When Heterogeneous Federated Learning Meets Noisy Label

Yijie Liu$^{*\mathrm{AI}}$, **Shu Chen**$^{*\mathrm{AI}}$, **Yizhou Chen**$^{*\mathrm{AI}}$, **Hezhao Liu**$^{*\mathrm{Info}}$, **Shanshan Yan**$^{*\mathrm{AI}}$,

Xiamen University, Xiamen 361005, China
{23020231154211, 23020231154170, 36920231153184, 23020231154209, 36920231153252}@stu.xmu.edu.cn

Abstract

Federated learning has gained popularity for distributed learning without aggregating sensitive data from clients. But the distributed and isolated nature of data isolation may be complicated by data quality, making it more vulnerable to noisy labels. Meanwhile, each client may independently design its own model based on its hardware conditions. We attempt to study a challenging and trustworthy federated learning framework in the next to months during the Deep Learning course project to simultaneously handle label-noise and model-heterogeneity. (1) For the aggregation between heterogeneous models, we plan to align the models feedback by utilizing public data (such as CIFAR-100), which does not require additional shared global models for collaboration. (2) To tackle internal noise, we design a noise-resistant loss which combines CE loss and RCE loss to reduce the impact. (3) To tackle the noise from other participants, propose a new weighting approach to reduce the impact from noisy clients during federated communication. We conduct experiments on CIFAR-10 and CIFAR-100, and the results show that our method reduces the impact of noise and improves classification accuracy in the setting of model-heterogeneous federated learning.

Introduction

Local clients such as mobile devices or whole organizations generally have limited private data and limited generalizability. However, due to the existence of data silos and data privacy, we cannot use traditional centralized learning in practical applications (Kairouz et al. 2021). To address these challenges, Federated Learning (FL) has been proposed by McMahan et al. (McMahan et al. 2017). Federated learning is a distributed machine learning framework that enables multiple clients to collaboratively train models with decentralized data. The clients never share private data with server ensuring basic privacy. Recently, the widely used federated learning algorithms, e.g., FedAvg (McMahan et al. 2017) and FedProx (Li et al. 2020), are based on averaging the model parameters of the participating clients. Most of these federated learning methods are developed based on the assumption that participating client models have the same neural architecture.

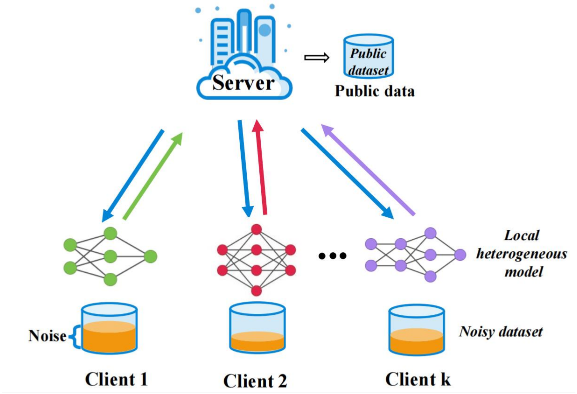

Figure 1: Illustration of federated learning with noisy and heterogeneous clients, where clients have different local models and noisy datasets with different noise.

However, In real-world scenarios, due to the differences in the personalized requirements, each client might expect to design its own model independently (Shen et al. 2020; Sun and Lyu 2020; He, Annavaram, and Avestimehr 2020), resulting in the model heterogeneous federated learning problem, as illustrated in Fig.1. Therefore, to perform federated learning with heterogeneous models, a number of heterogeneous federated learning methods have been proposed (Li and Wang 2019; Lin et al. 2020; Liang et al. 2020). FedMD (Li and Wang 2019) is a framework based on knowledge distillation, which is implemented through the class scores by client models on the public dataset. FedDF (Lin et al. 2020) leverage unlabeled data to perform ensemble distillation for each different model architecture. These strategies mainly rely on a unified global consensus or shared models. However, learning a global consensus has a major limitation in that the clients cannot individually adjust their learning direction to accommodate the differences among clients. Therefore, how to perform federated learning with heterogeneous clients without relying on a global consensus or shared models is challenging.

In addition, when the clients contain inevitable noisy samples, existing federated learning methods cannot eliminate the negative effect caused by label noise, suffering from a significant performance drop (Tam et al. 2021). Generally,

[*] These authors contributed equally.

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

in practical applications, the label noise is caused by the following two aspects: 1) Due to the limitation and scarcity of human expertise, the quality of labeled data will be affected by human subjective factors, thus inevitably results in some wrong annotation. 2) In the federated learning framework, considering the user fairness issue, there may be some participants in the system who want to learn from the global model, but do not want to provide useful information. Therefore, some users are reluctant to share their real information with other users and deliberately generate some wrong labels. Under the federated learning framework, we expect that each class of samples will be learned sufficiently while avoiding overfitting to noisy samples. Therefore, how to reduce the negative impact of the internal label noise on the local model convergence during the local update phase is an important issue.

Furthermore, the above mentioned two problems lead to a new issue, i.e., how to reduce the negative and noisy influence from other clients while collaborative learning in the federated learning framework. Due to model heterogeneity, the participating clients will have different decision boundaries and varying noisy patterns. As a result, besides local noise, we also need to pay attention to the noise from other clients, and then it is crucial to reduce the contribution of noisy clients in the whole federated system.

In this report, we provide the framework for the problem with noisy and heterogeneous clients: 1) Utilizing the public dataset CIFAR-10 to align the output distributions in heterogeneous model architecture. We update the models by the feedback on public dataset rather than upload local models to handle the model-heterogeneous problem. 2) Introducing Reverse cross-entropy (RCE) loss combinating with cross-entropy (CE) loss to reduce overfitting to the noise. 3) Proposing a new weighting approach to reduce the impact from noisy clients during federated communation. We first quantifies the label quality among clients, and then align the client weight adaptively.

Related work

Federated Learning

The concept of federated learning was first proposed in 2017 by McMahan et al. (McMahan et al. 2017). It is a machine learning setting that allows clients to collaboratively train models while protecting data privacy. McMahan et al. propose FedAvg, in which the client uses private data to reduce the local gradient of the local model, and the server uses the averaged model parameters to aggregate the local model. Li et al. (Li et al. 2020) build a framework similar to FedAvg, but it can adaptively set the local calculations according to different devices and iterations. Wang et al. (Wang et al. 2020) propose to collect the weight of each layer of the client and performs one-layer matching to obtain the weight of each layer of the federated model.

For learning with model heterogeneous clients, Li et al. (Li and Wang 2019) implement communication between models through knowledge distillation. The server collects the class scores of the public data set on each client model and calculates the average value as the updated consensus. Lin et al. (Lin et al. 2020) leverage ensemble distillation for model fusion, and it can be carried out through unlabeled data. Diao et al. (Diao, Ding, and Tarokh 2020) propose to adaptively allocate a subset of global model parameters as local model parameters according to the corresponding capabilities of the local client. Liang et al. (Liang et al. 2020) introduce an algorithm to jointly train the compact local representation and global model of the client.

In summary, existing methods are usually developed under the assumption that all clients possess clean data without noise, dedicated to making federated learning more efficient, and preserving the privacy of user data.

Label Noise Learning

In machine learning, many methods have been proposed to handle label noise. They can be divided into four main categories:

- **Label transition matrix** (Sukhbaatar et al. 2014; Patrini et al. 2017; Yao et al. 2019). The main idea is to estimate the probability of each label class flipping to another class. Sukhbaatar et al. (Sukhbaatar et al. 2014) add a noise layer to the network to make the network output match the noisy label distribution. Patrini et al. (Patrini et al. 2017) design an end-to-end loss correction framework that makes recent noise estimation techniques applicable to the multi-class setting. Yao et al. (Yao et al. 2019) transform the noise into a Dirichlet-distributed space, use the dynamic label regression method iteratively infer the potential real labels, and jointly train the classifier and noise modeling.
- **Robust regularization** (Zhang et al. 2017; Arpit et al. 2017; Miyato et al. 2018; Laine and Aila 2016). Robust regularization can effectively prevent the model from overfitting to noisy labels. Zhang et al. (Zhang et al. 2017) propose Mixup, which trains the convex combination of pairs of samples and their labels to regularize the hybrid neural network. Arpit et al. (Arpit et al. 2017) demonstrate regularization can reduce the memory speed of noise without affecting the learning of real data. Miyato et al. (Miyato et al. 2018) introduce a regularization method based on virtual adversarial loss, and defined the adversarial direction without label information, which makes it suitable for label noise setting.
- **Robust loss function** (Van Rooyen, Menon, and Williamson 2015a; Ghosh, Kumar, and Sastry 2017). Some methods achieve robust learning by using noise-tolerant loss functions. Rooyen et al. (Van Rooyen, Menon, and Williamson 2015a) propose a convex classification calibration loss, which is robust on symmetric label noise. Ghosh et al. (Ghosh, Kumar, and Sastry 2017) analyze some loss functions that are widely used in deep learning and proved that MAE is robust to noise.
- **Selecting possibly clean samples** (Han et al. 2018a; Wei et al. 2020; Jiang et al. 2018). The methods select clean samples from the noisy training dataset for learning, or re-weighting for each sample. The core idea is to reduce the attention to noisy-labeled samples in each iteration for training. Han et al. (Han et al. 2018a)

propose Co-teaching, which trains two deep neural networks at the same time and selects data with potentially clean labels for cross-trains. Wei et al. (Wei et al. 2020) present JoCoR, which calculates the joint loss with Co-Regularization, and then select small loss samples to update network parameters. Jiang et al. (Jiang et al. 2018) introduce MentorNet, which provides a sample weighting scheme for StudentNet, and MentorNet learns a data-driven curriculum dynamically with StudentNet.

Previous methods for solving label noise are mainly under the centralized setting. However, in the federated setting, the server cannot directly access the private datasets of clients. In the model heterogeneous setting, different model architectures will lead to different noisy patterns.

Proposed Solution

Problem Definition. We consider K clients under the federated learning scenario.We define $\mathbb{C}$ as the collection of all clients, where $\mathbb{C} = K$ and the k-th client $c_k \in \mathbb{C}$ has a private dataset $D_k = \{(x_i^k, y_i^k)_{i=1}^{N_k}\}$ with $\|x^k\| = N_k$. In the model-heterogeneous scenario, each client c_k has the local model θ_k, and $f(x^k, \theta_k)$ denotes the logits output of x^k calculated by θ_k. The server cannot access the clients' private datasets, and it has a public dataset$D_0 = \{x_i^0\}_{i=1}^{N_0}$.

Heterogeneous Model Alignment

During the aggregating phase, since the clients have different networks, we cannot use FedAvg algorithm to get the global model. We use the public dataset D_0 to complete the communication betwween clients. In the $t_c \in T_c$ communication rounds, each active client c_k uses the local model $\theta_k^{t_c}$ to calculate the logits on the public dataset D_0. Then we can get the knowledge distribution $R_k^{t_c} = f(D_0, \theta_k^{t_c})$ on the client c_k. In this way, we use Kullback-Leibler (KL) divergence to measure the difference of probability distributions from other clients:

$$\mathcal{KL}(R_{k_1}^{t_c} \| R_{k_2}^{t_c}) = \sum R_{k_1}^{t_c} \log(\frac{R_{k_1}^{t_c}}{R_{k_2}^{t_c}}), \tag{1}$$

It's obvious that the greater the knowledge distribution difference between $R_{k_1}^{t_c}$ and $R_{k_2}^{t_c}$, the more c_{k_1} and c_{k_2} can learn from each other. Therefore, minimizing the KL difference between probability distribution $R_{k_1}^{t_c}$ and $R_{k_2}^{t_c}$ can be considered as a process in which c_{k_1} learns knowledge from c_{k_2}.

Then we define the aggregation loss as follow:

$$\mathcal{L}_{kl}^{k,t_c} = \sum_{k_0=1, k_0 \neq k}^{K} \mathcal{KL}(R_{k_1}^{t_c} \| R_{k_2}^{t_c}), \tag{2}$$

where k_0 denotes the clients other thanc_k. In this way, the clients can update their own model parameters from the knowledge distribution difference:

$$\theta_k^{t_c} \leftarrow \theta_k^{t_c-1} - \alpha \nabla_\theta(\frac{1}{K-1} \cdot \mathcal{L}_{kl}^{k,t_c-1}), \tag{3}$$

where α represents the learning rate.

Local Model Loss

To reduce the negative impact of local noise, we learn the Symmetric Cross Entropy (Wang et al. 2019). In the presence of label noise, CE loss shows several limitations. Due to the different levels of difficulties among classes, CE loss can not make all classes be sufficiently learned or correctly classify all categories. In order to fully converge the difficult-to-learn classes, more rounds of learning will be performed. At this time, the easy-to-learn classes will tend to overfitting the noisy labels, and the overall performance of the model will begin to decline. if we denote p and q as the label class distribution and the predicted distribution respectively, we can see that p might not be the true class distribution due to the presence of label noise, on the contrary, q reflects the true class distribution to a degree. Then the CE loss and RCE loss can be expressed as:

$$\mathcal{L}_{ce} = -\sum_{i=1}^{N} p(x_i) \log(q(x_i)), \tag{4}$$

$$\mathcal{L}_{rce} = -\sum_{i=1}^{N} q(x_i) \log(p(x_i)). \tag{5}$$

We can combine the CE loss and the RCE loss to fully learn the difficult-to-learn classes while preventing overfitting noisy labels on the easy-to-learn classes. Then we formulated the Combinational Learning(CL) loss as:

$$\mathcal{L}_{cl} = \lambda \mathcal{L}_{ce} + \mathcal{L}_{rce}, \tag{6}$$

where λ is the hyper-parameter controlling the proportion of two loss. With L_{cl}, we can expand local update as:

$$\theta_k^{t_l} \leftarrow \theta_k^{t_l-1} - \alpha \nabla_\theta(f(x^k, \theta^{t_l - 1_k}), \tilde{y}^k), \tag{7}$$

where $t_l \in T_l$ represents the t_l-th communication round, $\tilde{y}^k$ represents the noisy labels on k-th client.

Client Weighting Scheme

We propose the new Re-weighting scheme to reduce the adverse impact of label noise from other clients during the aggregation phase. This scheme can pay more attention to clients with clean datasets and efficient models while reduce the contribution of noisy clients. To estimate the label quality, we use CL loss to calculate the loss between the predictive output of the local model θ_k on the private noisy dataset $\tilde{D}_k$ and the given label $\tilde{y}^k$. A small CL loss $\mathcal{L}_{cl}(f(x^k, \theta_k), \tilde{y}^k)$ indicates that the predicted pseudo-label has a similar distribution to the given labels, which means that the private dataset $\tilde{D}_k$ of the client k has accurate labels. On the contrary, a large loss signifies that the distribution of predicted predicted pseudo-labels and the given labels are different, i.e., the private dataset $\tilde{D}_k$ of the client c_k might possess many noisy labels. In this way, the label quality of the dataset $\tilde{D}_k$ can be formulated as:

$$\mathcal{Q}_{t_c}(\tilde{D}_k) = \frac{1}{\frac{1}{N_k} \sum_{i=1}^{N_k} \mathcal{L}_{cl}^{k,t_c}(f(x_i^k, \theta_k), \tilde{y}_i^k)}. \tag{8}$$

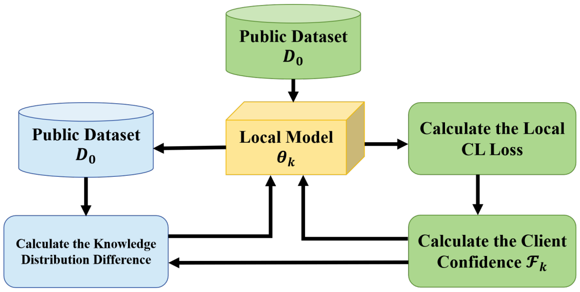
Public Dataset D_0
Public Dataset D_0
Local Model θ_k
Calculate the Local CL Loss
Calculate the Knowledge Distribution Difference
Calculate the Client Confidence $\mathcal{F}_k$

Figure 2: The pipeline of our method.

At the same time, the CL drop rate of the client c_k in the T_c round can reflect the learning efficiency of the model to some extent. To quantify that, we calculate the CL drop rate $\Delta\mathcal{L}_{cl}^{k,t_c}$. Then we simply quantify the learning efficiency of the client c_k with:

$$\mathcal{P}(\theta_k^{t_c}) = \Delta\mathcal{L}_{cl}^{k,t_c} = \mathcal{L}_{cl}^{k,t_c-1} - \mathcal{L}_{cl}^{k,t_c}, \tag{9}$$

where $t_c \in T_c$ represents the t_c-th communication round. By both label quality and learning efficiency, we can define the k-th client confidence as:

$$\mathcal{F}_k^{t_c} = \mathcal{Q}_{t_c}(\tilde{D}_k) \cdot \mathcal{P}(\theta_k^{t_c}). \tag{10}$$

It measure the confidence for each client respectively. We then determines the weight by $F_k^{t_c}$ as:

$$w_k^{t_c} = \frac{1}{K-1} + \eta \frac{\mathcal{F}_k^{t_c}}{\sum_{k=1}^{K} \mathcal{F}_k^{t_c}}, \tag{11}$$

where η is a hyper-parameter to control the impact of client confidence $\mathcal{F}$. Then perform softmax normalization:

$$\mathcal{W}_k^{t_c} = \frac{\exp(w^{t_c})_k}{\sum_{k=1}^{K} \exp(w_k^{t_c})}. \tag{12}$$

The above weighted regularization can minimize the knowledge of the noisy client from being learned. We dynamically weight the knowledge distribution learned by the client in each round as:

$$\theta_k^{t_c} \leftarrow \theta_k^{t_c-1} - \alpha \nabla_\theta (\mathcal{W}^{t_c} \cdot \mathcal{L}_{cl}^{k,t_c-1}). \tag{13}$$

with the training iteration, each model will be updated in the direction of the clean and efficient clients.

Summary

The entire pipeline of our method is summarized in Figure.2. First, each client c_k updates the local model θ_k with the private noisy dataset $\tilde{D}_k$ to get a set of pre-trained models. Then in aggregation phase, the clientc_k aligns the logit distribution of other clients to learn the knowledge from others. Therefore, in order to reduce the impact of intre-client noise, we use CL loss to update the local model in Eq.6. In order to reduce the impact of inter-client noise, we use label quality and learning efficiency in Eq.8 and Eq.9 to get the client confidence in Eq.12. Thus we align the involvement of the noisy client in the federated learning scenario and reduce the impact of noise during communication.

Experiments

Experimental Setting

Datasets and Models. Our experiments are conducted on CIFAR-10 and CIFAR-100 (Krizhevsky, Hinton et al. 2009). We set public dataset on the server as a subset of CIFAR-100, and we randomly divide CIFAR-10 to all clients as private datasets. The size of private dataset and public dataset are specified as $N_k = 10000$ and $N_0 = 5000$ respectively.

In our model-heterogeneous scenario, we apply ResNet10, ResNet12 (He et al. 2016), Mobilenet (Howard et al. 2017) and Shufflenet (Zhang et al. 2018) to four clients as local models.

Noise Type. We use the label transition matrix $\mathcal{M}$ to add label noise to the dataset, where $\mathcal{M}_{mn} = flip(\tilde{y} = n|y = m)$ represents that label y is flipped from the clean m,class to the noisy n class. We both use symmetric flip (Van Rooyen, Menon, and Williamson 2015b) and pair flip (Han et al. 2018b) as our noisy structures. *Symmetric flip* flips the original class label to any wrong class labels with equal probability, while *pair flip* flips the original class to a very similar wrong category.

Implementation Details. We perform $T_c = 40$ communication rounds for the server and $T_l = 2$ local learning epochs for all clients. Furthermore, we use the Adam optimizer(Kingma and Ba 2014) with an initial learning rate of α = 0.001 and the batch size of 256. For the hyper-parameters in our method, we set $\lambda = 0.1$ and $\eta = 0.5$. For the noisy rate in the label transition matrix, we set the noisy rate μ as 0.1 under both symmetric flip and pair flip noise types. To generate the noisy dataset $\tilde{D}$, we flip 20% of the labels in the training dataset of CIFAR-10 to the wrong labels and keep the test dataset of CIFAR-10 constant to observe the model performance. The c_k client randomly selects N_k samples from the shuffled CIFAR-10, so the client may have different proportions of noise.

Comparisons. We compare our method with the model-heterogeneous FL algorithm FedMD(Li and Wang 2019) and FedDF(Lin et al. 2020) under the same settings. FedMD is based on knowledge distillation, in which each client computes the class scores on public data and then approaches the consensus. FedDF builds a distillation framework for robust federated model fusion, which allows for heterogeneous models and data. To demonstrate the validity of our method in the homogeneous model case, we compare it with FedAvg(McMahan et al. 2017), FedMD and FedDF. FedAvg leverages the private dataset for local gradient descent, followed by the server aggregating the updated model on average. Since our setting is not the same as theirs, we use the key of these algorithms for our experiments.

Comparison Performance

Heterogeneous Federated Learning Methods. We first compare with the state-of-the-art heterogeneous FL method under the same setting. The baseline refers to the method in which the client trains local model on private dataset without FL. Therefore, the comparisons on two noise rates are shown in Table 1&2 The experiments demonstrate that our proposed method outperforms the existing strategies under

Method	Pairflip					Symflip				
	θ_1	θ_2	θ_3	θ_4	Avg	θ_1	θ_2	θ_3	θ_4	Avg
Baseline	77.98	76.75	66.89	74.33	73.99	76.20	76.05	64.96	74.31	72.88
FedMD	74.98	76.89	67.1	76.64	73.9	73.23	73.66	67.72	75.54	72.54
FedDF	76.26	75.51	68.41	76.04	74.06	72.07	75.18	67.38	74.47	72.28
Our Method	**78.86**	**78.76**	**69.6**	71.83	**74.76**	**78.40**	**78.36**	**69.47**	**76.93**	**75.79**

Table 1: Compare with the existing methods when the noise rate $\mu = 0.1$, θ_k represents the local model of the client c_k.

Method	Pairflip					Symflip				
	θ_1	θ_2	θ_3	θ_4	Avg	θ_1	θ_2	θ_3	θ_4	Avg
Baseline	72.31	71.84	61.78	69.67	68.90	72.01	70.15	59.62	69.42	67.80
FedMD	68.00	67.81	65.67	**74.02**	68.88	67.31	68.54	**64.48**	71.75	68.02
FedDF	68.66	69.68	62.36	72.12	68.21	67.36	68.56	63.60	70.83	67.59
Our Method	**77.81**	**76.09**	**66.61**	72.78	**73.32**	**78.14**	**76.77**	64.23	**73.90**	**73.26**

Table 2: Compare with the existing methods when the noise rate $\mu = 0.2$, θ_k represents the local model of the client c_k

Components			Pairflip	Symflip
HMA	CL	CWS		
			68.90	67.8
✓			66.51	66.26
	✓		69.27	69.61
✓	✓		70.96	73.26
✓	✓	✓	**73.32**	73.35

Table 3: Ablation study with the noise rate $\mu = 0.2$, θ_k means the local model of the client c_k.

various noisy settings. As the noise rate rises from 0.1 to 0.2, it can be seen that the average test accuracy of FedMD and FedDF drops significantly, by 5.02% for FedMD and 5.85% for FedDF on Pair-flip noise, and by 4.52% for FedMD and 4.69% for FedDF on Sym-flip noise, As for our method, it drops 1.44% on Pair-flip noise and 2.53% on Sym-flip noise, The above can prove that our proposed solution is robust against different noise settings.

Ablation Study. We first evaluate the effect of each component on noise rates $\mu = 0.2$ with two noise types (pairflip and symflip) in the heterogeneous model scenario to prove the effectiveness of each component.

*Effectiveness of Heterogeneous Model Alignment (**HMA**):* According to Table 3, we observe that the effect of adding HMA will have some degree of degradation than without FL. In our analysis, because HMA causes the clients to keep communicating learning the wrong knowledge and updating the model in the wrong direction.

*Effectiveness of Combinational Learning Loss (**CL**):* We add teh CL loss to the baseline to avoid the influence of nosy data during the local update phase. We can see in Table 3 that the performance of most models has been significantly improved. It can be inferred that the higher the noise rate, the better the performance of CL loss.

*Effectiveness of Client Weighting Scheme (**CWS**):* We add the CWS component to improve the robustness against noisy data from other clients in FL. As shown in Table 3, each model has achieve better performance. We can see that when the noise type is pairflip, the average test accuracy of models has increased from 70.96% to 73.32%.

Conclusion

In our Deep Learning Project, we study a thorny problem of how to perform the robustness with heterogeneous clients with noise. To address this issue, We propose our method with three components. We first align the feedback distribution on public dataset on the server to enable the aggregation between heterogeneous model on clients. Then to avoid each model overfitting to noise on own private data, we combine the CE loss and RCE loss to update the local model. Finally, for the noisy feedback from other participants, we propose a flexible re-weighting method to estimate the client label quality and learning efficiency, which effectively avoids the impact from noisy clients and achieves robust federated learning. We design experiments to prove the effectiveness of each component included in our approach.

References

Arpit, D.; Jastrzebski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M. S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; et al. 2017. A closer look at memorization in deep networks. In *International conference on machine learning*, 233–242. PMLR.

Diao, E.; Ding, J.; and Tarokh, V. 2020. Heterofl: Computation and communication efficient federated learning for heterogeneous clients. *arXiv preprint arXiv:2010.01264*.

Ghosh, A.; Kumar, H.; and Sastry, P. S. 2017. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.; and Sugiyama, M. 2018a. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31.

Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.; and Sugiyama, M. 2018b. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31.

He, C.; Annavaram, M.; and Avestimehr, S. 2020. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in Neural Information Processing Systems*, 33: 14068–14080.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

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 *International conference on machine learning*, 2304–2313. PMLR.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2): 1–210.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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

Laine, S.; and Aila, T. 2016. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*.

Li, D.; and Wang, J. 2019. Fedmd: Heterogenous federated learning via model distillation. *arXiv preprint arXiv:1910.03581*.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2: 429–450.

Liang, P. P.; Liu, T.; Ziyin, L.; Allen, N. B.; Auerbach, R. P.; Brent, D.; Salakhutdinov, R.; and Morency, L.-P. 2020. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*.

Lin, T.; Kong, L.; Stich, S. U.; and Jaggi, M. 2020. Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33: 2351–2363.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Miyato, T.; Maeda, S.-i.; Koyama, M.; and Ishii, S. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8): 1979–1993.

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*, 1944–1952.

Shen, T.; Zhang, J.; Jia, X.; Zhang, F.; Huang, G.; Zhou, P.; Kuang, K.; Wu, F.; and Wu, C. 2020. Federated mutual learning. *arXiv preprint arXiv:2006.16765*.

Sukhbaatar, S.; Bruna, J.; Paluri, M.; Bourdev, L.; and Fergus, R. 2014. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*.

Sun, L.; and Lyu, L. 2020. Federated model distillation with noise-free differential privacy. *arXiv preprint arXiv:2009.05537*.

Tam, K.; Li, L.; Han, B.; Xu, C.; and Fu, H. 2021. Federated noisy client learning. *arXiv preprint arXiv:2106.13239*.

Van Rooyen, B.; Menon, A.; and Williamson, R. C. 2015a. Learning with symmetric label noise: The importance of being unhinged. *Advances in neural information processing systems*, 28.

Van Rooyen, B.; Menon, A.; and Williamson, R. C. 2015b. Learning with symmetric label noise: The importance of being unhinged. *Advances in neural information processing systems*, 28.

Wang, H.; Yurochkin, M.; Sun, Y.; Papailiopoulos, D.; and Khazaeni, Y. 2020. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*.

Wang, Y.; Ma, X.; Chen, Z.; Luo, Y.; Yi, J.; and Bailey, J. 2019. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, 322–330.

Wei, H.; Feng, L.; Chen, X.; and An, B. 2020. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13726–13735.

Yao, J.; Wu, H.; Zhang, Y.; Tsang, I. W.; and Sun, J. 2019. Safeguarded dynamic label regression for noisy supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 9103–9110.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhang, X.; Zhou, X.; Lin, M.; and Sun, J. 2018. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6848–6856.