Alleviating Non-IID in Federated Learning by Generating Fake Samples

Yiqun Diao, Jun Zhi, Yilin Zhang and Menglin Li

National University of Singapore

Abstract

Machine learning algorithms are data hungry. However, due to privacy concerns, it is impractical in most cases to collect all original data for centralized training. In order to perform machine learning on different data silos while preserving data privacy, federated learning (FL) is proposed as a promising solution. Nevertheless, it has been shown that FL algorithms suffer from data heterogeneity. To alleviate the accuracy loss induced by data heterogeneity, we propose to generate fake samples to facilitate FL training. Experiments show that generated fake samples can improve global model's test accuracy in most cases, especially in extreme non-IID cases.

Introduction

Federated learning (FL) is proposed to let data owners collaboratively train a better machine learning model without exposing raw data. For example, hospitals can collectively train a FL model for diagnosing diseases, while protecting the privacy of individual patient.
It has been shown that data heterogeneity is a challenging problem in federated learning, since non-IID data distribution among FL clients can degrade global model accuracy [1].

FedGFS: Generate Fake Samples

In order to alleviate the data heterogeneity in FL, we train a GAN for each class, generate fake data, distribute to all clients, and finally perform FL on both original samples and fake samples.

- **Calling for Generators:** For each class, we assign the task of training GAN to client with the most samples of that class. Server collects generators and generates fake samples of all classes. Those fake samples will be distributed to all clients.
- **Post-training:** After receiving fake samples, the server just performs FL algorithms among clients. In our experiments, we directly apply FedAvg [2].

FedGFS Experiments

We utilize the settings in [1]. $\#C = k$ means each client only possesses k different labels. $p_k \sim Dir(\beta)$ denotes that for each label, we sample from Dirichlet distribution $p_k \sim Dir_N(\beta)$ and assign a $p_{k,j}$ proportion of class k samples to client j. By default, we set 10 clients, batch size as 64, local epoch as 10 and local learning rate as 0.01.
For FedGFS, we train 200 epochs for GAN of each class, with learning rate 0.01, batch size 64. For each class, we generate two batches of fake samples, i.e. 128 samples per class.

Table 1:The top-1 accuracy of different approaches. Test accuracy of four benchmark algorithms are from [1].

dataset	partitioning	FedAvg	FedProx	SCAFFOLD	FedNova	FedGFS
MNIST	$p_k \sim Dir(0.5)$	98.9%	98.9%	**99.0%**	98.9%	98.9%
	$\#C = 1$	29.8%	0.9%	9.9%	39.2%	**98.4%**
	$\#C = 2$	97.0%	96.4%	95.9%	94.5%	**98.5%**
	$\#C = 3$	98.0%	97.9%	96.6%	98.0%	**98.7%**
FMNIST	$p_k \sim Dir(0.5)$	88.1%	88.1%	88.4%	88.5%	**89.0%**
	$\#C = 1$	11.2%	28.9%	12.8%	14.8%	**82.1%**
	$\#C = 2$	77.3%	74.9%	42.8%	70.4%	**82.8%**
	$\#C = 3$	80.7%	82.5%	77.7%	78.9%	**84.9%**
CIFAR-10	$p_k \sim Dir(0.5)$	68.2%	67.9%	**69.8%**	66.8%	50.1%
	$\#C = 1$	10.0%	12.3%	10.0%	10.0%	**29.8%**
	$\#C = 2$	49.8%	**50.7%**	49.1%	46.5%	40.3%
	$\#C = 3$	**58.3%**	57.1%	57.8%	54.4%	44.4%
SVHN	$p_k \sim Dir(0.5)$	86.1%	86.6%	**86.8%**	86.4%	86.2%
	$\#C = 1$	11.1%	9.6%	6.7%	10.6%	**77.8%**
	$\#C = 2$	**80.2%**	79.3%	62.7%	75.4%	79.4%
	$\#C = 3$	82.0%	82.1%	77.2%	80.5%	**82.4%**

Results are shown in Table 1. In most cases, FedGFS can outperform other four baseline algorithms in [1]. FedGFS can achieve much better accuracy in extreme non-IID cases, e.g. $\#C = 1$ partition. However, in slight non-IID case like $p_k \sim Dir(0.5)$, its accuracy may be a little lower. Moreover, FedGFS is better in MNIST and Fashion-MNIST, but does not perform well in CIFAR-10 and SVHN.

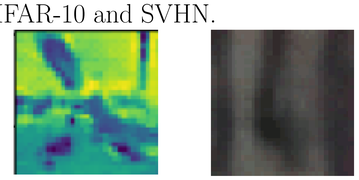

Figure 1:Generated fake samples of CIFAR-10 and SVHN. The left figure is labeled plane and the right one is labeled four.

Discussions

The performance of FedGFS under different settings can be explained by the quality of generator. In $\#C = 1$ partition, each generator is trained on all samples of that class. However in $p_k \sim Dir(0.5)$ partition, the client in charge of training generator may have only a part of samples for that class. Fewer samples lead to lower generator quality. For MNIST and Fashion-MNIST dataset, their images are 28*28 tensors, while CIFAR-10 and SVHN images are 32*32*3 tensors. Therefore, it would be more difficult to train a high-quality generator for CIFAR-10 and SVHN. Low-quality generator leads to low-quality generated fake samples, which can harm the accuracy of final FL models.
We can see from Figure 1 that the generated image of CIFAR-10 and SVHN is not clear.

Future Work: Adversarial Learning

1. Fast gradient sign method

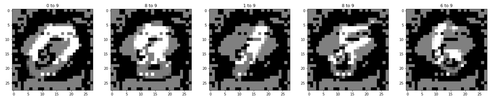

Figure 2:Fast gradient sign method on MNIST dataset.

One pixel attack

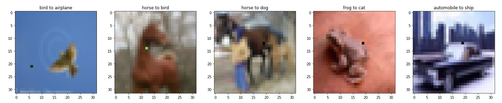

Figure 3:One pixel attack on CIFAR dataset.

2. Defense strategy
   1. Generate sets of adversarial images in client side.
   2. Client trains on both private samples and adversarial samples.
   3. Server aggregates models with FedAvg [2] or other FL algorithms.
3. Adversarial defence method can also be applied to augment more samples to train GAN in FedGFS. More samples may improve the quality of generators.

Future Work: FL on GAN

One can apply FedAvg [2] to train GAN with data of multiply clients to improve generator's quality.
As shown in Figure 4, there are two possible options to train GAN using FedAvg: (1) only aggregate discriminators, while training generators on server side; (2) aggregate both generators and discriminators.

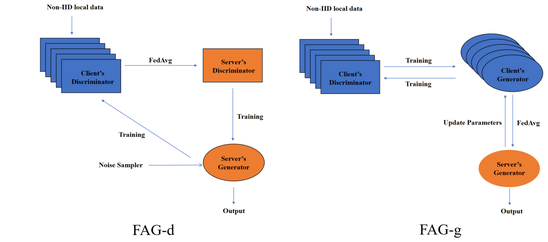

Figure 4:Two possible approaches to train GAN using FedAvg

Conclusion

We propose FedGFS to generate fake samples to alleviate non-IID data distribution in FL. Experimental results show that FedGFS can outperform four baseline FL algorithms in most cases. We also propose methods to further improve the quality of generators.

References

[1] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study.
IEEE International Conference on Data Engineering, 2022.

[2] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data.
arXiv preprint arXiv:1602.05629, 2016.

Contact Information

- Web: github.com/Shueryaoli/CS5260project
- Email: yiqun@comp.nus.edu.sg

NUS National University of Singapore | School of Computing