{Beta}(\alpha, \alpha), \lambda_\beta \sim \mathrm{Beta}(\beta, \beta)$
  Sample (x_i, y_i), (x_j, y_j) from $\mathcal{D}_T$
  $\mathcal{L}_T = \mathcal{L}(f(\lambda_\alpha x_i + (1-\lambda_\alpha)x_j), \lambda_\alpha y_i + (1-\lambda_\alpha)y_j)$
  Sample (x'_i, y'_i), (x'_j, y'_j) from $\mathcal{D}_U$
  Predict $y''_i = f(x'_i)$ and $y''_j = f(x'_j)$
  $\mathcal{L}_{\text{robust}} = \mathcal{L}(f(\lambda_\alpha x'_i + (1-\lambda_\alpha)x'_j), \lambda_\alpha y'_i + (1-\lambda_\alpha)y'_j)$
  $\mathcal{L}_{\text{semi}} = \mathcal{L}(f(\lambda_\beta x'_i + (1-\lambda_\beta)x'_j), \lambda_\beta y''_i + (1-\lambda_\beta)y''_j)$
  $\mathcal{L}_U = \gamma\mathcal{L}_{\text{robust}} + (1-\gamma)\mathcal{L}_{\text{semi}}$
  Update parameters of f with $\mathcal{L}_T + \sigma(k)\mathcal{L}_U$ ▷ For σ, see Section 3

In Verma et al. (2018), mixup is used for SSL as consistency regularization (Oliver et al. (2018)). Here, for unlabeled inputs $x_i^{\mathbf{U}}, x_j^{\mathbf{U}}$, neural networks are learned to map $\lambda x_i^{\mathbf{U}} + (1-\lambda)x_j^{\mathbf{U}}$ to $\lambda y_i^U + (1-\lambda)y_j^U$, where $y_i^{\mathbf{U}}, y_j^{\mathbf{U}}$ are predicted outputs corresponding to $x_i^{\mathbf{U}}, x_j^{\mathbf{U}}$.

We show the details in Algorithm 1. The parameters of neural networks are updated with a combination of losses on trusted data $\mathcal{L}_T$ and untrusted data $\mathcal{L}_U$ as $\mathcal{L}_T + \sigma(k)\mathcal{L}_U$. We use sigmoid scheduling $\sigma(k)$ as Verma et al. (2018).

Since this method mixes mixup losses, we refer to our method *mixmixup* in the rest of this paper.

4 EXPERIMENTS

4.1 SETTINGS

We use CIFAR-10 (Krizhevsky (2009)), which has 50,000 of 10-category images. Following the common protocol of SSL in Oliver et al. (2018), we split the training dataset to 45,000 for training and 5,000 for validation. To simulate bi-quality data, we sample 4,000 examples from the training data as trusted data and the rest as untrusted data. We randomly replace each label of untrusted samples with another one with a given probability, which is a common protocol in RLL (Zhang et al. (2017b); Jiang et al. (2018)). We employ the non-corrupted validation data and test data for hyperparameter tuning and final evaluation, respectively.

Following Oliver et al. (2018) and Verma et al. (2018), we use WRN-28-2 (Wide ResNet 28-2, proposed in Zagoruyko & Komodakis (2016)) as image classifier. To optimize the network, we use SGD with the learning rate of 0.1, momentum of 0.9, weight decay of 1.0×10^{-4} and mini-batch size of 256. We train networks for 1.6×10^5 iterations, following Verma et al. (2018). We implement the model with PyTorch v1.0 (Paszke et al. (2017)) and tune hyperparameters α, β and γ in Algorithm 1 with Bayesian optimization algorithm in Optuna v0.8 [1].

4.2 RESULTS

4.2.1 LEARNING FROM BI-QUALITY DATA

According to the settings in Section 4.1, we trained WRN-28-2 with bi-quality data of different quality. We randomly replace the 40% and 100% of labels of untrusted data to simulate label corruption. From the definition of quality (equation 1), the quality of these data is 0.6 and 0.0, respectively. The former corresponds to RLL with trusted data, and the other corresponds to SSL.

We show the test accuracy in Table 1. The results of models trained with the standard training method referred to as "Basic". The results show that mixmixup can regularize neural networks even when untrusted data are entirely random ($q = 0$, Table 1 C), while the standard training method leads

[1] `https://optuna.org/`

Table 1: **Our mixmixup can handle bi-quality data effectively.** Test accuracy of WRN-28-2 trained on bi-quality CIFAR-10. In the columns of TRUSTED and UNTRUSTED, we show the number of samples of trusted and untrusted data with quality q.

	METHOD	TRUSTED	UNTRUSTED	ACCURACY
A	Basic	4,000	N/A	0.72
	input mixup (Zhang et al. (2017b))	4,000	N/A	0.78
B	Basic	4,000	41,000 ($q = 0.0$)	0.29
	Basic	4,000	41,000 ($q = 0.6$)	0.78
C	mixmixup (ours)	4,000	41,000 ($q = 0.0$)	0.88
	mixmixup (ours)	4,000	41,000 ($q = 0.6$)	0.90

Table 2: **Semi-supervised methods can surpass robust learning methods under the shared setting.** Test accuracy of WRN-28-2 trained on CIFAR-10 with other state-of-the-art methods for semi-supervised learning and robust learning. † and †† refer to the scores in the tables are from Oliver et al. (2018) or our re-implementations, respectively.

	METHOD	TRUSTED	UNTRUSTED	ACCURACY
A	input mixup (Verma et al. (2018))	4,000	41,000 (no label)	0.89
	Mean teacher† (Tarvainen & Valpola (2017))	4,000	41,000 (no label)	0.84
	VAT† (Miyato et al. (2018))	4,000	41,000 (no label)	0.86
B	MentorNet DD-MLP†† (Jiang et al. (2018))	4,000	41,000 ($q = 0.6$)	0.87
	GLC††(Hendrycks et al. (2018))	4,000	41,000 ($q = 0.6$)	0.84
C	mixmixup (ours)	4,000	41,000 ($q = 0.0$)	0.88
	mixmixup (ours)	4,000	41,000 ($q = 0.6$)	0.90

to significant performance decrease (Table 1 B). Besides, compared to the model trained with mixup on 4,000 samples (Table 1 A), mixmixup yields higher performance, which suggests mixmixup can exploit untrusted data to boost the performance.

4.2.2 COMPARISON WITH OTHER METHODS

In table 2, we show the test accuracy of SSL methods (Verma et al. (2018); Tarvainen & Valpola (2017); Miyato et al. (2018), Table 2 A) and RLL methods with trusted data (Jiang et al. (2018); Hendrycks et al. (2018), Table 2 B) with the results of mixmixup (Table 2 C). In all experiments, the same network (WRN-28-2) and the same dataset split mentioned above are used. Note that SSL methods in Table 2 A does not use labels of untrusted data.

Surprisingly, our results suggest SSL (Table 2 A) might be superior to RLL (Table 2 B) with trusted data under the same setting. This result suggests the difficulty of learning robustly from partially corrupted labels preventing overfitting.

On the other hand, mixmixup can handle both SSL and RLL situation. Moreover, the accuracy when the quality is 0.6 (corresponds to RLL) is superior to one when the quality is 0.0 (corresponds to SSL), which shows that mixmixup can effectively employ information from corrupted labels.

5 CONCLUSION

In this paper, we introduce a novel framework of weakly supervised learning by unifying semi-supervised learning and robust learning to label noise, which have been independently studied. To handle this problem, we propose to mix *mixup* for semi-supervised and robust learning. This method empirically works well and achieves competitive results with semi-supervised and robust learning specific methodologies.

In addition, our experiments indicate the performance of some robust learning with trusted data might be inferior to that of semi-supervised learning under the shared setting. This result suggests that existing robust learning methods cannot effectively exploit the information which should be extracted from noisy labels.

In this paper, our proposal does not use the estimated quality and instead introduces some hyper-parameters. The use of quality estimation might ease the hyperparameter tuning but still is an open question.

REFERENCES

Devansh Arpit, Stanisaw Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A Closer Look at Memorization in Deep Networks. In *ICML*, 2017.

Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from Untrusted Data. In *STOC*, 2017.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using Trusted Data to Train Deep Networks on Labels Corrupted by Severe Noise. In *NIPS*, 2018.

Simon Jenni and Paolo Favaro. Deep Bilevel Learning. In *ECCV*, 2018.

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. In *ICML*, 2018.

Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. Technical report, 2009.

Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li Jia Li. Learning from Noisy Labels with Distillation. *ICCV*, 2017.

Takeru Miyato, Shin Ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning. *TPAMI*, 2018.

Avital Oliver, Augustus Odena, Colin Raffel, Ekin D. Cubuk, and Ian J. Goodfellow. Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. In *NIPS*, 2018.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS WS*, 2017.

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

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. Manifold Mixup: Better Representations by Interpolating Hidden States. 2018. URL `http://arxiv.org/abs/1806.05236`.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. *CVPR*, 2015.

Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. In *BMVC*, 2016.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017a.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In *ICLR*, 2017b.