Auditing Privacy Defenses in Federated Learning via Generative Gradient Leakage

CVPR 2022

Zhuohang Li, Jiaxin Zhang, Luyang Liu, Jian Liu

A Paper Report

Visual Computing (CS6450)

Submitted by

M Anand Krishna

(CS22MTECH14003)

Department of

Computer Science and Engineering

Indian Institute of Technology

Hyderabad

May 2023

ABSTRACT

The Federated Learning (FL) framework enhances privacy in distributed learning by allowing clients to collaborate on learning tasks under a central server's guidance without sharing sensitive data. However, private information can still be exposed through shared gradient data. To mitigate privacy risks, strategies like adding noise or compressing gradients have been proposed.

In this study, the authors demonstrate that private training data can still be leaked through a new form of leakage called Generative Gradient Leakage (GGL). Their method utilizes the latent space of generative adversarial networks (GAN) from public image datasets to compensate for information loss during gradient degradation, unlike methods that only rely on gradient data.

The authors explore gradient-free optimization techniques, such as evolution strategies and Bayesian optimization, to address nonlinearity caused by the gradient operator and GAN model. Their approach demonstrates superior image reconstruction from gradients compared to gradient-based optimizers. The authors intend for their method to serve as a benchmark for evaluating privacy leakage and inspiring the development of robust defense mechanisms.

Contents

Chapter 1

Introduction

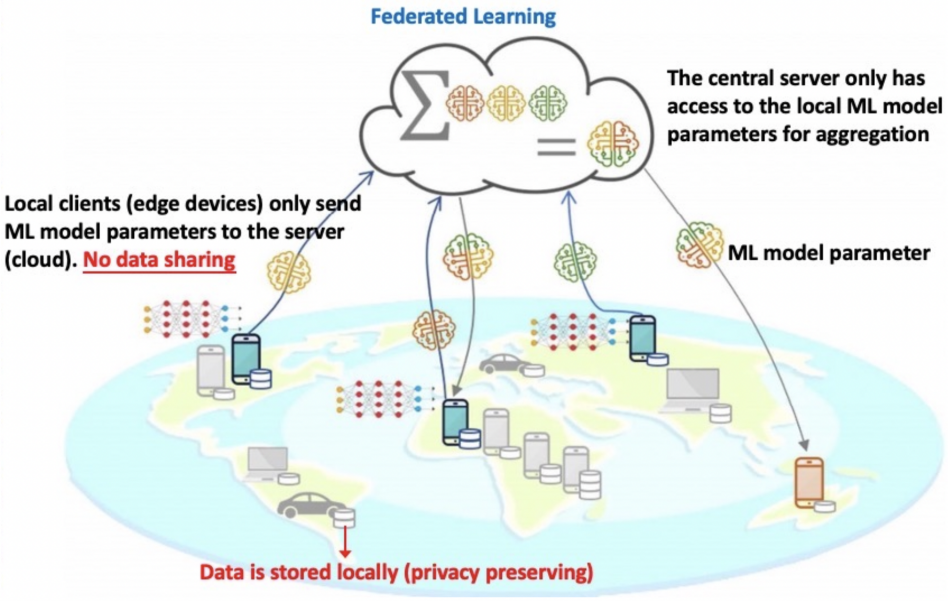
This chapter provides an overview of Federated Learning, a collaborative machine learning technique that allows multiple clients to learn a shared model without exposing their private data.

1.1 Federated Learning

Collaborative learning has become increasingly important in the field of machine learning, especially as the amount of data generated by individuals and organizations continues to grow at an exponential rate. Collaborative learning techniques allow multiple participants to share their data and knowledge to build more accurate models. However, traditional collaborative learning approaches often face significant challenges in terms of data privacy and security, which has led to the development of innovative techniques such as Federated Learning. Federated Learning is a machine learning paradigm where multiple clients, each having their own private data, collaborate to learn a shared model under the coordination of a central server. In traditional machine learning, data is collected and stored in a centralized location where the model is trained. However, this approach poses significant privacy concerns, as sensitive information may be exposed.

In Federated Learning, the data remains on the clients' devices, and only the model parameters are shared with the central server. This allows for the creation of more robust and secure models that can be trained without exposing private data.

The central server initializes the shared model and distributes it to the participating clients. Each client trains the model on its local data and sends the updated model parameters back to the central server. The server then aggregates these updates and applies them to the shared model. This process is repeated iteratively until the shared model converges to an acceptable accuracy.



Federated Learning (FL) has shown great potential for use in mobile computing and telemedicine, enabling collaborative machine learning without compromising data privacy.

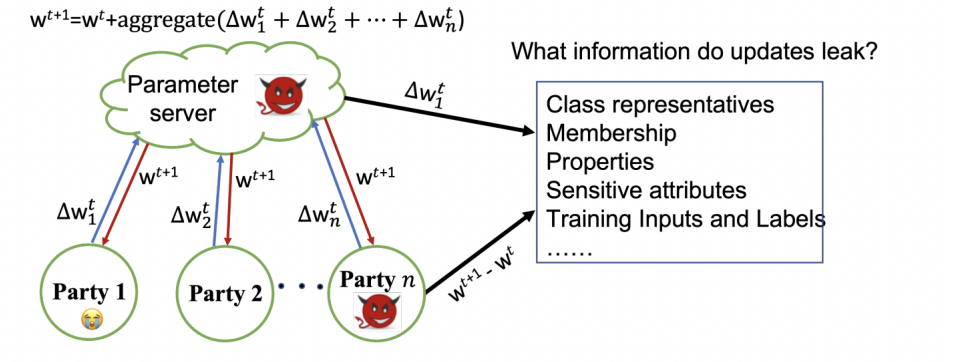
In mobile computing, FL can be used for tasks such as text prediction in Google Gboard. Instead of sending users' typing data to a central server, which could raise privacy concerns, FL allows the model to be trained locally on the device and only sends the model updates to the central server. This approach allows the model to be improved without exposing any sensitive user data.

In telemedicine, FL can be used for multi-institutional collaborations to learn medical diagnosis models without sharing patient's private data. Medical data is highly sensitive, and the traditional approach of collecting and sharing data for machine learning purposes is not feasible due to privacy regulations. FL can solve this problem by enabling hospitals and research institutions to collaborate on model development without sharing patient data. Each institution trains the model locally on their own data, and only the model updates are shared with the central server, ensuring that patient privacy is maintained.

1.2 Privacy Risks in FL

Despite the fact that Federated Learning (FL) only shares model parameters or gradient information from clients, studies have found that private information can still be extracted from the exchanged gradients. Although FL is designed to ensure data minimization and protect privacy, recent studies have shown that in certain cases, sensitive information can still be leaked through the shared gradients. This highlights

the importance of incorporating additional privacy-enhancing techniques such as differential privacy and secure aggregation to mitigate the risk of privacy breaches in FL.



1.3 How to prevent privacy leakage? - Currently Available Privacy Defenses

Cryptographic techniques such as Secure Multi-party Computation and Homomorphic Encryption have been proposed as potential solutions for securing sensitive data in various applications. However, recent research has highlighted the potential vulnerability of these solutions to inference attacks over the output.

To address this issue, researchers have proposed Gradient Degradation-based solutions such as Differential Privacy and Gradient Sparsification.

Differential Privacy is a privacy-preserving technique that aims to protect sensitive data while allowing for its use in data analysis. It involves the use of randomised techniques to distort the gradients before sharing them with a central server. The gradients are perturbed with random noise, which ensures that any information that can be inferred from the output is limited. By distorting the gradients, the technique provides an added layer of privacy protection.

However, there is a trade-off between privacy and utility. This means that as the level of privacy protection increases, the usefulness of the data may be compromised. In other words, the more noise that is added to the data, the less accurate the output will be. This trade-off is a challenge that researchers must consider when implementing Differential Privacy.

Gradient Sparsification is a technique that aims to reduce the amount of data that needs to be shared while preserving the overall accuracy of the model. It involves compressing the gradients by pruning small values. The idea behind this technique is that only the most important information needs to be transmitted to the central server, and small values can be discarded. By pruning small values, the technique

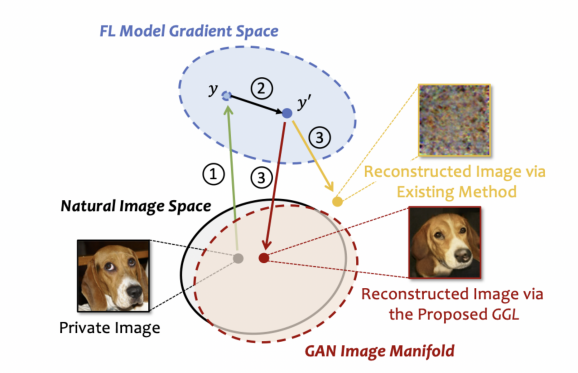


Figure 1.1: Illustration of data leakage via gradient: (1) Client computes gradients on its private data; (2) Client applies defense to degrade the computed gradients y; (3) Adversary attempts to reconstruct the private image from the shared gradients y'.

reduces the amount of data that needs to be transmitted, which can be beneficial in situations where bandwidth or latency is a concern.

However, Gradient Sparsification is not without limitations. The selection of the threshold value for pruning can be a challenging task, and there may be cases where important information is lost during the pruning process. Moreover, the accuracy of the model can be affected if too many values are pruned. Therefore, it is essential to carefully evaluate the potential benefits and limitations of Gradient Sparsification before implementing it in a specific application.

In conclusion, the vulnerability of crypto-based solutions to inference attacks over the output is a challenge that researchers are actively working to address. Gradient Degradation-based solutions such as Differential Privacy and Gradient Sparsification are promising techniques that can be used to secure sensitive data in various applications. While Differential Privacy provides an added layer of privacy protection, there is a trade-off between privacy and utility. On the other hand, Gradient Sparsification can reduce the amount of data that needs to be transmitted while preserving the overall accuracy of the model, but the selection of the threshold value can be challenging. Researchers and practitioners must carefully evaluate the available options and select the most appropriate technique for their specific use case, considering the trade-offs between privacy and utility.

1.4 Are these defenses sufficient?

The author of this work has studied a challenging scenario in which clients implement local defenses prior to sharing the gradients. The focus of the research is to reconstruct high-fidelity images from the shared degraded gradients. See the Figure 1.1

This work demonstrates the feasibility of recovering high-fidelity images from shared gradients, even under certain defense settings, by introducing a new type of leakage called Generative Gradient Leakage (GGL). The method presented in this study leverages the manifold of the generative adversarial network (GAN), which is learned from a large public image dataset, as prior information to provide a good approximation of the natural image space. By minimizing the gradient matching loss in the GAN image manifold, the method can find images that are highly similar to the client's private training data with high quality.

Chapter 2

Literature review

This Chapter provides an idea of the problem statement and related work

2.1 Privacy Leakage via Gradient

The investigations into privacy leakage in Federated Learning (FL) have their origins in membership inference attacks. In these attacks, a malicious analyst infers whether a particular data sample was part of the training set. Additionally, researchers have identified that exchanged model updates can be exploited to infer unintended private information, such as identifying certain input attributes. Further studies have shown that it is feasible to recover class-level information, or even client-level data representatives, which are prototypical samples of the private training set, through generative modeling.

2.2 Data Reconstruction Attacks

Data Reconstruction Attacks pose a severe privacy threat, wherein attackers can restore a client's private data samples by matching the exchanged gradients with the optimal input and label pair. Zhu et al. demonstrated this attack, while a follow-up work proposed a method for extracting the label information. However, these methods have limited applicability and can only be used for shallow networks trained with low-resolution images. Geiping et al. extended this attack to more realistic scenarios and successfully restored high-resolution ImageNet-level data from deeper networks (e.g., ResNet). Yin et al. achieved image batch reconstruction using the strong prior encoded in batch normalization statistics. However, the current research on data reconstruction attacks often assumes an ideal setting and does not consider any additional privacy-preserving measures or defenses, which contradicts industrial practices.

2.3 Privacy Preservation in FL

Privacy preservation in Federated Learning (FL) is a crucial area of research. Researchers have explored two broad categories of approaches to achieve this: cryptography-based and gradient-degradation-based methods. Cryptography-based methods rely on secure multi-party computation (MPC) that allows parties to jointly compute the output of a function over their private inputs, revealing only the intended output to the parties. MPC can be implemented using custom protocols, homomorphic encryption or secret sharing . However, MPC alone may not be sufficient to resist inference attacks on the output .

Gradient-degradation-based approaches, on the other hand, aim to limit the amount of sensitive information leakage by sharing degraded gradients. Differential privacy (DP) is a standard method to quantify and limit privacy disclosure about individual users. DP can be applied either at the server-side (central DP) or the client-side (local DP), where local DP is preferred as it does not require clients to trust anyone. Local DP uses a randomized mechanism to distort gradients before sharing them with the server. However, adding too much noise to the gradients can reduce the utility of the trained models.

In addition to DP, researchers have found that gradient compression/sparsification can also help prevent information leakage from the gradients. Recent work by Sun et al. has identified data representation leakage from gradients as the root cause of privacy leakage in FL. They propose a defense mechanism named Soteria, which computes gradients based on perturbed data representations. Soteria achieves a certifiable level of robustness while maintaining good model utility.

Chapter 3

Motivation

This chapter deals with background and motivation

3.1 Problem statement

The reconstruction of a training image from its gradients can be represented as a non-linear inverse problem. Specifically, given a training image $x \in \mathbb{R}^d$ and its corresponding gradients $y \in \mathbb{R}^m$, the goal is to obtain an estimate of x from y. This can be formulated as:

$$y = F(x)$$

where $F(x)$ is the forward operator that calculates the gradients of the loss function, provided with label c and federated learning model f_θ. Here, $\nabla_\theta L(f_\theta(x), c)$ is the gradient of the loss function with respect to the model parameters θ.

When defense mechanisms are applied at the client's side, the problem becomes more challenging. The gradients are first subjected to a lossy transformation $T(\cdot)$, such as sparsification, which reduces the size of the gradients by discarding some of the information. This results in the modified gradients $\tilde{y} = T(y)$. Additionally, differential privacy (DP) is used to add noise to the gradients to ensure privacy. The resulting noisy gradients can be represented as:

$$y = T(F(x)) + \varepsilon$$

where ε represents the additive noise introduced by the DP mechanism. DP is a technique used to protect the privacy of the data by adding controlled noise to the gradients, making it more difficult for an adversary to infer sensitive information about the data.

The challenge of reconstructing the training image from the noisy and modified gradients requires a careful balance between privacy and accuracy. The design of the lossy transformation and the DP mechanism should be optimized to preserve the privacy of the data while maintaining the accuracy of the reconstruction.

3.2 Current Approach

The current approach to solving the inverse problem of reconstructing a training image from its gradients involves using image priors in a penalty form. Specifically, existing methods aim to find the optimal solution $\mathbf{x}^*$ by minimizing the objective function:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathbb{R}^d}{\operatorname{argmin}} \mathcal{D}(\mathbf{y}, F(\mathbf{x})) + \lambda\omega(\mathbf{x}),$$

where $\mathcal{D}(\cdot)$ is a distance metric and $\omega(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ is the standard image prior (e.g., total variation).

The goal of this approach is to incorporate prior knowledge about the image in the form of the penalty term $\omega(\mathbf{x})$ to improve the accuracy of the reconstruction.

3.3 Limitations of Current Approach

The current approach of using image priors in a penalty form to reconstruct training images from their gradients has certain limitations. One major limitation is that this approach is only effective for reconstructing images from the actual gradients, which may not be readily available or may be prohibitively expensive to obtain. Moreover, when the gradients are low-fidelity and noisy, this approach may suffer from the limited identification ability of hand-crafted priors, resulting in false solutions that are not valid natural images.

The use of hand-crafted priors limits the flexibility of the reconstruction approach, as it assumes that the prior knowledge about the image can be captured by a fixed set of priors. This may not always be the case, especially for complex images, where the prior information may be difficult to model using a fixed set of priors. As a result, the reconstructed images may not accurately reflect the true characteristics of the original training images, especially in the presence of noise or low-quality gradients. Therefore, more advanced techniques that are capable of capturing complex image priors may be required to improve the accuracy of the reconstruction process.

Chapter 4

Methodology

This chapter deals with paper methodology - "Generative gradient leakage" along with experiment results and implementation details

4.1 Generative Gradient Leakage

The use of a generative model trained on public image datasets as a learned natural image prior can help improve the quality of the reconstructed image. Specifically, a well-trained generator $G(\cdot)$ can be leveraged to ensure that the reconstructed image is of good image quality.

Given a well-trained generator, the problem of reconstructing a training image from its gradients can be formulated as follows:

$$\mathbf{z}^* = \underset{\mathbf{z}\in\mathbb{R}^k}{\operatorname{argmin}} \underbrace{\mathcal{D}(\mathbf{y}, \mathcal{T}(F(G(\mathbf{z}))))}_{\text{gradient matching loss}} + \lambda \underbrace{\mathcal{R}(G;\mathbf{z})}_{\text{regularization}} ,$$

where $\mathbf{z} \in \mathbb{R}^k$ is the latent space of the generative model, and $\mathcal{R}(G;\mathbf{z})$ is a regularization term that penalizes latent vectors which deviate from the prior distribution.

The objective function consists of two terms. The first term, $\mathcal{D}(\mathbf{y}, \mathcal{T}(F(G(\mathbf{z}))))$, is the gradient matching loss, which measures the similarity between the reconstructed gradients and the actual gradients. The second term, $\mathcal{R}(G;\mathbf{z})$, is the regularization term that encourages the latent vectors to conform to the prior distribution of the generative model.

By leveraging a well-trained generative model as a learned natural image prior, this approach can improve the quality of the reconstructed image and produce more realistic results. Moreover, the use of a generative model allows for greater flexibility in capturing the complex image priors, which may be difficult to model using hand-crafted priors. An overview of the proposed method is provided in Figure 4.1.

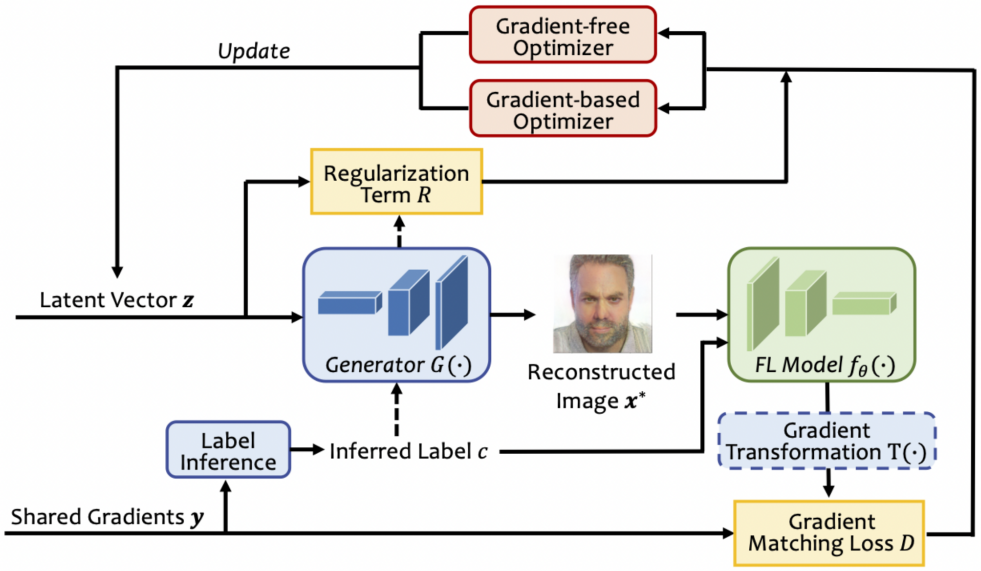


Figure 4.1: GGL

4.1.1 Label Inference attack

The Label Inference attack allows an adversary to infer the ground truth label associated with a client's private image using the shared gradients. This is achieved by adopting an analytical method.

For FL models performing classification tasks over n classes, the i-th entry of the gradients with respect to the weights of the final fully-connected (FC) classification layer (denoted as $\nabla \mathbf{W}^i_{FC}$) can be computed using

$$\nabla \mathbf{W}^i_{FC} = \frac{\partial \mathcal{L}(f_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{c})}{\partial z_i} \times \frac{\partial z_i}{\partial \mathbf{W}^i_{FC}}$$

The first term of this equation represents the partial derivative of the loss function with respect to the output of the final FC layer, while the second term represents the post-activation outputs of the previous layer.

To retrieve the ground truth label, the index of the negative entry of $\nabla_{W_i} F_C$ is identified. For networks trained with cross-entropy loss on one-hot labels, the first term will be negative only if $i = c$, where c is the ground truth label. Thus, the inferred label can be used to evaluate the FL model training loss $L(f_\theta(x), c)$.

In the case of conditional GANs, the inferred label can also be used as the class condition. It is important to note that this attack can compromise the privacy of the client's private image, as it allows the adversary to infer sensitive information about the image. Therefore, appropriate defenses must be implemented to prevent such attacks in FL systems.

4.1.2 Estimating the gradient transformation

Adversaries can attempt to minimize the effects of these defenses by applying similar transformations while assessing the loss of reconstructed images. Even though the transformation process on the client's end is not directly known to the adversary, they can still estimate the transformation parameters by observing gradients. Following are the gardient degarding methods that are estimated from:

Gradient Clipping: This technique involves limiting the magnitude of gradients in the training process to prevent them from becoming too large, which can lead to unstable learning. By setting a predefined threshold, the gradients are scaled down if their magnitude exceeds the threshold. This ensures that each client's contribution is constrained, helping maintain privacy and avoiding issues like exploding gradients in the optimization process.

Gradient Sparsification: This method involves making gradients sparse by setting small gradient values to zero, effectively pruning them. The purpose of gradient sparsification is two-fold: it reduces the communication bandwidth required in distributed training systems by transmitting only non-zero gradient values, and it defends against gradient leakage attacks by making it harder for adversaries to exploit gradient information. A pruning rate is used to determine the threshold for setting gradient values to zero, and the process is usually applied layer-wise in a neural network.

Representation Perturbation: This technique aims to protect privacy by introducing perturbations into the learned representations of data in a specific fully-connected layer of a neural network (the defended layer). By carefully selecting and perturbing specific elements of the representation, the technique maximizes the reconstruction error, making it difficult for adversaries to infer sensitive information from the data. A pruning rate is used to determine which elements of the representation should be perturbed, and gradients are computed based on the perturbed representation. This can be seen as applying a mask only to the gradients of the defended layer, making it harder for adversaries to reverse-engineer the information.

4.1.3 Gradient Matching Loss

The primary term in the objective function directs the solver to discover images that bear contextual similarity to the client's private training images within the generator's latent space. This is achieved by minimizing the distance between the transformed gradients of the generated images (denoted by ˜y) and the observed gradients (denoted by y). We investigate the following distance metrics to compute the gradient matching loss:

- Squared l2 norm

- Cosine Distance: The cosine distance is magnitude-invariant, meaning it only considers the direction of the gradients, not their magnitudes. This property makes it equivalent to optimizing the Euclidean distance between two normalized gradient vectors.

4.1.4 Regularization Term

Relying solely on gradient matching loss during optimization may lead to latent vectors that stray from the generator's latent distribution, potentially resulting in unrealistic images with significant artifacts. To circumvent this issue, we examine the following loss functions to regularize the latent vector during the optimization process:

- **KL-based regularization**: With µi and i representing the element-wise mean and standard deviation, this regularization term aims to minimize the Kullback-Leibler divergence (KLD) between the latent distribution and the standard Gaussian distribution N(0, I).

- **Norm-based regularization**: This regularization approach penalizes latent vectors that deviate significantly from the prior distribution, helping to ensure that the generated images remain realistic and artifact-free.

4.1.5 Optimization strategies

The primary challenge in tackling the target inverse problem lies in its highly non-convex and non-linear nature. Current data reconstruction attacks predominantly rely on gradient-based optimizers, such as L-BFGS and Adam. However, gradient-based optimizers come with their own set of issues:

Dependency on initialization: The outcome of gradient-based optimization is heavily influenced by the choice of initialization. The starting point of optimization significantly impacts the convergence to a particular local minimum or saddle point.

Suboptimal convergence for complex models: In the context of complex models, there is a considerable risk that the cost function will converge to a suboptimal minimum instead of the global minimum. This can lead to subpar performance or even failure to learn the underlying patterns in the data.

Addressing these challenges requires further research and development of optimization techniques that are more robust to the non-convexity and non-linearity of the target inverse problem.

The author investigates two gradient-free optimization strategies to overcome the challenges associated with gradient-based optimizers, particularly when dealing with non-convex and non-linear problems. These strategies are:

Bayesian Optimization (BO): Bayesian optimization is a global optimization technique that works efficiently on expensive, noisy, and high-dimensional functions. It builds a probabilistic model, usually a Gaussian Process, to represent the objective function. The algorithm then iteratively samples points based on an acquisition function, which balances the trade-off between exploration (searching unexplored regions) and exploitation (refining the search around the current best solution). As a result, it can efficiently find the global minimum of the objective function, even when gradients are not available or when the function is non-convex and non-linear.

Covariance Matrix Adaptation Evolution Strategy (CMA-ES): CMA-ES is a derivative-free optimization algorithm inspired by evolutionary principles. It iteratively samples candidate solutions from a multivariate Gaussian distribution, which is adapted throughout the optimization process. The algorithm updates both the mean and covariance matrix of the distribution based on the fitness of sampled solutions. This adaptation allows CMA-ES to navigate complex search spaces and converge to the global minimum, even in high-dimensional, non-convex, and non-linear problems.

By employing these two gradient-free optimization strategies, the author aims to address the limitations of gradient-based optimizers, such as the dependency on initialization and suboptimal convergence in complex models.

4.2 Experiment and results

4.2.1 Experiment setup

The method is evaluated on two Federated Learning (FL) tasks: (1) Binary gender classification on the CelebFaces attributes dataset (CelebA) with 32×32 image size, and (2) 1000-class image classification on the ImageNet ILSVRC 2012 dataset with 224×224 image size. The FL model used for both tasks is based on the ResNet18 architecture with randomly initialized weights. The client performs one local step with a batch size of 1 to compute the gradient

4.2.2 Datasets used in the paper

1. CelebA: The CelebA dataset is a large-scale face attributes dataset collected from the internet, containing more than 200,000 celebrity images. These images

showcase a diverse range of ethnicities, ages, and facial features, making it an ideal dataset for various computer vision tasks related to facial analysis. Each image is annotated with 40 binary attributes, such as gender, hair color, and the presence of glasses. The CelebA dataset has become a popular benchmark for tasks like face recognition, facial attribute prediction, and training generative models, including Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs).

2. ImageNet: ImageNet is an extensive image database containing over 14 million images spanning 20,000 categories. The dataset was designed to support research in computer vision, particularly for tasks like object recognition and classification. The images in ImageNet are organized according to the WordNet hierarchy, with each image being annotated with one or more WordNet synsets (i.e., groups of synonyms that describe a concept). ImageNet has played a crucial role in the development of deep learning and has become a standard benchmark dataset for various computer vision tasks, including object recognition, object detection, and image classification.

Both CelebA and ImageNet have significantly contributed to the advancement of computer vision research by providing diverse and challenging datasets for training and evaluating machine learning models.

4.2.3 Results in paper

The impact of different optimizers on the reconstruction results was studied in the paper. Random images were selected from the CelebA and ImageNet dataset for the reconstruction process, and the experiment was repeated by varying the random seed. The number of updates for Adam, BO, and CMA-ES was set to 2500, 1000, and 800, respectively. The results were summarized in Figure 4.2, and the reconstruction samples were visualized in Figure 4.3. It was observed that the gradient-based and gradient-free optimizers demonstrated similar performance on the CelebA dataset, with Adam outperforming slightly in terms of visual and statistical metrics. However, on the ImageNet dataset, the gradient-based Adam optimizer failed to recover any useful information from the gradients except for the class label. Moreover, its reconstruction results were found to be highly dependent on the initialization. In contrast, the gradient-free optimizers (BO and CMA-ES) were found to be more effective on the ImageNet dataset.

Performance metrics used to evaluate are:

Dataset	Metric	Adam		BO		CMA-ES	
		Mean	Std.	Mean	Std.	Mean	Std.
CelebA	MSE-I ↓	**0.0427**	0.0025	0.0813	0.0131	0.0708	**0.0008**
	PSNR ↑	**13.6965**	0.2593	10.9455	0.6816	11.4989	**0.0533**
	LPIPS ↓	**0.1435**	**0.0083**	0.2162	0.0328	0.2136	0.0133
	MSE-R ↓	**0.0003**	**0.0001**	0.0012	0.0003	0.0015	0.0022
ImageNet	MSE-I ↓	0.5918	0.1955	**0.2648**	0.0181	0.2667	**0.0119**
	PSNR ↑	2.4433	1.3565	**5.7783**	0.2992	5.7420	**0.1988**
	LPIPS ↓	0.7983	0.0280	0.6166	0.0590	**0.5736**	**0.0209**
	MSE-R ↓	0.1051	0.0703	0.0035	0.0005	**0.0018**	**0.0002**

Figure 4.2

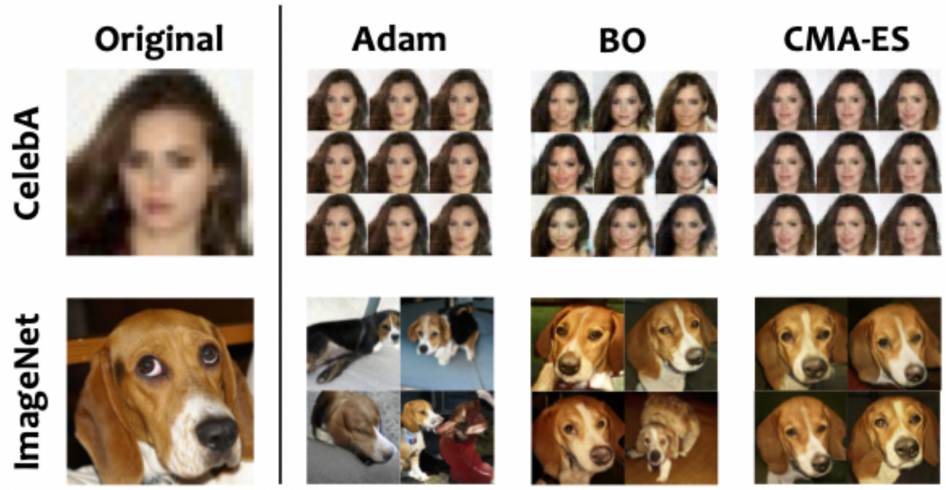


Figure 4.3: The images on the right are the reconstruction samples produced by three types of optimizers with different random seeds

- MSE-I: the mean squared error (MSE) between the generated and true gradients of the model at iteration i.
- PSNR: the peak signal-to-noise ratio (PSNR) between the generated and true gradients of the model at iteration i.
- LPIPS: the learned perceptual image patch similarity (LPIPS) between the generated and true gradients of the model at iteration i
- MSE-R: the MSE between the generated and true gradients of the model over all iterations

4.2.4 My implementation results

I have taken CelebA dataset to perform GGL reconstruction attack.

See the performance metrics results in Figure 4.1

Table 4.1: Performance Metrics

	Observed Values
MSE ↓	0.068
PSNR ↑	5.8
LPIPS ↓	0.53

Figure 4.4: The Input (ie: Private Data)

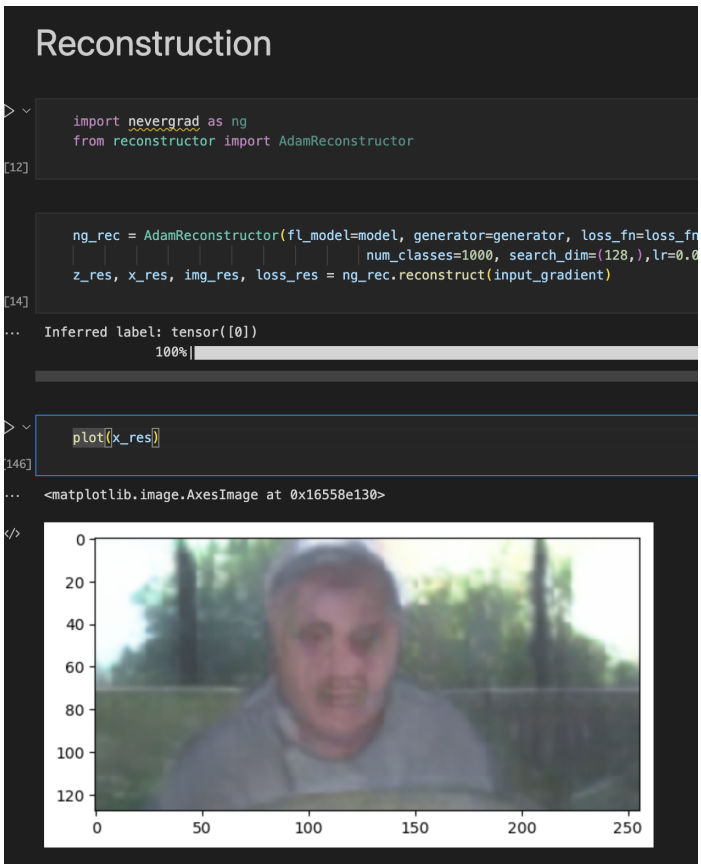


Figure 4.5: Reconstructed image through GGL attack under Adam optimization

Chapter 5

Novelty

This chapter deals novelty and its implementation

5.1 Developing a new privacy defense technique

The generative gradient leakage attack is a powerful threat to the privacy of machine learning models. It allows attackers to infer sensitive information by analyzing the gradients shared during federated learning. As a result, developing effective privacy defenses has become an important research direction. One promising approach is to degrade the gradients in such a way that attackers are unable to reconstruct them.

Gradient quantization is a defense technique that involves mapping the gradient values to a smaller set of discrete values, thereby reducing the amount of information that can be inferred from the gradients. This technique has not yet been tested against the gradient reconstruction attack known as GGL, which highlights the need for further investigation into its effectiveness.

Developing novel privacy defenses specifically designed to defend against the gradient leakage attack is critical for ensuring the privacy and security of machine learning models. This will involve developing new metrics to quantify the degree of privacy protection offered by different defense techniques and identifying potential weaknesses that can be exploited by attackers.

It is also important to develop a comprehensive understanding of the limitations and trade-offs associated with different privacy defenses. For example, while gradient quantization is a promising technique, it may have negative impacts on model accuracy and convergence speed. There should be a careful balance between the need for privacy protection and the need for model performance.

So defending against the gradient leakage attack is a critical challenge for the machine learning community. Gradient quantization was succesful in defending the attack

5.2 Gradient Quantization

Gradient quantization is a technique that has gained much attention for its ability to enhance the privacy of the gradients used in federated learning. The technique involves reducing the precision of gradients by using fewer bits to represent their values, thereby limiting the amount of information that is transmitted in each round of federated learning. This reduction in information can help protect the privacy of the gradients and prevent potential attacks by malicious adversaries seeking to gain access to sensitive information.

In addition to enhancing privacy, gradient quantization also offers benefits such as a reduction in memory usage and communication costs. This is particularly important in federated learning, where large amounts of data are distributed across multiple devices, and communication can be a significant bottleneck.

5.2.1 The Algorithm/Steps

Algorithm 1 Gradient Quantization

Require: Gradient tensor ∇, number of bits $numbits$
Ensure: Quantized gradient tensor $\tilde{\nabla}$
Determine the number of levels: $numlevels \leftarrow 2^{numbits}$
Calculate the range of gradient values: $minvalue \leftarrow \min(\nabla)$, $maxvalue \leftarrow \max(\nabla)$
Compute the bin width: $binwidth \leftarrow \frac{maxvalue - minvalue}{numlevels}$
for each value v in ∇ **do**
  Quantize the value: $\tilde{v} \leftarrow round\left(\frac{v - minvalue}{binwidth}\right) \times binwidth + minvalue$
end for
Replace the original gradient values: $\tilde{\nabla} \leftarrow \tilde{v} \mid v \in \nabla$
return $\tilde{\nabla}$

Algorithm explained : The gradient quantization algorithm consists of several steps that are designed to reduce the precision of gradients, thus protecting the privacy of federated learning. Firstly, the number of bits for quantization is determined. This determines the number of unique quantization levels that will be used. Next, the minimum and maximum values in the gradient tensor are identified to calculate the range of gradient values. The bin width for uniform quantization is then calculated. This is the range of values that each quantization level will represent.For each value in the gradient tensor, the quantized value is calculated using the bin width and the minimum value. This is done to ensure that the quantized value falls within the correct range of values. The original gradient values are then replaced with their quantized

counterparts in the gradient tensor.The resulting quantized gradient tensor has lower precision than the original gradient tensor, reducing the amount of information that can be inferred from the gradients.

5.2.2 Why it can be good solution to prevent attack?

1. Gradient quantization is a technique that reduces the amount of information that is transmitted in each round of federated learning. This is achieved by compressing the gradients into fewer unique values, which degrades the granularity of the information in the gradients. By doing so, sensitive information is made less accessible to potential attackers who may attempt to reconstruct the original data from the shared gradients. The quantized gradients are now less representative of the original training data, which makes it more difficult for an attacker to accurately reconstruct the data. This is because the quantized gradients are compressed into fewer values, which leads to a loss of information. The reduction in the granularity of the gradients helps protect the privacy of the training data, as it becomes much harder to extract useful information from the shared gradients.

2. Moreover, gradient quantization reduces the communication cost and memory usage during the federated learning process. This is because the quantized gradients require less space to transmit and store than the full precision gradients. The algorithm for gradient quantization involves determining the number of bits for quantization, calculating the range of gradient values, computing the bin width, quantizing the gradient values, and replacing the original gradient values with their quantized counterparts in the gradient tensor.

5.3 Gradient Quantization implementation

In **defense.py** file in GGL folder of the code, one more defense method is added as function called quantize_gradient(input_gradient, num_bits=16). Here num_bits is a hyperparameter. See the Figure 5.1

Also modified the reconstructor.py and Imp.ipynb file as seen the figure 5.2 AND 5.3

To reproduce the implementation result and novelty, run the Imp.ipynb notebook file

```python
def quantize_gradient(input_gradient, num_bits=16):
    """
    Quantize gradients to reduce the number of unique values.

    Args:
    - input_gradient (list of torch.Tensor): the input gradient
    - num_bits (int): number of bits for quantization

    Returns:
    - list of torch.Tensor: quantized gradient
    """
    device = input_gradient[0].device
    gradient = [None] * len(input_gradient)

    # Concatenate the gradients into a single tensor
    grad_tensor = torch.cat([grad.flatten() for grad in input_gradient])
    grad_min = grad_tensor.min().item()
    grad_max = grad_tensor.max().item()

    # Calculate the bin width based on the number of bits
    bin_width = (grad_max - grad_min) / (2 ** num_bits)

    for i in range(len(input_gradient)):
        grad_tensor = input_gradient[i].detach().cpu().numpy()
        quantized_grad = np.round((grad_tensor - grad_min) / bin_width) * bin_width + grad_min
        gradient[i] = torch.Tensor(quantized_grad).to(device)

    return gradient
```

Figure 5.1: Added gradient quantization as one of the defense setting

Figure 5.2

Figure 5.3

Figure 5.4: The Input (ie: Private Data)

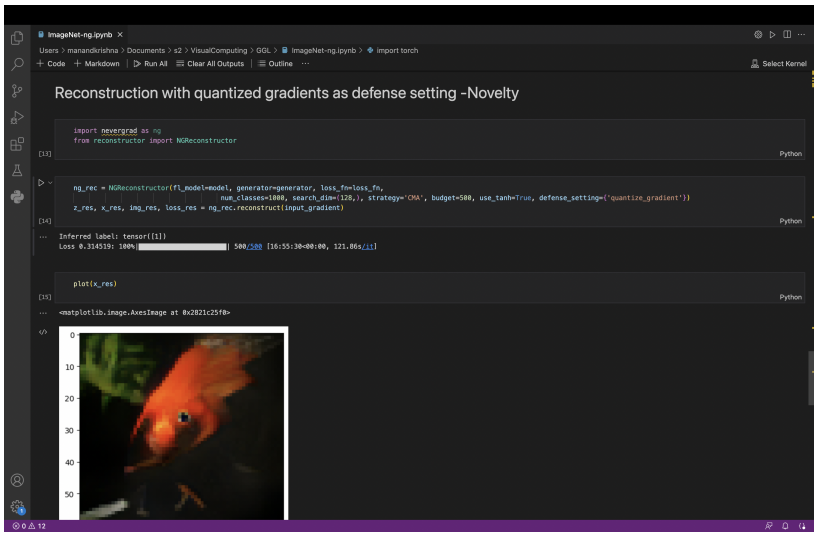


Figure 5.5: The Output (Gave an human face as input, but GGL reconstruction gave meaningless output)

As summary, the use of gradient quantization as a defense mechanism against the gradient leakage attack has proven to be successful. The quantization process effectively reduced the granularity of information in the gradients, making it difficult for an attacker to reconstruct the original data accurately. Our reconstruction attack GGL failed to generate meaningful reconstructions using quantized gradients, demonstrating the efficacy of this defense technique as seen in Figure 5.5. This highlights the importance of incorporating privacy-enhancing techniques in federated learning systems to protect sensitive data from potential attacks.

5.4 Conclusion

GGL is a powerful tool that can handle different types of perturbations and transformations in gradients, making it a robust method. It can effectively maintain its performance even when the gradients are subjected to various types of attacks. This makes GGL an important tool for evaluating the effectiveness of existing privacy defenses that are designed to protect sensitive data from unauthorized access.

Moreover, GGL can be used to reveal information about the original images and evaluate the strength of existing privacy defenses. This can help in identifying the weaknesses of current privacy defenses and inform the design of more robust and effective privacy protection techniques. By utilizing the insights gained from GGL's performance, researchers can develop better defense mechanisms that can provide more comprehensive protection against privacy attacks.

Therefore, GGL can be considered as a valuable asset for researchers working in the field of privacy protection. Its ability to handle various types of perturbations and transformations in gradients, and to evaluate the strength of existing privacy defenses, can help researchers to develop more robust and effective privacy protection techniques. Overall, the use of GGL can lead to significant improvements in privacy protection, making it an essential tool for researchers in the field of data privacy.

Bibliography

Nitin Agrawal, Ali Shahin Shamsabadi, Matt J Kusner, and Adria‘ Gasco n Quotient: two-party secure neural network training and prediction. *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (2019)

[0] [1] Amir Beck and Marc Teboulle "Amir Beck and Marc Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems". *IEEE transactions on image processing* (2009)

[2] Mordido, Keirsbilck and Keller Monte Carlo Gradient Quantization *CVPR* (2020)