Federated Label-Noise Learning with Local Diversity Product Regularization

Xiaochen Zhou, Xudong Wang

Shanghai Jiao Tong University
xiaochenzhou@sjtu.edu.cn, wxudong@ieee.org

Abstract

Training data in federated learning (FL) frameworks can have label noise, since they must be stored and annotated on clients' devices. If trained over such corrupted data, the models learn the wrong knowledge of label noise, which highly degrades their performance. Although several FL schemes are designed to combat label noise, they suffer performance degradation when the clients' devices only have limited local training samples. To this end, a new scheme called federated label-noise learning (FedLNL) is developed in this paper. The key problem of FedLNL is how to estimate a noise transition matrix (NTM) accurately in the case of limited local training samples. If a gradient-based update method is used to update the local NTM on each client's device, it can generate too large gradients for the local NTM, causing a high estimation error of the local NTM. To tackle this issue, an alternating update method for the local NTM and the local classifier is designed in FedLNL, where the local NTM is updated by a Bayesian inference-based update method. Such an alternating update method makes the loss function of existing NTM-based schemes not applicable to FedLNL. To enable federated optimization of FedLNL, a new regularizer on the parameters of the classifier called local diversity product regularizer is designed for the loss function of FedLNL. The results show that FedLNL improves the test accuracy of a trained model by up to 25.98%, compared with the state-of-the-art FL schemes that tackle label-noise issues.

Introduction

Labels of training data are indispensable for a supervised learning task. In a federated learning (FL) framework, the labels need to be annotated on clients' devices since the training data are kept by clients' devices. However, some of the labels can be wrong due to carelessness or a lack of expert knowledge. Those wrong labels are referred to as noisy labels and the corresponding training samples are referred to as noisy samples. When a classifier (e.g., a neural network) is trained over such a training dataset with noisy labels, it learns wrong knowledge from such noisy labels, which highly degrades its performance.

So far several FL schemes have been developed to tackle the label-noise issue. These schemes can be classified into the following categories according to the utilized techniques.

Schemes	5000 samples/clients	500 samples/clients
FedLSR	82.7	78.3
FedCorr	86.3	75.3
RoFL	89.9	80.4
VolMinNet-FL	90.4	69.1
Accurate NTM	90.7	89.2

Table 1: Test accuracies ($\%$) of the selected schemes over CIFAR-10 dataset under different settings of local training samples (pair-flipping noise, noise rate 0.4).

The first category (e.g., FedCorr (Xu et al. 2022)) utilizes a label-correction mechanism to relabel noisy labels. The second category (e.g., RoFL (Yang et al. 2022b) and FedLSR (Jiang et al. 2022)) leverages self-supervised learning to obtain more robust representations. Besides the above two categories, another type of FL schemes can be developed by extending a noise transition matrix-based scheme to the FL framework. For example, the state-of-the-art NTM-based scheme called VolMinNet (Li et al. 2021) can be extended to the FL framework (denoted as VolMinNet-FL) by minimizing the loss function of VolMinNet via FedAvg (McMahan et al. 2017). The performance of these schemes is evaluated over **CIFAR-10** dataset with a high noise rate under two settings: one with 5000 samples on each client's device and the other with 500 samples on each client's devices, as presented in Table 1. The results show that only RoFL and VolMinNet-FL achieve acceptable accuracies when there are enough local training samples (i.e. 5000 samples per client), while all the four schemes suffer low accuracies when there are only limited local training samples (i.e., 500 samples per client). However, it is common that most clients' devices in an FL system only have limited local training samples (McMahan et al. 2017). Thus, it is necessary to develop a new FL scheme to tackle the label-noise issue, especially in the case of limited local training samples.

According to Table 1, self-supervised learning and NTM-based schemes are two promising approaches to tackling the label-noise issue in the FL framework, since they are effective to combat label noise when there are enough local training samples. However, when there are only limited lo-

cal training samples, self-supervised learning is less feasible because it is difficult to learn robust representations with limited training samples. On the other hand, if an accurate NTM is obtained, e.g., for VolMinNet-FL, such a scheme (denoted as Accurate NTM in Table 1) can achieve a high accuracy with limited local training samples. Thus, this paper develops a new FL scheme called federated label-noise learning (FedLNL), by using NTM.

In FedLNL, the NTM and the classifier need to be learned in a federated manner. Note that if the NTM is accurately estimated, the classifier can be trained by minimizing the robust loss of LNL with an FL algorithm. Thus, the problem lies in how to estimate the NTM in a federated manner. In the CL setting, the state-of-the-art estimation methods (e.g., (Li et al. 2021)) regard the NTM as a trainable matrix, and train the NTM and the classifier simultaneously with a gradient-based optimization algorithm. If applying these methods to the clients' devices, the gradients for a local NTM can be too large due to limited local training samples, causing the local NTM to converge to a solution with a high estimation error. To this end, an alternating update method is developed in FedLNL to estimate the local NTM and train the local classifier. More specifically, in each local iteration, a local NTM is sampled from a prior distribution constructed by K Dirichlet distributions where K is the number of classes. When updating the prior distribution, the likelihood of the local NTM is first determined, based on the prediction of the local classifier and the noisy labels. The prior distribution is then updated by determining the posterior distribution of the local NTM via Bayes' rule, using the current prior distribution and the likelihood. When updating the local classifier, the local NTM is fixed and substituted into the local loss function of FedLNL. The local classifier is then updated by minimizing the local loss function. In the step of updating the prior distribution, the values in the likelihood are upper bounded by the number of local training samples, making the update of the prior distribution more stable than the gradient-based method.

As the NTM is no longer a trainable matrix in FedLNL, the loss functions of the state-of-the-art LNL schemes (e.g., (Li et al. 2021)) are not suitable for FedLNL. The reason is that these loss functions impose regularizers on the NTM to guarantee the uniqueness of the optimal solution. Such regularizers are effective only when the NTM is trainable. To design a loss function for FedLNL, a regularizer on the parameters of the classifier is first derived, which is actually a diversity-promoting regularizer (Malkin and Bilmes 2008) on the softmax outputs of all the training samples. However, the diversity-promoting regularizer couples the softmax outputs of all the training samples, making it difficult to minimize the loss function of FedLNL in a federated manner. A decomposable regularizer called local diversity product (LDP) regularizer is then designed to enable federated optimization for FedLNL. It is proved that FedLNL with LDP regularizer can achieve the same optimal solution to the original optimization problem of FedLNL with the diversity-promoting regularizer.

The performance of FedLNL is evaluated with extensive experiments. The effectiveness of the alternating update method and LDP regularizer is first verified in an ablation study. The overall performance of FedLNL is then evaluated in the case of limited local training data samples. Compared with the state-of-the-art FL schemes, FedLNL improves the test accuracy of a trained classifier by up to 25.98%.

Related Work

Noise Transition Matrix-Based Schemes

Label-noise learning (LNL) is a class of methods that combat label noise with theoretical guarantees. It estimates a stochastic matrix, i.e., NTM, to summarize the probability that one class is mistaken for another class. Based on the NTM, a corrected loss is designed to train a neural network with noisy labels. It has been proved that such a neural network can converge to the neural network trained with the correct labels, as the number of training samples goes to infinity.

The vanilla LNL (Patrini et al. 2017) requires two elements to estimate the NTM: 1) a neural network that is trained with the noisy labels; 2) anchor points of each class of training data. Here the anchor points are the data instances whose labels are correct with a probability of 1 (or close to 1). As a result, this scheme needs to train an additional neural network for estimating the NTM, which increases the computational cost. As for anchor points, it is not practical to assume that there always exist anchor points in training data. To reduce the computational cost, some end-to-end label-noise learning schemes are developed to learn an NTM and a neural network simultaneously. If the NTM is unknown and directly minimize the corrected loss to determine both the NTM and the neural network, there exist an infinite number of solutions. Hence, these end-to-end schemes design new types of regularization to constrain the training of the NTM (e.g., (Li et al. 2021)) or that of the neural network (e.g., (Zhang, Niu, and Sugiyama 2021)). Moreover, some schemes try to relieve the requirement on anchor points. In (Xia et al. 2019), a coarse NTM is first estimated from a training dataset without anchor points. The coarse NTM is then modified by adding a slack variable that can be learned together with the neural network with the noisy data. In (Li et al. 2021), the estimation of the NTM is reformulated into a problem of minimizing the volume of the simplex whose vertices are the rows of the NTM. Solving this problem requires no anchor points.

The above end-to-end LNL schemes are designed for the CL setting. If applying them to the FL framework, the local NTMs trained on clients' devices can overfit the limited local training data. Consequently, only a global NTM with a high estimation error is obtained.

Label-Noise Learning Schemes

The label-noise issue has been widely studied in the setting of centralized learning (CL). Many developed schemes rely on the techniques such as sample re-weighting (Han et al. 2018; Jiang et al. 2018; Chen et al. 2019; Mirzasoleiman, Cao, and Leskovec 2020; Huang, Zhang, and Zhang 2020) and label correction (Kun and Jianxin 2019; Guo et al. 2020; Zheng et al. 2020; Zheng, Awadallah, and Dumais 2021;

Kye et al. 2022). Sample re-weighting provides the data samples whose labels are more likely to be correct with larger weights during model training so that the neural network learns fewer patterns of the noisy labels. Label correction aims at correcting the noisy labels to boost the accuracy of the trained classifier. Besides, self-supervised representation learning is also adopted by some schemes, e.g., JointOpt (Tanaka et al. 2018) and DivideMix (Li, Socher, and Hoi 2020). However, if applying these schemes to the FL framework, their test accuracies can highly degrade due to the limite local training samples on clients' devices.

In the FL framework, many schemes have also been developed to tackle the issue of noisy labels. The mechanisms proposed in these schemes can classified into three categories: client selection, label-correction, and self-supervised learning. The first categorie utilizes client selection to select the clients' devices with fewer noisy labels more frequently, so as to mitigate the impacts of noisy labels on FL (Chen et al. 2020; Yang et al. 2022a; Wang et al. 2022; Yang et al. 2022b). Most of them need an additional clean dataset to guide the selection, while such a clean dataset is not always available. The second category utilizes a label-correction mechanism to relabel noisy labels, based on the representations extracted from the training data, e.g., the nearest neighbors in the embedding space (Tsouvalas et al. 2022) and the prediction of the global model (Xu et al. 2022). The third category (e.g., RoFL (Yang et al. 2022b) and FedLSR (Jiang et al. 2022)) leverages self-supervised learning to obtain more robust representations. Although these schemes are designed for the FL framework, they still suffer performance degradation when the clients' devices only have limited local training samples.

Preliminaries

Consider an FL framework consisting of one central server and M clients' devices. Let $\mathcal{M}$ denote set $\{1, ..., M\}$. Each device $m \in \mathcal{M}$ holds a local training dataset with N_m data samples. The sample IDs of these N_m data samples constitute a set denoted as $\mathcal{L}_m$. Let $\mathbf{x}_i, i \in \mathcal{L}_m$ denotes one data sample on device m, and its noisy label is denoted as $\tilde{y}_i \in \{1, ..., K\}$ where K is the total number of classes. Note that $\tilde{y}_i$ can be different from its corresponding clean label denoted as $y_i \in \{1, ..., K\}$, due to the existence of label noise. Define the clean class posterior for a data sample $\mathbf{x}$ as $P(\mathbf{Y}|X = \mathbf{x}) = [P(Y = 1|X = \mathbf{x}), ..., P(Y = K|X = \mathbf{x})]^\top$, where Y and X represent the random variable of true labels and data samples, respectively. A classifier (e.g., a neural network) is then trained in a federated manner over these noisy data samples, in order to model the clean class posterior. The classifier is denoted as $f(\cdot; \mathbf{w})$ where $\mathbf{w}$ represents its parameters.

Define the noisy class posterior as $P(\tilde{\mathbf{Y}}|X = \mathbf{x}) = [P(\tilde{Y} = 1|X = \mathbf{x}), ..., P(\tilde{Y} = K|X = \mathbf{x})]^\top$ where $\tilde{Y}$ represents the random variable of noisy labels. If the classifier is trained via traditional supervised learning, it can only approach the noisy class posterior due to the existence of label noise. To tackle the label-noise issue, label-noise learning (LNL) (Patrini et al. 2017) is adopted in this paper. LNL models the generation process of label noise with a noise transition matrix (NTM) $\mathbf{T}(\mathbf{x}) \in \mathbb{R}^{K \times K}$. Each item $T_{i,j}(\mathbf{x})$ of $\mathbf{T}(\mathbf{x})$ represents the probability that a data sample $\mathbf{x}$ with a clean label $y = i$ is mislabeled by a noisy label $\tilde{y} = j$, i.e.,

$$T_{i,j}(\mathbf{x}) = P(\tilde{Y} = j|Y = i, X = \mathbf{x}).$$

With $\mathbf{T}(\mathbf{x})$, the noisy class posterior and the clean class posterior can be connected via

$$P(\tilde{\mathbf{Y}}|X = \mathbf{x}) = \mathbf{T}(\mathbf{x})^\top P(\mathbf{Y}|X = \mathbf{x}). \quad (1)$$

Recall that the noisy class posterior can be directly estimated from the noisy data samples. If $\mathbf{T}(\mathbf{x})$ is also known for each data sample $\mathbf{x}$, the clean class posterior can be determined by minimizing the following loss function via an FL algorithm:

$$\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} L(\tilde{y}_i, \mathbf{T}(\mathbf{x}_i)^\top f(\mathbf{x}_i; \mathbf{w})), \quad (2)$$

where $L(\cdot, \cdot)$ denotes the cross-entropy loss.

It can be seen from equation (2) that the key component of LNL is the NTM $\mathbf{T}(\mathbf{x})$. Note that $\mathbf{T}(\mathbf{x})$ is generally unidentifiable without any assumption. Thus, this paper focuses on widely studied class-dependent and instance-independent label noise, i.e., $\mathbf{T}(\mathbf{x}) = \mathbf{T}$. The problem then lies in how to determine $\mathbf{T}$ in a federated manner.

Federated Label-Noise Learning

In the CL setting, a state-of-the-art LNL scheme called VolMinNet (Li et al. 2021) proposes the following loss function that can learn the NTM and the classifier simultaneously:

$$\min_{\mathbf{w}, \tilde{\mathbf{T}}} \frac{1}{N} \sum_{i=1}^{N} L(\tilde{y}_i, \tilde{\mathbf{T}}^\top f(\mathbf{x}_i; \mathbf{w})) + \lambda \log\det(\tilde{\mathbf{T}}), \quad (3)$$

where $\tilde{\mathbf{T}}$ denotes the estimate of $\mathbf{T}$, $\det(\tilde{\mathbf{T}})$ denotes the determinant of $\tilde{\mathbf{T}}$, and λ is a balancing hyperparameter. In VolMinNet, the estimate $\tilde{\mathbf{T}}$ is modeled by a trainable matrix with K^2 unconstrained parameters. A softmax operation is then applied to each row of $\tilde{\mathbf{T}}$. Based on VolMinNet, an intuitive idea to accomplish FedLNL is to solve the optimization problem in equation (3) via an FL algorithm (e.g., FedAvg (McMahan et al. 2017)). Specifically, each device m adopts VolMinNet (Li et al. 2021) to train a local NTM and a local classifier via a gradient-based optimization algorithm over its local dataset. These local NTMs and local classifiers are then consolidated into a global NTM and a global classifier in the central server.

However, such an intuitive approach can obtain a global NTM with a high estimation error, especially when each client's device only has limited local training data. As the parameters of a local NTM are unconstrained, the gradients for the local NTM can be too large due to limited local training samples, causing the local NTM to converge to a solution with a high estimation error. To tackle this issue, an alternating update method is developed in FedLNL to estimate the local NTM and train the local classifier.

Alternating Update Method for Local NTM and Local Classifier

A Bayesian statistical modeling is adopted in FedLNL for the estimate $\tilde{\mathbf{T}}$. That is, the k-th row of $\tilde{\mathbf{T}}$ follows a Dirichlet distribution, i.e., $\tilde{\mathbf{T}}_k \sim \text{Dir}(\mathbf{d}_k), k = 1, ..., K$, where $\mathbf{d}_k \in \mathbb{R}^K$ denotes the concentration parameters of the Dirichlet distribution. These concentration parameters consist of a concentration matrix $\mathbf{D} = [\mathbf{d}_1^\top, ..., \mathbf{d}_K^\top]^\top$. To obtain an accurate estimate of $\mathbf{T}$, $\mathbf{D}$ needs to be updated such that $\mathbf{T}$ can be sampled from the K Dirichlet distributions with a high probability.

According to Bayesian inference, $\mathbf{D}$ can be updated by determining the posterior distribution of $\tilde{\mathbf{T}}$. Given N noisy training data samples (denoted as $(\mathbf{X}, \tilde{\mathbf{y}})$), the probability density function (PDF) of the posterior distribution can be written as

$$\begin{aligned} P(\tilde{\mathbf{T}}|\mathbf{X}, \tilde{\mathbf{y}}; \mathbf{D}) &= \frac{P(\tilde{\mathbf{T}}; \mathbf{D}) P(\tilde{\mathbf{y}}|\mathbf{X}, \tilde{\mathbf{T}})}{P(\tilde{\mathbf{y}}|\mathbf{X})} \\ &\propto P(\tilde{\mathbf{T}}; \mathbf{D}) P(\tilde{\mathbf{y}}|\mathbf{X}, \tilde{\mathbf{T}}), \end{aligned} \tag{4}$$

where $P(\tilde{\mathbf{T}}; \mathbf{D})$ and $P(\tilde{\mathbf{y}}|\mathbf{X}, \tilde{\mathbf{T}})$ represent the PDF of the prior distribution of $\tilde{\mathbf{T}}$ and the likelihood of $\tilde{\mathbf{T}}$, respectively. $P(\tilde{\mathbf{T}}; \mathbf{D})$ is determined by the PDFs of the K Dirichlet distributions, i.e.,

$$P(\tilde{\mathbf{T}}; \mathbf{D}) = \prod_{k=1}^{K} P(\tilde{\mathbf{T}}_k; \mathbf{d}_k), \tag{5}$$

where $P(\tilde{\mathbf{T}}_k; \mathbf{d}_k)$ is the PDF of $\text{Dir}(\mathbf{d}_k)$. As for $P(\tilde{\mathbf{y}}|\mathbf{X}, \tilde{\mathbf{T}})$, if the clean labels $\mathbf{y}$ are known, it can be expanded as

$$\begin{aligned} P(\tilde{\mathbf{y}}|\mathbf{X}, \tilde{\mathbf{T}}) &= \prod_{n=1}^{N} P(\tilde{y}_n|\mathbf{x}_n, y_n, \tilde{\mathbf{T}}) \\ &= \prod_{n=1}^{N} \tilde{T}_{y_n, \tilde{y}_n}. \end{aligned} \tag{6}$$

Let $\mathbf{C}$ denote a confusion matrix. Each item $C_{i,j}$ of $\mathbf{C}$ denotes the number of data samples whose true label and noisy label are $y = i$ and $\tilde{y} = j$, respectively. The likelihood can then be rewritten as

$$P(\tilde{\mathbf{y}}|\mathbf{X}, \tilde{\mathbf{T}}) = \prod_{i=1}^{K} \prod_{j=1}^{K} (\tilde{T}_{i,j})^{C_{i,j}}. \tag{7}$$

According to the principle of Dirichlet-categorical models (Diaconis and Ylvisaker 1979), $P(\tilde{\mathbf{T}}|\mathbf{X}, \tilde{\mathbf{y}}; \mathbf{D})$ determined via equation (4) belongs to the same probability distribution family as $P(\tilde{\mathbf{T}}; \mathbf{D})$. The concentration matrix of $P(\tilde{\mathbf{T}}|\mathbf{X}, \tilde{\mathbf{y}}; \mathbf{D})$, denoted as $\mathbf{D}_e$, is determined by $\mathbf{D}_e = \mathbf{D} + \mathbf{C}$. $\mathbf{D}_e$ is also regarded as the updated value of $\mathbf{D}$, i.e., the update rule of $\mathbf{D}$ is

$$\mathbf{D} \leftarrow \mathbf{D} + \mathbf{C}. \tag{8}$$

Note that $P(\tilde{\mathbf{y}}|\mathbf{X}, \tilde{\mathbf{T}})$ can be expanded with respect to the local training data on edge devices, i.e., $P(\tilde{\mathbf{y}}|\mathbf{X}, \tilde{\mathbf{T}}) = \prod_{m=1}^{M} P(\tilde{\mathbf{y}}_m|\mathbf{X}_m, \tilde{\mathbf{T}})$, where $(\mathbf{X}_m, \tilde{\mathbf{y}}_m)$ represents the local training data on edge device m. Let $\mathbf{C}_m$ denote the confusion matrix of $(\mathbf{X}_m, \tilde{\mathbf{y}}_m)$, then the update rule of $\mathbf{D}$ can be rewritten as

$$\mathbf{D} \leftarrow \mathbf{D} + \sum_{m=1}^{M} \mathbf{C}_m. \tag{9}$$

Hence, in FedLNL, device m needs to determine $\mathbf{C}_m$, so as to update $\mathbf{D}$ in a federated manner.

In practice, the clean labels $\mathbf{y}_m$ are unknown so that $\mathbf{C}_m$ cannot be directly obtained. To this end, FedLNL leverages the local classifier to estimate $\mathbf{C}_m$. More specifically, in the step of updating the local concentration matrix, the local classifier is performed over the local training samples to generate the softmax outputs. An estimate $\hat{\mathbf{y}}_m$ of $\mathbf{y}_m$ is then sampled from the softmax outputs. If the local classifier is well-trained, the clean labels $\mathbf{y}_m$ can be sampled with a high probability from the softmax outputs. Based on $\hat{\mathbf{y}}_m$ and $\tilde{\mathbf{y}}_m$, the items in the local confusion matrix $\mathbf{C}_m$ can be determined. Afterwards, the local concentration matrix is updated by

$$\mathbf{D}_m \leftarrow \alpha_1 \mathbf{D}_m + \alpha_2 \mathbf{C}_m, \tag{10}$$

where α_1 and α_2 are the weights for $\mathbf{D}_m$ and $\mathbf{C}_m$, respectively. In equation (10), the weights are used to control the step of the update since both $\mathbf{C}_m$ and $\mathbf{D}_m$ are not accurate initially. The global concentration matrix is then updated via

$$\mathbf{D} \leftarrow \sum_{m=1}^{M} \frac{N_m}{N} \mathbf{D}_m. \tag{11}$$

In the step of updating the local classifier, a local NTM $\tilde{\mathbf{T}}_m$ is sampled from $P(\tilde{\mathbf{T}}_m; \mathbf{D}_m)$. The local classifier is then trained by solving a local optimization problem that is designed in the next subsection.

Local Diversity Product Regularizer

As the NTM in FedLNL is sampled from K Dirichlet distributions, the loss function of VolMinNet (Li et al. 2021) is no longer suitable for FedLNL. The reason is that the regularizer $\log\det(\tilde{\mathbf{T}})$ is effective only when the NTM is trainable. To design a loss function for FedLNL, a regularizer on the parameters of the classifier is first derived as follows. Let $\tilde{\mathbf{P}} \in \mathbb{R}^{K \times N}$ denote the matrix of the noisy class posterior of N data samples and $\mathbf{P} = [f(\mathbf{x}_1; \mathbf{w}), ..., f(\mathbf{x}_N; \mathbf{w})] \in \mathbb{R}^{K \times N}$ denote the matrix of the softmax outputs of a classifier. Given a noisy dataset (i.e., $\tilde{\mathbf{P}}$ is determined), the equation $\tilde{\mathbf{P}} = \tilde{\mathbf{T}}^\top \mathbf{P}$ has infinite solutions $(\tilde{\mathbf{T}}, \mathbf{P})$. Let $\mathcal{T}$ denote the set of all possible estimates of $\mathbf{T}$ and $\mathcal{W}$ denote the set of the corresponding $\mathbf{w}$. Among these estimates $\tilde{\mathbf{T}} \in \mathcal{T}$, it is proved in (Li et al. 2021) that the ground-truth NTM $\mathbf{T}$ has the minimal volume, i.e.,

$$\mathbf{T} = \arg\min_{\tilde{\mathbf{T}} \in \mathcal{T}} \det(\tilde{\mathbf{T}}).$$

By rewriting $\tilde{\mathbf{P}} = \tilde{\mathbf{T}}^\top \mathbf{P}$ in the form of determinant

$$\det(\tilde{\mathbf{P}} \tilde{\mathbf{P}}^\top) = (\det(\tilde{\mathbf{T}}))^2 \det(\mathbf{P}\mathbf{P}^\top),$$

it can be seen that minimizing $\det(\tilde{\mathbf{T}})$ is equivalent to maximizing $\det(\mathbf{P}\mathbf{P}^\top)$ if $\tilde{\mathbf{P}}$ is fixed. As a result, the new regularizer $\log\det(\mathbf{P}\mathbf{P}^\top + \mathbf{I})$ is obtained and the global optimization problem of FedLNL can be written as

$$\min_{\mathbf{w}} \left\{ \frac{1}{N}\sum_{i=1}^{N} L(\tilde{y}_i, \tilde{\mathbf{T}}^\top f(\mathbf{x}_i;\mathbf{w})) - \lambda \log\det(\mathbf{P}\mathbf{P}^\top + \mathbf{I}) \right\}, \tag{12}$$

where $\mathbf{I} \in \mathbb{R}^{K\times K}$ is a identity matrix. The form $\det(\mathbf{P}\mathbf{P}^\top)$ is actually a diversity-promoting regularizer (Malkin and Bilmes 2008) on the softmax outputs of the classifier.

However, it is challenging to solve the optimization problem in equation (12) in a federated manner, because the diversity-promoting regularizer couples the softmax outputs of all the training data. To this end, a decomposable regularizer called local diversity product (LDP) regularizer is developed as follows.

Note that $\mathbf{P} = [\mathbf{P}_1, ..., \mathbf{P}_M]$, so $\mathbf{P}\mathbf{P}^\top$ can be decomposed as

$$\mathbf{P}\mathbf{P}^\top = \sum_{m=1}^{M} \mathbf{P}_m \mathbf{P}_m^\top, \tag{13}$$

where $\mathbf{P}_m \in \mathbb{R}^{K\times N_m}$ represents the matrix of the softmax outputs of the local classifier on edge device m. Inspired by equation (13), the following theorem is leveraged to guide the design of LDP regularizer.

Theorem 1. *Let* $(\mathbf{w}^*, \mathbf{T})$ *denote the optimal solution to the optimization problem in equation* (12) *and* $\mathbf{P}^* = [f(\mathbf{x}_1;\mathbf{w}^*), ..., f(\mathbf{x}_N;\mathbf{w}^*)]$ *denote the matrix of the softmax outputs of the optimal classifier.* $\mathbf{P}^*$ *not only maximizes* $\det(\mathbf{P}\mathbf{P}^\top)$*, i.e.,* $\mathbf{P}^* = \arg\max_{\mathbf{w}\in\mathcal{W}} \det(\mathbf{P}\mathbf{P}^\top)$ *but also satisfies*

$$\mathbf{P}^* = \arg\max_{\mathbf{w}\in\mathcal{W}} \prod_{m=1}^{M} \det(\mathbf{P}_m\mathbf{P}_m^\top + \mathbf{I}). \tag{14}$$

According to Theorem (1), the diversity-promoting regularizer $\log\det(\mathbf{P}\mathbf{P}^\top + \mathbf{I})$ in equation (12) can be replaced by LDP regularizer $\sum_{m=1}^{M}\log\det(\mathbf{P}_m\mathbf{P}_m^\top + \mathbf{I})$ without changing the optimal solution.

The global optimization problem in (12) can then be reformulated as

$$\min_{\mathbf{w}} \frac{1}{N}\sum_{i=1}^{N} L(\tilde{y}_i, \tilde{\mathbf{T}}^\top f(\mathbf{x}_i;\mathbf{w})) - \lambda \sum_{m=1}^{M} \log\det(\mathbf{P}_m\mathbf{P}_m^\top + \mathbf{I}). \tag{15}$$

The global optimization problem in (15) can be easily decomposed with respect to the local datasets of the edge devices. For each device m, its local optimization problem can be written as

$$\begin{aligned}\min_{\mathbf{w}_m} &\frac{1}{N_m}\sum_{i\in\mathcal{L}_m} L(\tilde{y}_i, \tilde{\mathbf{T}}_m^\top f(\mathbf{x}_i;\mathbf{w}_m)) \\ &- \frac{\lambda N}{N_m}\log\det(\mathbf{P}_m\mathbf{P}_m^\top + \mathbf{I}),\end{aligned} \tag{16}$$

where $\mathbf{w}_m$ denotes the parameters of the local classifier and $\tilde{\mathbf{T}}_m$ denotes the local NTM. In the step of updating the local classifier, the local optimization problem in (16) is then solved to update the local classifier.

Dataset	CIFAR-10	CIFAR-100	Clothing1M
# samples (train)	50,000	50,000	1,000,000
# samples (test)	10,000	10,000	10,526
# classes	10	100	14
# clients	100	50	500
Selection ratio	0.1	0.2	0.02
Architecture	ResNet-18	ResNet-34	Pre-trained ResNet-50

Table 2: Datasets used in the experiments.

Experiments

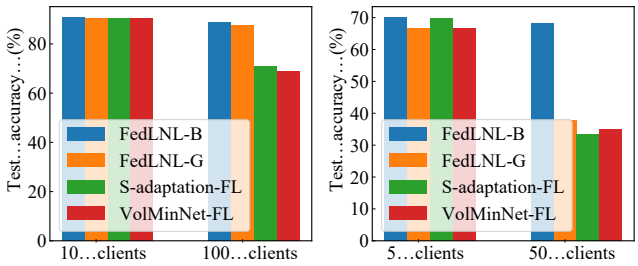
In this section, the effectiveness of the alternating update method and LDP regularizer is first verified in an ablation study. The overall performance of FedLNL is then evaluated by comparing it with the state-of-the-art FL schemes for label-noise.

Experimental Setup

Datasets With Label-Noise Three datasets are adopted in the experiments: **CIFAR-10** (Krizhevsky, Hinton et al. 2009), **CIFAR-100** (Krizhevsky, Hinton et al. 2009), and **Clothing1M** (Xiao et al. 2015)). The statistical information of each dataset is provided in Table 2. The statistical information of each dataset is provided in Table 2. Since **CIFAR-10** and **CIFAR-100** are originally clean datasets, their labels are manually corrupted with three types of synthesis label noise: pair flipping (denoted as flip)(Han et al. 2018), symmetry (denoted as sym) (Patrini et al. 2017), and asymmetry (denoted as asym) (Tanaka et al. 2018). The noise rate of pair flipping noise and symmetry noise is selected from $\{0.2, 0.4, 0.45\}$ and $\{0.2, 0.4, 0.45\}$, respectively, while the noise rate of asymmetry noise is set to 0.4. As for **Clothing1M** dataset, it contains real-world label noise whose noise rate is near 0.4.

Both i.i.d. data partition (**CIFAR-10**, **CIFAR-100**, and **Clothing1M**) and non-i.i.d. data partition (**CIFAR-10** and **Clothing1M**) are considered in the experiments. For **CIFAR-10**, the pathological non-i.i.d. partition (McMahan et al. 2017) is used and the number of classes held by each client is selected from $\{5, 6, 8\}$. For **Clothing1M**, the non-i.i.d. partition adopted in (Xu et al. 2022) is utilized.

Compared Schemes FedLNL is compared with the following schemes: DivideMix (Li, Socher, and Hoi 2020), S-adaptation (Goldberger and Ben-Reuven 2017), VolMinNet (Li et al. 2021), FedAvg (McMahan et al. 2017), RoFL (Yang et al. 2022b), FedLSR (Jiang et al. 2022), and FedCorr (Xu et al. 2022). Among these schemes, DivideMix, S-adaptation, and VolMinNet are centralized schemes combating label noise, and they are extended to the FL framework. RoFL, FedLSR, and FedCorr represent the state-of-the-art FL schemes tackling label-noise issues. FedAvg is the simplest FL algorithm that does not consider label noise.



(a) CIFAR-10 (flip-04) (b) CIFAR-100 (flip-04)

Figure 1: Test accuracies of the selected schemes under two settings of local training samples.

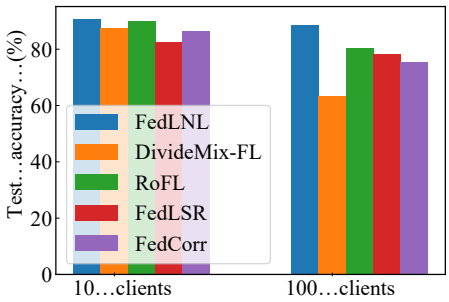


Figure 2: Test accuracies of FedLNL and the compared schemes over CIFAR-10 (flip-0.4) under two settings of local training samples.

Implementation Details For each dataset, the number of clients, the selection ratio of clients in each iteration, and the architecture of the classifier are given in Table 2. Note that the number of clients is set to 10 and 5 to simulate the setting of enough local training samples for **CIFAR-10** and **CIFAR-100**, respectively. The number of clients is set to 100, 50, and 500 to simulate the setting of limited local training samples for **CIFAR-10**, **CIFAR-100**, and **Clothing1M**, respectively. In FedLNL, each client trains its local classifier via stochastic gradient descent (SGD) with a momentum of 0.9. The learning rate is set to 0.01. The batch size is set to 64. The number of local iterations is set to 3 and the total number of communication rounds between the clients' devices and the central server is set to 300. Hyperparameter λ is set to 0.01. Each experiment is repeated 5 times and average test accuracy is recorded. All the experiments are executed on a server with one i9-10900k CPU, one GeForce RTX 3090 GPU, and 64 GB RAM. The implementation of FedLNL is based on Pytorch-1.7.0 (Paszke et al. 2019).

Experimental Results

The effectiveness of LDP regularizer is verified by comparing FedLNL with two existing LNL schemes that are extended to the FL framework (i.e., S-adaptation (Goldberger and Ben-Reuven 2017), VolMinNet (Li et al. 2021)). Since both S-adaptation and VolMinNet utilize a gradient-based update method to update local NTMs, FedLNL in this comparison needs to adopt the same update method (denoted as FedLNL-G) for a fair comparison. Afterwards, FedLNL employing the alternating update method (denoted as FedLNL-B) is compared with FedLNL-G to show the effectiveness of FedDU.

As shown in Figure 1(a), FedLNL-G maintains a high test accuracy in both the setting of sufficient local training samples (i.e., the case of 10 clients) and that of limited local training samples (i.e., the case of 10 clients). As for S-adaptation and VolMinNet, they perform well when the clients' devices have sufficient local training data. However, in the case of limited local training samples, the test accuracy of S-adaptation and that of VolMinNet is reduced by 19.16% and 21.30%, respectively. These results indicate that DP regularizer effectively mitigates the overfitting of local NTMs caused by limited local training samples.

Schemes	flip-0.2	flip-0.4	flip-0.45
FedAvg (clean)	71.95	71.95	71.95
FedAvg	50.63	38.34	32.46
S-adaptation-FL	52.38	34.82	33.32
VolMinNet-FL	55.49	34.85	32.58
DivideMix-FL	57.88	42.89	36.77
RoFL	55.93	44.22	39.75
FedLSR	57.60	37.70	30.34
FedCorr	63.72	48.69	40.14
FedLNL	69.14	68.14	66.12

Table 3: Test accuracies (%) of different schemes on CIFAR-100 dataset.

In the experiment over **CIFAR-10**, the advantage of the alternating update method is obvious since the two versions of FedLNL achieve comparable performance. In the experiment over **CIFAR-100**, the advantage of FedLNL-B becomes much more obvious. As shown in Figure 1(b), the test accuracy of FedLNL-G degrades dramatically in the case of limited local training samples, while FedLNL-B maintains a high test accuracy.

The reason that FedLNL-B achieves such a large gain on **CIFAR-100** is as follows. In the training data of **CIFAR-100**, each class only has 500 images. After the partition of the training data, the number of each class of data held by each client's device becomes much smaller. If a gradient-based update method is employed to update the local NTM, the local NTM suffers overfitting due to limited local training data. Consequently, the central server can only obtain a global NTM with a high estimation error, which further degrades the performance of the global classifier.

The overall performance of FedLNL is then evaluated over **CIFAR-10** and **CIFAR-100** with synthesis noise. In the setting of i.i.d. data partition and limited local training samples, the test accuracies of FedLNL and the compared schemes on **CIFAR-10** and **CIFAR-100** are shown in Table 4 and Table 3, respectively. For **CIFAR-10**, the performance of the compared schemes under the setting of sufficient local training samples is shown in Figure 2.

Schemes	flip-0.2	flip-0.4	flip-0.45	sym-0.2	sym-0.4	sym-0.5	asym-0.4
FedAvg (clean)	91.24	91.24	91.24	91.24	91.24	91.24	91.24
FedAvg	85.31	60.34	53.25	82.16	61.87	48.37	78.87
DivideMix-FL	75.11	63.40	54.90	76.19	71.82	65.52	70.05
S-adaptation-FL	88.65	71.10	62.96	87.56	82.89	75.75	80.69
VolMinNet-FL	88.37	69.06	53.25	87.63	83.20	74.79	80.72
RoFL	86.32	80.44	72.74	86.25	83.07	73.88	72.04
FedLSR	86.17	78.29	68.89	86.98	83.48	80.87	79.23
FedCorr	88.37	75.31	57.87	87.32	79.23	72.50	77.80
FedLNL	89.20	88.63	88.40	87.76	84.19	81.19	88.01

Table 4: Test accuracies (%) of FedLNL and the compared schemes over CIFAR-10 dataset.

Schemes	8 classes	6 classes	5 classes
FedAvg (clean)	89.60	88.78	86.60
FedAvg	54.63	50.38	48.79
S-adaptation-FL	68.40	74.57	74.34
VolMinNet-FL	70.28	77.22	77.80
DivideMix-FL	56.84	55.06	49.80
RoFL	79.19	67.77	59.85
FedLSR	75.54	74.08	70.62
FedCorr	67.20	61.19	58.48
FedLNL	88.02	87.53	85.47

Table 5: Test accuracy (%) on CIFAR-10 with flip-0.4 noise under different settings of non-i.i.d. data.

Schemes	i.i.d.	Non-i.i.d.
FedAvg	69.93	67.12
S-adaptation-FL	68.92	65.45
VolMinNet-FL	68.44	65.49
DivideMix-FL	68.01	67.97
RoFL	69.00	68.92
FedLSR	56.23	64.01
FedCorr	69.94	68.96
FedLNL	73.36	72.95

Table 6: Test accuracies (%) of different schemes on Clothing1M dataset.

The results in Figure 2 show that some CL and FL schemes (e.g., DivideMix, RoFL, and FedCorr) are effective when the local training samples are sufficient, while they are vulnerable to the limited local training samples. The results of Table 4 show that FedLNL achieves the highest test accuracies among all the settings of label noise. In the settings of flip-0.45 and asym-0.4, FedLNL improves the test accuracy by 19.51% and 8.78%, compared with the second best scheme FedLSR (Jiang et al. 2022). Moreover, in the settings of pair flipping noise and asymmetry noise, the test accuracies of FedLNL are close to that of FedAvg over the clean data (denoted as FedAvg (clean)), indicating that FedLNL nearly achieves the upper bound of its test accuracy. For **CIFAR-100**, FedLNL improves the test accuracy by up to 25.98% compared with the second best scheme FedCorr (Xu et al. 2022).

The performance of FedLNL is also evaluated on **CIFAR-10** under the settings of non-i.i.d. data. The noise type is pair flipping and the noise rate is set to 0.4. As shown in Table 5, FedLNL still achieves the highest test accuracy even under the settings of non-i.i.d. data, and the test accuracy is improved by up to 17.74%.

In addition to **CIFAR-10** and **CIFAR-100**, FedLNL is also applied to **Clothing1M** dataset with real-world label-noise, and the corresponding results are shown in Table 6. It can be seen that FedLNL outperforms the state-of-the-art FL scheme FedCorr (Xu et al. 2022) under both the settings of i.i.d. data partition and non-i.i.d. data partition.

Conclusion

In this paper, an FL scheme called federated label-noise learning (FedLNL) based on a noise transition matrix (NTM) was developed to tackle the label-noise issue, especially when there were only limited local training samples. In FedLNL, in order to estimate the NTM accurately, each client's device updated its local NTM and local classifier alternatingly, where a Bayesian inference-based update method was designed to update the prior distribution of the local NTM. To enable federated optimization for FedLNL, a decomposable regularizer called local diversity product (LDP) regularizer was designed for the loss function of FedLNL. Extensive experiments were conducted to verify the alternating update method and LDP regularizer and to evaluate the overall performance of FedLNL. The experimental results showed that FedLNL outperformed the state-of-the-art FL schemes that tackle label-noise issues.

References

Chen, P.; Liao, B. B.; Chen, G.; and Zhang, S. 2019. Understanding and utilizing deep neural networks trained with noisy labels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, 1062–1070.

Chen, Y.; Yang, X.; Qin, X.; Yu, H.; Chan, P.; and Shen, Z. 2020. Dealing with label quality disparity in federated learning. *Federated Learning: Privacy and Incentive*, 108–121.

Diaconis, P.; and Ylvisaker, D. 1979. Conjugate Priors for Exponential Families. *The Annals of Statistics*, 7(2): 269–281.

Goldberger, J.; and Ben-Reuven, E. 2017. Training deep neural-networks using a noise adaptation layer. In *Proceedings of the International Conference on Learning Representations*.

Guo, J.; Gong, M.; Liu, T.; Zhang, K.; and Tao, D. 2020. LTF: A Label Transformation Framework for Correcting Label Shift. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, 3843–3853.

Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.; and Sugiyama, M. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, volume 31.

Huang, L.; Zhang, C.; and Zhang, H. 2020. Self-Adaptive Training: beyond Empirical Risk Minimization. In *Advances in Neural Information Processing Systems*, volume 33.

Jiang, L.; Zhou, Z.; Leung, T.; Li, L.-J.; and Fei-Fei, L. 2018. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, 2304–2313.

Jiang, X.; Sun, S.; Wang, Y.; and Liu, M. 2022. Towards federated learning against noisy labels via local self-regularization. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 862–873.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Kun, Y.; and Jianxin, W. 2019. Probabilistic End-to-end Noise Correction for Learning with Noisy Labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Kye, S. M.; Choi, K.; Yi, J.; and Chang, B. 2022. Learning with Noisy Labels by Efficient Transition Matrix Estimation to Combat Label Miscorrection. In *Proceedings of the European Conference on Computer Vision*, 717–738.

Li, J.; Socher, R.; and Hoi, S. C. 2020. DivideMix: Learning with Noisy Labels as Semi-supervised Learning. In *Proceedings of the International Conference on Learning Representations*.

Li, X.; Liu, T.; Han, B.; Niu, G.; and Sugiyama, M. 2021. Provably End-to-end Label-noise Learning without Anchor Points. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, 6403–6413.

Malkin, J.; and Bilmes, J. 2008. Ratio semi-definite classifiers. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 4113–4116.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 1273–1282.

Mirzasoleiman, B.; Cao, K.; and Leskovec, J. 2020. Coresets for robust training of deep neural networks against noisy labels. In *Advances in Neural Information Processing Systems*, volume 33, 11465–11477.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, volume 32.

Patrini, G.; Rozza, A.; Krishna Menon, A.; Nock, R.; and Qu, L. 2017. Making Deep Neural Networks Robust to Label Noise: A Loss Correction Approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2233–2241.

Tanaka, D.; Ikami, D.; Yamasaki, T.; and Aizawa, K. 2018. Joint Optimization Framework for Learning with Noisy Labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Tsouvalas, V.; Saeed, A.; Ozcelebi, T.; and Meratnia, N. 2022. Federated Learning with Noisy Labels. *arXiv preprint arXiv:2208.09378*.

Wang, Z.; Zhou, T.; Long, G.; Han, B.; and Jiang, J. 2022. FedNOiL: a simple two-level sampling method for federated learning with noisy labels. *arXiv preprint arXiv:2205.10110*.

Xia, X.; Liu, T.; Wang, N.; Han, B.; Gong, C.; Niu, G.; and Sugiyama, M. 2019. Are anchor points really indispensable in label-noise learning? In *Advances in Neural Information Processing Systems*, volume 32, 6838–6849.

Xiao, T.; Xia, T.; Yang, Y.; Huang, C.; and Wang, X. 2015. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2691–2699.

Xu, J.; Chen, Z.; Quek, T. Q.; and Chong, K. F. E. 2022. FedCorr: Multi-Stage Federated Learning for Label Noise Correction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 10184–10193.

Yang, M.; Qian, H.; Wang, X.; Zhou, Y.; and Zhu, H. 2022a. Client Selection for Federated Learning With Label Noise. *IEEE Transactions on Vehicular Technology*, 71(2): 2193–2197.

Yang, S.; Park, H.; Byun, J.; and Kim, C. 2022b. Robust Federated Learning With Noisy Labels. *IEEE Intelligent Systems*, 37(2): 35–43.

Zhang, Y.; Niu, G.; and Sugiyama, M. 2021. Learning noise transition matrix from only noisy labels via total variation regularization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, 12501–12512.

Zheng, G.; Awadallah, A. H.; and Dumais, S. 2021. Meta Label Correction for Noisy Label Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35.

Zheng, S.; Wu, P.; Goswami, A.; Goswami, M.; Metaxas, D.; and Chen, C. 2020. Error-bounded correction of noisy labels. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, 11447–11457.