Proceedings of the 4th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/52/20241136

Exploring the potential of federated learning for diffusion model: Training and fine-tuning

Shuo Chen

School of Informatics, University of Edinburgh, Edinburgh, EH8 9AB, UK

s.chen-136@sms.ed.ac.uk

Abstract. Diffusion models, a state-of-the-art generative model, have drawn attention for their capacity to produce high-quality, divers, and flexible content. However, the training of these models typically necessitates large datasets, a task that can be hindered by challenges related to privacy concerns and data distribution constraints. Due to the amount of data and hardware required for large model training, all centralized training will be done by large companies or labs with computing power. Federated Learning provides a decentralized method that allows for model training across several data sources while maintaining the data's localization, reducing privacy threats. This research proposes and evaluate a novel approach for utilizing Federated Learning in the context of diffusion models. This paper investigates the feasibility of training and fine-tuning diffusion models in a federated setting, considering various data distributions and privacy constraints. This study used the Federated Averaging (FedAvg) technique to train the unconditional diffusion model as well as to fine-tune the pre-trained diffusion mode. The experimental results demonstrate that federated training of diffusion models can achieve comparable performance to centralized training methods while preserving data locality. Additionally, Federated Learning can be effectively applied to fine-tune pre-trained diffusion model, enabling adaptation to specific tasks without exposing sensitive data. Overall, this work demonstrates Federated Learning's potential as a useful tool for training and fine-tuning diffusion models in a privacy-preserving manner.

Keywords: Federated Learning, Diffusion Model, AIGC.

1. Introduction

In recent years, the popularity of Artificial Intelligence Generated Content (AIGC) has surged, spanning disciplines like Computer Vision (CV) and Natural Language Processing (NLP). For instance, the recently emerged image generation models Stable Diffusion (SD) [1], Imagen [2] and DALL-E [3], or the language generation model Chat Generative Pre-Trained Transformer (ChatGPT) [4]. They all generate high-quality synthetic content and guarantee authenticity and diversity. Large generative models are typically trained with hundreds of millions of parameters [1] and require large amounts of training data up to terabytes or petabytes to perform high-resolution text-to-image, text-to-video tasks. Simultaneously, the training process for extensive generative models mandates significant computational and storage capabilities. Resource-constrained devices are not capable of complete training of large generative models. Therefore, most of the big models are centralized trained by a

DOI: 10.54254/2755-2721/52/20241136

handful of Big Tech companies. Moreover, as the scale of these models expands, there's a parallel increase in the accessibility of data.

The centralized data on the server-side raise considerations about the data authority, data security and data privacy in the training dataset. The training dataset is predominantly compiled from online sources, although users are often unaware if their personal data has been utilized. Taking face generation [5] as well as portrait generation as an example, training a portrait generation model requires the collection of many portraits or face images sourced from online platforms. However, an individual's face holds a pivotal role in their identity, constituting one of the most sensitive pieces of personal information. This stands in contrast to online accounts and passwords, which can be altered. Given the relative permanence of facial features, the accumulation of facial data unavoidably raises concerns over personal privacy. Hence, the collection of facial data will inevitably involve personal privacy issues. And collection of facial data has the potential risk of massive facial data leakage [6].

To address these challenges mentioned above, a distributed training approach that is different from centralized training can be considered. Federated learning (FL) [7] is a distributed learning algorithm proposed by Google. In Federated learning, multiple clients collaboratively train a shared model by leveraging local data. During each training round, clients keep all training data locally. Only the learned parameters (model weights) need to be transferred to the central server. Finally, server-side aggregates of local updates are used to update the global model. The updated model is then broadcast to all clients. The entire training process does not require any direct data interaction to build a robust model. In addition, federated learning also significantly reduced communication costs since it avoids the raw data exchange. Thus, FL is promising in training generative model in distributed and privacy-preserving manner.

This study focused on leveraging federated learning for generation diffusion model [8] training. The diffusion model was chosen because the backbone recent popular image generation models is the diffusion model. This study set up two types of federated diffusion model framework. One is pre-training the unconditional diffusion model using Federated Averaging (FedAvg) algorithm [7], and the other is fine-tuning [9] the stable diffusion model in a distributed way by using FedAvg algorithm. This study experimented with the above two frameworks separately. By comparing the results of centralized training with results of distributed training, this study evaluated whether distributed diffusion model training also guarantees the generation of high-quality and photorealistic synthetic images. Finally, the feasibility and limitations of distributed learning applied to diffusion modelling were discussed.

2. Method

This section presents a short overview of federated learning algorithm and deep generative models. Furthermore, this section provides the technical background of diffusion fine-tuning.

2.1. Denoising Diffusion Models (DDPM)

DDPM is a novel generative model proposed in 2020 by Ho et al. [8]. The diffusion model is a step-by-step transformation of a noise obeying a Gaussian distribution into an image that matches desired distribution. A U-net [10] usually acts as a backbone to predict noise and complete the denoise process. There are two parts in training diffusion model. The first part is forward diffusion process, and the other part is reverse diffusion process.

2.1.1. Forward diffusion process

A sequence of noisy samples, $x_1 ..., x_T$, are produced by the forward diffusion process given a data set $x_0 \sim q(x)$ sampled from the true distribution. The variance scheduler t regulates the level of noise. The data sample x_0 gradually loses its distinguishing characteristics as the step t increases. Eventually, as t increases, an isotropic Gaussian distribution replaces x_t. Equation 1 below can be used to express the forward diffusion process:

$$\mathrm{q}(x_t \mid x_{t-1}) = \mathrm{N}(x_t; \sqrt{1-\beta_t} x_{t-1}, \beta_t \mathrm{I}) \tag{1}$$

DOI: 10.54254/2755-2721/52/20241136

Where $\sqrt{1-\beta_t}x_{t-1}$ is the Gaussian distribution's mean and β_t is the Gaussian distribution's variance. So the noisy sample xt can be express as the following equation 2:

$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta}_t\epsilon \quad (2)$$

Where ϵ is random sampled Gaussian noise. The Figure 1 illustrated both diffusion processes.

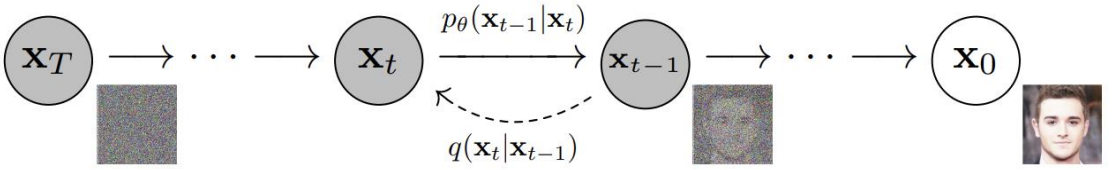

Figure 1. Directed Diffusion model graphical [8].

2.1.2. Reverse diffusion process

The reverse process is the denoising inference process of diffusion. By reversing the forward process and sampling from $q(x_{t-1} \mid x_t)$, the real sample can be reconstructed from a Gaussian input sample. Due to the need for the whole dataset, $q(x_{t-1} \mid x_t)$ is not easily to estimate. In order to conduct the reverse process, a model p_θ must be trained to understand how to approximate these conditional probabilities. Typically, a U-net is used to predict the noise at time t, and by denoise this predicted noise the image at time x_{t-1} can be restored. The training objective can be formulated as minimizing the distance between the real noise ϵ_t and the noise estimation $\epsilon_\theta(xt, t)$ by the model.

2.2. FederatedAveraging (FedAvg)

FedAvg [7] is a distributed framework that allows multiple users to train a machine learning model simultaneously. FedAvg algorithm involves the following steps: 1) Initialization of global model parameters w_0. 2) The central server transmits global model parameters w_0 to all participating clients and proportionally selects a number of clients for local training. 3) The Client uses the global model parameters to initialize the local model parameters, followed by local training. When a predefined number of local training times E is reached, the locally trained model parameters w will be uploaded into the server. 4) Using weighted average aggregation, the server transforms local model parameters into global model parameters. Repeat from step 2 until the number of communications rounds reaches t times.

2.3. Diffusion Fine-tuning

Fine-tuning is a commonly used strategy within transfer learning methodologies. Its primary aim is to leverage the knowledge acquired from pre-trained models and apply it effectively to various new problems, often referred to as downstream tasks. Conventional fine-tuning entails making adjustments to the entire set of pre-trained model weights using a limited number of samples. In 2021, Microsoft introduced a fine-tuning approach for large language models known as Low Rank Adaptation (LoRA) fine-tuning [11]. Subsequently, this method was also adopted for fine-tuning the Stable Diffusion model. Assuming that the amount of weight change in the model during task adaptation is low rank, a bypass weights Δw is added next to the original pre-trained model to do a dimensionality reduction and then dimensionality increase operation. The parameters of the pre-trained model are fixed during training, and only the downscaling matrix A and the upscaling matrix B are trained. Since $\Delta w = BA$, for LoRA, only the bypass weight Δw need to be trained.

2.4. Federated diffusion pre-training and Federated LoRA diffusion fine-tuning

In this section the entire framework of federated diffusion training and fine-tuning will be laid out.

DOI: 10.54254/2755-2721/52/20241136

2.4.1. Federated diffusion pre-training

The federated diffusion model training considers scenarios where a C number of clients with sufficient computing power and large amounts of data. The FedAvg algorithm is used to optimise the global model. The following is the training procedure: 1) Initialization of global diffusion model parameters w 2) Server send the global parameter w to a random selected subset $c \in C$ to do the local training ($Local\ training\ step = E$) and transfer the updated weight v to the server 3) Server aggregates the updated v to the new parameters w'. And repeat the step 2 and 3 until the reach the communication round T.

2.4.2. Federated LoRA diffusion fine-tuning

FedAvg algorithm is used to optimise the global model. But this time the weight w of the global model is not updated, only the bypass weight Δw is updated. The whole process is as follows: 1) Initialization of global bypass weights Δw 2) Server send bypass weights Δw to a random selected subset c $\in$ C to do the local fine-tuning (Local finetuning step = E) and transfer the updated bypass weights Δv to the server 3) Server aggregates the updated bypass weights Δv to the new parameters $\Delta w'$. And repeat the step 2 and 3 until the reach the communication round T.

3. Experiment results and discussion

3.1. Federated diffusion pre-training experimental details

The first experiments carried out is federated pre-training. The code is implemented by the PyTorch [12] framework. The dataset used in experiment is CelebA [13] for an unconditional face generation pre-training. This dataset contains over 200k images of celebrities. To reduce the amount of computation, the images is resized to 64×64. The linear diffusion schedule ranging from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$ and time step $T = 1000$ are used. The learning rate is 10^{-4} and the local batch size b is 16.

Fréchet Inception Distance (FID) [14] metric is employed to rate the images' quality. FID quantifies the dissimilarity between a reference distribution and a distribution of generated samples by utilizing statistical properties such as mean vectors and covariance matrices extracted through a pretrained Inception V3 model [15]. Lower FID values indicate higher image quality.

The experimentation began with centralized training, where we established a baseline. Initially, the plan was to train for a total of 20 epochs. However, it became apparent that there was minimal improvement in image quality beyond the 10th epoch. To optimize the utilization of computational resources and time, we decided to employ the results obtained at the 10th epoch as the baseline. Given the extensive time and computational resources required for sampling, the FID score is calculated by using a sample size of 64. The resulting FID score for the centralized model after 10 epochs of training was 17.96.

Subsequently, this article delves into experimentation concerning the selection of both the number of clients and the number of local train steps. Initially, when the number of local steps (denoted as E) was set to 2, various experiments were conducted with different numbers of clients, specifically C = 2, 4, and 8. The outcomes of this experimentation are illustrated in Figure 2 below.

DOI: 10.54254/2755-2721/52/20241136

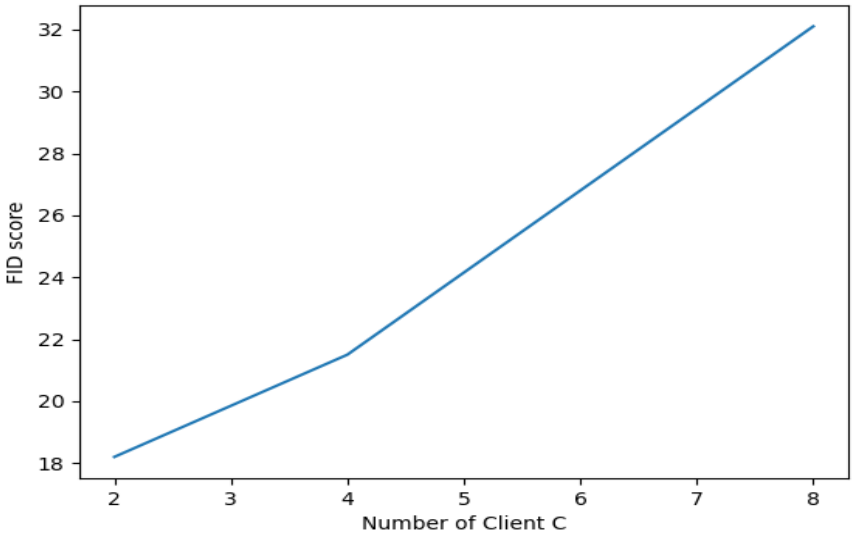

Figure 2. The FID score trained with different number of Clients C (E=2) (Photo/Picture credit: Original).

As depicted in Figure 2, the performance of the global model decreases as the number of clients increases. For instance, with only 2 clients, the FID score stands at 18.2, a value closely resembling the centralized baseline. However, as the number of clients escalates to 8, the FID score reaches approximately 32.

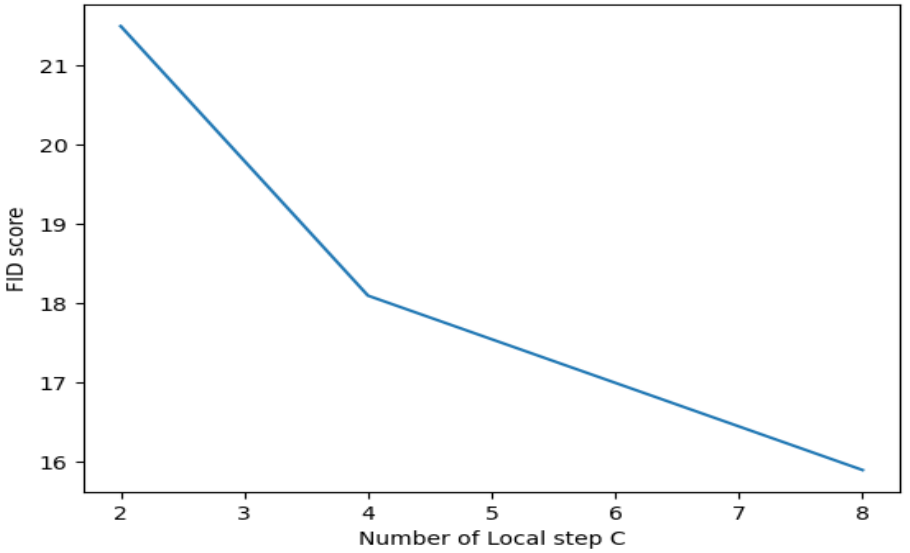

Figure 3. The FID score trained with different number of local step E (C=4) (Photo/Picture credit: Original).

In addition to exploring the impact of the number of clients, this paper conducted an additional experiment to assess the influence of the number of local steps. As illustrated in Figure 3, it becomes evident that increasing the local step (E) leads to improved performance in the training results of the global model. However, it also leads to an extended local training time. Striking a balance between local training duration and performance, the study ultimately settled on E=4 and C=4. With C = 4 and E = 4,

DOI: 10.54254/2755-2721/52/20241136

the model was ultimately trained to achieve an FID score of 18.1, closely resembling the results of the centralized model. The outcomes of the federated diffusion model sampling are shown in Figure 4.

Figure 4. The generated sample of federated diffusion model (C=4, E=4 and T =10) (Photo/Picture credit: Original).

3.2. Federated LoRA diffusion fine-tuning experimental details

Subsequently, this paper conducts the federated fine-tuning experiment, employing a dataset consisting of 16 photos of Cavachon dogs, each with an image size of 512 x 512 pixels. Figure 5 below illustrates a few examples from the dataset. The setting of number of clients is 4, each client has 4 images for local training. The global training round is 200. For LoRA fine-tuning the low-rank dimensions setting is 8.

Figure 5. Overview of the Cavachon dog dataset (Photo/Picture credit: Original).

The base pre-trained diffusion model is employed to generate images associated with the prompts " A photo of little dog", " A photo of little dog in a bucket", " A photo of little dog running", and " A photo of little dog swimming". Figure 6 illustrates the generation results of base model.

Figure 6. Results of base model before fine-tuning (Photo/Picture credit: Original).

Figure 7 illustrates the outcomes of the generation process following the fine-tuning of the model using federated LoRA. As depicted in Figure 7, after undergoing the federated LoRA fine-tuning procedure, the model has successfully adapted to the new task. The generated image samples now closely resemble the input reference images. Moreover, the volume of parameters exchanged between the client and server during each communication round remains remarkably low at just 3.1 MB (Megabytes), in stark contrast to the substantial 433MB parameter size of the entire U-net model. Federated LoRA not only ensures the quality of image generation, but also improves the efficiency of each communication round.

Figure 7. Results of base model after federated LoRA fine-tuning (Photo/Picture credit: Original).

DOI: 10.54254/2755-2721/52/20241136

4. Conclusion

This paper explores the potential of application of federated learning algorithms in diffusion training as well as fine-tuning. By experimenting with federated diffusion pre-training, the model obtained by centralised training and federated training can be similar in terms of the quality of image generation. However, the choice of local steps and number of clients can have a huge impact on the result of the model. The choice of local step will be different for different amounts of data. For federated fine-tuning, the combination of LoRA and federated learning not only ensures the quality and fidelity of the generated images, but also significantly reduces the consumption and time of each communication. At the same time federated learning ensures data privacy and avoids large data transfers.

References

[1] Rombach R Blattmann A Lorenz D Esser P and Ommer B 2022 High-resolution image synthesis with latent diffusion models In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition pp 10684-10695

[2] Saharia C Chan W Saxena S Li L Whang J Denton E L ... Norouzi M 2022 Photorealistic text-to-image diffusion models with deep language understanding Advances in Neural Information Processing Systems 35 36479-36494

[3] Ramesh A Pavlov M Goh G Gray S Voss C Radford A ... Sutskever I 2021 Zero-shot text-to-image generation In International Conference on Machine Learning (pp 8821-8831) PMLR

[4] Brown T Mann B Ryder N Subbiah M Kaplan J D Dhariwal P ... Amodei D 2020 Language models are few-shot learners Advances in neural information processing systems 33 1877-1901

[5] Borji A 2022 Generated faces in the wild: Quantitative comparison of stable diffusion, midjourney and dall-e 2 arXiv preprint arXiv:2210.00586.

[6] Malandrino D Scarano V 2013 Privacy leakage on the Web: Diffusion and countermeasures. Computer Networks 57(14) 2833-2855

[7] McMahan B Moore E Ramage D Hampson S y Arcas B A 2017 Communication-efficient learning of deep networks from decentralized data In Artificial intelligence and statistics (pp 1273-1282) PMLR

[8] Ho J Jain A Abbeel P 2020 Denoising diffusion probabilistic models Advances in neural information processing systems 33 6840-6851

[9] Ruiz N Li Y Jampani V Pritch Y Rubinstein M Aberman K 2023 Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp 22500-22510)

[10] Ronneberger O Fischer P & Brox T 2015 U-net: Convolutional networks for biomedical image segmentation In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference Munich Germany October 5-9 2015 Proceedings Part III 18 (pp 234-241) Springer International Publishing

[11] Hu E J Shen Y Wallis P Allen-Zhu Z Li Y Wang S ... & Chen W 2021 Lora: Low-rank adaptation of large language models arXiv preprint arXiv:2106.09685

[12] Pytorch 2023 https://pytorch.org/

[13] Liu Z Luo P Wang X Tang X 2015 Deep learning face attributes in the wild In Proceedings of the IEEE international conference on computer vision (pp 3730-3738)

[14] Heusel M Ramsauer H Unterthiner T Nessler B Hochreiter S 2017 Gans trained by a two time-scale update rule converge to a local nash equilibrium Advances in neural information processing systems 30

[15] Szegedy C Vanhoucke V Ioffe S Shlens J Wojna Z 2016 Rethinking the inception architecture for computer vision In Proceedings of the IEEE conference on computer vision and pattern recognition (pp 2818-2826)