PRIVATE FL-GAN: DIFFERENTIAL PRIVACY SYNTHETIC DATA GENERATION BASED ON FEDERATED LEARNING

Bangzhou Xin[1], *Wei Yang*[1,⋆], *Yangyang Geng*[1], *Sheng Chen*[1], *Shaowei Wang*[2,⋆], *Liusheng Huang*[1]

[1]University of Science and Technology of China
[2]Tencent Games

ABSTRACT

Generative Adversarial Network (GAN) has already made a big splash in the field of generating realistic "fake" data. However, when data is distributed and data-holders are reluctant to share data for privacy reasons, GAN's training is difficult. To address this issue, we propose private FL-GAN, a differential privacy generative adversarial network model based on federated learning. By strategically combining the Lipschitz limit with the differential privacy sensitivity, the model can generate high-quality synthetic data without sacrificing the privacy of the training data. We theoretically prove that private FL-GAN can provide strict privacy guarantee with differential privacy, and experimentally demonstrate our model can generate satisfactory data.

Index Terms— information security, federated learning, differential privacy, data generation

1. INTRODUCTION

With the advent of the era of big data, machine learning has ushered in its spring. In order to meet the needs of different scenarios, many machine learning models have been developed, such as Convolution Neural Network (CNN) [1], Recurrent Neural Network (RNN) [2], Generative Adversarial Network (GAN) [3], etc. Most of them are based on a large amount of training data. In some areas, we cannot easily get the data we need. For example, data-holders may refuse to disclose personally identifiable information because of privacy concerns.

Fortunately, generative models [4] provide a promising direction to alleviate data scarcity. GAN and its variants are impressive in generating high-quality "fake" samples [5, 6] that are difficult to be distinguished from reality. Examples of applications include: generating images from text descriptions [7], generating videos from still images [8], increasing image resolution [9]. [10] first proposed using GANs to generate synthetic data, and made some progress in the field of synthetic data. However, the program does not provide strict privacy protection.

⋆Corresponding authors:qubit@ustc.edu.cn; seawellwang@tencent.com

This work was supported by the Anhui Initiative in Quantum Information Technologies (No. AHY150300).

Relying on the Post-Processing Theorem, DPGAN [11] proposed a framework for modifying the GAN framework to be differentially private [12] (a formal definition of Differential Privacy is supplied in Sec. 2.4). The key is to add noise to the discriminator's gradient during training, which allows the entire GAN framework to meet differential privacy requirements. Based on this, [13] proposed PATE-GAN which modified the PATE framework and applied it to GAN to generate synthetic data. Later, Xu et al. developed GANobfuscator with a precision-designed gradient pruning strategy in [14], achieving high-quality synthetic data generation.

However, some situations have limitations in training the above learning models, e.g., data storage is distributed. For privacy reasons, data-holders are reluctant to share their data, which makes centralized model training impossible. In response to this situation, Robin et al. [15] proposed a differential privacy solution for federated learning [16]. In addition, in order to optimize the training of GAN models in distributed datasets, [17] proposed a method called MD-GAN. Although MD-GAN did not consider privacy protection, it provided us with inspiring ideas in model training.

In this paper, we present private federated learning of GAN (private FL-GAN), a novel method to train a GAN in a distributed way. Through serialized training between clients, private FL-GAN achieves a well-trained privacy-preserving data generation model. Compared with the method proposed in [15], our algorithm not only saves communication expenses but also improves the performance of the generated model. We evaluate private FL-GAN under various benchmark datasets and demonstrate that private FL-GAN can generate high-quality data with an acceptable privacy budget.

2. PRIVATE FEDERATED LEARNING OF GAN

2.1. Algorithm summary

Compared with adding noise to the final parameters of the model in previous work, we choose to add noise during the training process to achieve the purpose of privacy protection. At the same time, the performance of the algorithm is greatly improved. We use moments accountant [18] to record real-time privacy losses in training. In terms of synthetic data

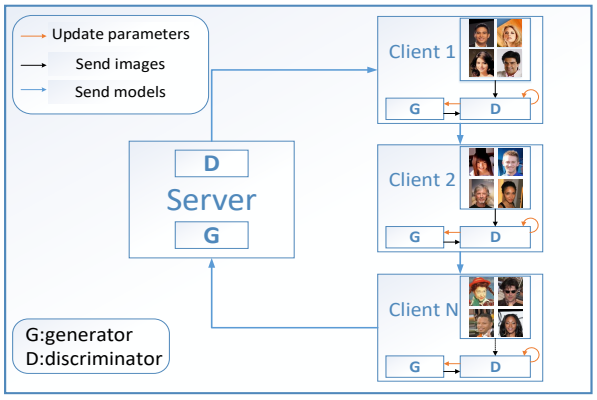

Fig. 1: Algorithm Framework

generation model selection, we choose WGAN with gradient penalty [19]. As to how to make overall use of distributed storage data to train the model, there are two options. One of them can be called "parallel training" as described in [15]. The other is our "serial training".

2.2. Algorithm framework

The core idea of "parallel training" proposed by Robin et al. [15] is that the server averages the parameter updates of multiple clients to get updates for each round. However, when each client has limited data, access to the data will be more frequent in order to train an effective model, increasing the risk of privacy leakage. Updating the model after accessing all the clients is undoubtedly a waste of data usage and increases the risk of privacy exposure. Inspired by [17], we propose that each client sequentially updates the parameters of the same model, training better models with less data access. The framework of our algorithm is shown in Fig. 1.

2.3. The model learning procedure

Each client has its own unique dataset. By adding noise to the training process, the resulting model can satisfy differential privacy. Therefore, even if the model is made public, the data privacy of the client will not be violated. We present the detailed algorithm of private FL-GAN in Alg. 1. The following is the algorithm flow:

1) The model is initialized by the server, including discriminator and generator. The server sends the model to any client i.

2) In each round of generator training for a total of T_g rounds of client i, generator is updated after discriminator being trained T_d times.

3) Randomly pick one of the remaining clients and perform step two until all clients participate in the training and return the final model to the server.

Algorithm 1

Input: α_d, learning rate of discriminator. α_g, learning rate of generator. c_g, bound on the gradient of Wasserstein distance with respect to weights. m, batch size. T_d, number of discriminator iterations per generator iteration. T_g, generator iteration. σ_n, noise scale. λ, penalty coefficient. N, the set of data-holders (clients).

Output: Differential privacy data generation model

1: Initialize discriminator parameters $\boldsymbol{\omega}$, generator parameters $\boldsymbol{\theta}$;
2: **while** N is not empty **do**
3: Take out a client from N
4: **for** t_1 = 1, 2, ... , T_g **do**
5: **for** t_2 = 1, 2, ... , T_d **do**
6: **for** i = i, 2, ..., m **do**
7: Sample real data $\boldsymbol{x} \sim p_n(\boldsymbol{x})$, latent variable $\boldsymbol{z} \sim p(\boldsymbol{z})$, a random number $\epsilon \sim U[0,1]$;
8: $\tilde{\boldsymbol{x}} \leftarrow G_\theta(\boldsymbol{z}), \hat{\boldsymbol{x}} \leftarrow \epsilon\boldsymbol{x} + (1-\epsilon)\tilde{\boldsymbol{x}}$
9: $\boldsymbol{g} \leftarrow (\|\nabla_{\hat{\boldsymbol{x}}} D_w(\hat{\boldsymbol{x}})\|_2 - 1)^2$
10: $\boldsymbol{L}^{(i)} \leftarrow D_w(\tilde{\boldsymbol{x}}) - D_w(\boldsymbol{x}) + \lambda \boldsymbol{g}$
11: $\boldsymbol{\eta} \sim N(0, \sigma_n^2 c_g^2 \boldsymbol{I})$ //generate noise
12: $\boldsymbol{w} \leftarrow Adam((\frac{1}{m}\Sigma_{i=1}^{m} \nabla_w \boldsymbol{L}^{(i)} + \boldsymbol{\eta}), \boldsymbol{w}, \alpha_d)$
13: Sample a batch of latent samples$\{\boldsymbol{z}_i\}_{i=1}^{m} \sim p(\boldsymbol{z})$
14: $\boldsymbol{\theta} \leftarrow Adam(\nabla_\theta \frac{1}{m}\Sigma_{i=1}^{m} - D_w(G_\theta(\boldsymbol{z})), \boldsymbol{\theta}, \alpha_g)$
15: updating privacy accountant with (T_g, σ_n, c_g)
return $\boldsymbol{\theta}$

2.4. Theoretical analysis

In general, private FL-GAN is built on the framework of W-GAN with gradient penalty and achieves differential privacy by injecting noise when updating discriminator. In particular, after calculating the discriminator gradient for each training data, gaussian noise is added to protect the privacy of training data (lines 10, 11 in Alg. 1). In dealing with privacy budgets, we use a privacy accountant [18] to track cumulative privacy loss in training.

Definition 1. *(Differential Privacy [20]) A randomized function M gives (ϵ, δ)-differential privacy if for all datasets D_1 and D_2 differing on a single record, and all $S \subseteq Range(M)$,*

$$Pr[M(D_1) \in S] \leq exp(\epsilon) \times Pr[M(D_2) \in S] + \delta \quad (1)$$

The parameter ϵ called privacy budget controls the degree of protection and noise level.

Definition 2. *(Parallel Composition [20]) For disjoint subsets $x_i \subseteq x$, let query $f(x_i)$ satisfy ϵ-differential privacy; then applying all queries $f(x_i)$ still satisfies ϵ-differential privacy.*

Lemma 1. *[11, 18] Given the sampling probability $q = m/M$ (M represents the dataset size), the number of discriminator iterations in each inner loop T_d and privacy violation*

(a) (b) (c)

Fig. 2: Synthetic data with three different ϵ on MNIST dataset. (a) $\epsilon = 8$. (b) $\epsilon = 6$. (c) $\epsilon = 4$.

(a) (b) (c)

Fig. 3: Synthetic data with three different ϵ on CelebA dataset. (a) $\epsilon = 8$. (b) $\epsilon = 6$. (c) $\epsilon = 4$.

δ, for any positive ϵ, the parameters of discriminator guarantee (ϵ, δ)-differential privacy with respect to all the data used in that outer loop if it satisfies:

$$\sigma_n = \frac{2q\sqrt{T_d log(\frac{1}{\delta})}}{\epsilon} \tag{2}$$

Theorem 1. *The output of generator learned in Alg.1 guarantees (ϵ, δ)-differential privacy.*

Proof. According to Lemma 1, the discriminator trained in each client has satisfied differential privacy, and with the effect of the post-processing property of differential privacy [20], the generator also satisfies differential privacy. Due to the Parallel Theory [20], when the model is passed among clients, it does not increase the risk of revealing data privacy of other clients. Therefore, the model derived from each round of client training is (ϵ, δ)-differentially private. □

Finally, let's consider the complexity of the entire algorithm, mainly the communication overhead among the clients. In every communication between clients, the parameters only need to be passed among the clients once, i.e., the total communication complexity is $N \cdot (|\omega| + |\theta|)$. However, according to the DP-FL scheme, the parameters need to be passed T_d times in one iteration of the generator, and the total communication complexity is $T_g \cdot T_d \cdot N(|\omega| + |\theta|)$, $T_d \cdot T_g$ times as much as ours.

3. EXPERIMENTS

We present extensive experiments to investigate the relationship between the privacy level and the quality of the generated data. The experiments are conducted on two benchmark datasets (MNIST[1] and CelebA[2]): MNIST, which consists of 70 K handwritten digit images of size 28×28; CelebA, which contains 200 K celebrity face images of size 64×64. We set the learning rate of discriminator α_d and generator α_g to be 1.0×10^{-4}, then exponentially decay. The batch size is 64. For each client to keep a certain amount of data for training, we split MNIST into $N_1 \in [1, 3, 6]$ parts, so we can simulate N_1 different data-holders. For CelebA, we divide the dataset into $N_2 \in [1, 10, 20]$ parts, respectively. According to [11, 18], we set the noise scale δ to 10^{-5}, and the number of iterations on discriminator (T_d) is 5. We set the activation function on discriminator network to leaky ReLU, so the

[1]MNIST: http://yann.lecun.com/exdb/mnist/

[2]CelebA: http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

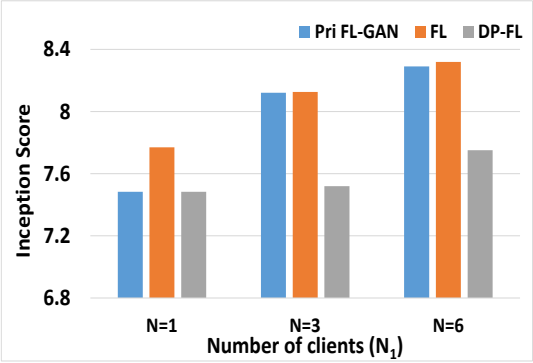

Fig. 4: Comparison of IS on MNIST dataset

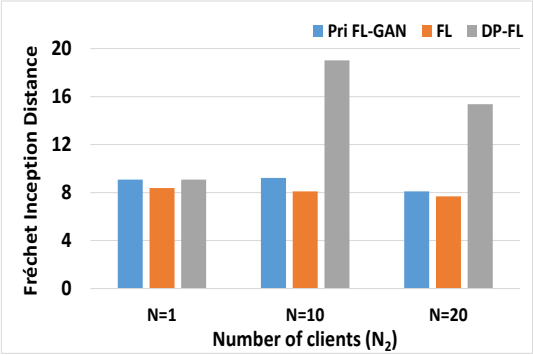

Fig. 5: Comparison of FID on CelebA dataset

bound on the derivative of the function $B_{\delta'} \leq 1$.

In order to explore the impact of the specific value of the privacy level on the quality of the images, we have conducted several experiments on datasets. In these experiments, we trained by setting different privacy parameters ϵ and got several models of privacy protection levels. The generated images are shown in Fig. 2 and Fig. 3, corresponding to different levels of privacy parameters. It can be seen that we can generate clear images when the privacy level is high. And large privacy parameters correspond to high-quality images, which indicates the distortion of the image is caused by noise rather than a poor quality training set. According to [20], large privacy parameter means great risk of privacy breaches, but it also means better generated data. This is a trade-off between privacy and performance.

Using the same configuration and parameters, we compared our algorithm with no privacy protection (FL-GAN) and the algorithm (DP-FL) proposed in [15]. After the three models have been trained for the same number of rounds, we calculate the Inception Score (IS) [21] of synthetic data, with the situation of three different numbers of data-holders. For this evaluation index, the higher the score, the higher the quality of the generated image, and the greater the diversity. From Fig. 4, we can see that our algorithm has higher quality of synthesized image than DP-FL under any privacy parameters.

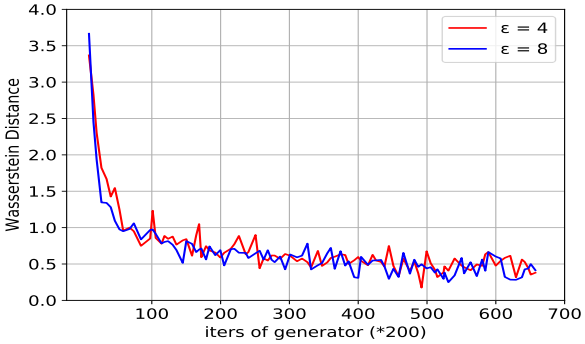

Fig. 6: Wasserstein distance with different ϵ

And when the privacy parameters reach a certain threshold, the quality of the synthesized image can be comparable to FL-GAN which has no privacy protection.

To make the results more convincing, we also used the Fréchet Inception Distance (FID) [22] to evaluate the generated data. Contrary to the Inception Score, under the Fréchet Inception Distance standard, low score represents a higher quality of synthetic data. As can be seen from Fig. 5, the data generated by our model is still outstanding under this standard.

After verifying that our model is capable of generating good quality images, we next evaluate the relationship between privacy levels and network convergence. Since WGAN with gradient penalty applies gradient penalty independently to each sample, and the Lipschitz limit requires that the discriminator's gradient does not exceed K (which is usually set to 1), gradient penalty is to set an additional loss term to associate gradient with K. The parameter K and the differential privacy sensitivity are perfectly matched. According to [18], before adding noise to the gradient, we need to clip it so that the gradient has a clear upper bound as the sensitivity. Here, we set the clipping value as K, to avoid vanishing and exploding of the gradient, and to ensure the boundedness of the gradient to facilitate the addition of gaussian noise. The experimental results are shown in Fig. 6, as can be seen from figure, our model can still converge rapidly even if the gradient is added with noise.

4. CONCLUSION

The private FL-GAN proposed in this paper is used as privacy-preserving synthetic data generation. Aiming at the distributed storage of data, our algorithm uses differential privacy technology to obtain a synthetic data generation model in the case of protecting training data. We strategically use "serial training" method to make maximum use of the data of each database, so as to quickly train a high-quality model. We conduct experiments on two public datasets, and the experimental results show that our algorithm can generate high-quality synthetic data. For future work, we consider extending the type of generated data.

5. REFERENCES

[1] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning.," *Nature*, vol. 521, no. 7553, pp. 436, 2015.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, MIT press, 2016.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[4] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[5] Alexey Dosovitskiy and Thomas Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Advances in neural information processing systems*, 2016, pp. 658–666.

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[7] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee, "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016.

[8] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba, "Generating videos with scene dynamics," in *Advances In Neural Information Processing Systems*, 2016, pp. 613–621.

[9] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[10] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun, "Generating multi-label discrete patient records using generative adversarial networks," *arXiv preprint arXiv:1703.06490*, 2017.

[11] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou, "Differentially private generative adversarial network," *arXiv preprint arXiv:1802.06739*, 2018.

[12] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[13] James Jordon, Jinsung Yoon, and Mihaela van der Schaar, "Pate-gan: generating synthetic data with differential privacy guarantees," 2018.

[14] Chugui Xu, Ju Ren, Deyu Zhang, Yaoxue Zhang, Zhan Qin, and Kui Ren, "Ganobfuscator: Mitigating information leakage under gan via differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 9, pp. 2358–2371, 2019.

[15] Robin C Geyer, Tassilo Klein, and Moin Nabi, "Differentially private federated learning: A client level perspective," *arXiv preprint arXiv:1712.07557*, 2017.

[16] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon, "Federated learning: Strategies for improving communication efficiency," *arXiv preprint arXiv:1610.05492*, 2016.

[17] Corentin Hardy, Erwan Le Merrer, and Bruno Sericola, "Md-gan: Multi-discriminator generative adversarial networks for distributed datasets," in *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2019, pp. 866–877.

[18] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 308–318.

[19] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville, "Improved training of wasserstein gans," in *Advances in neural information processing systems*, 2017, pp. 5767–5777.

[20] Cynthia Dwork, Aaron Roth, et al., "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[21] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, and Chen Xi, "Improved techniques for training gans," 2016.

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Advances in Neural Information Processing Systems*, 2017, pp. 6626–6637.