On the Effect of Quantization on Deep Leakage from Gradients and Generalization

Arvind S. Menon (354584), Lars C.P.M. Quadvlieg (352130), Sachin Bhadang (385246)
School of Computer and Communication Sciences, EPFL, Switzerland

***Abstract*—In distributed machine learning systems, sharing gradients is a common practice to safeguard the privacy of training data. However, Deep Leakage from Gradients (DLG) by (1) demonstrated that private data could be reconstructed from these shared gradients using an inversion attack. This study explored various defense mechanisms, including noise addition, precision reduction, and gradient compression and sparsification. The findings indicate that although these defenses can mitigate privacy breaches, they often substantially degrade model accuracy, except in the case of gradient compression. Notably, when gradients are compressed with sparsity at thresholds exceeding 20%, the DLG attack is ineffective with minimal impact on model accuracy. Motivated by these findings, we extend this line of research by exploring various quantization techniques to assess their effectiveness in preserving both data privacy and model performance. We show that our proposed quantization methods performs similarly or outperform popular sparsification-based approaches whilst sacrificing little to none performance during training.**

I. Introduction

Distributed machine learning systems have become increasingly popular due to their ability to leverage data from multiple sources to build more robust models. In these systems, sharing gradients instead of raw data has been a common practice aimed at protecting the privacy of the training data. However, recent research demonstrates that even shared gradients are susceptible to privacy breaches. Notably, the work in (1), known as Deep Leakage from Gradients (DLG), shows that it is possible to reconstruct private data from shared gradients using an inversion attack. One such example can be seen in Figure 1.

DLG's findings highlight significant vulnerabilities in the privacy guarantees of distributed learning systems. Their experiments explore various defense strategies such as adding noise, reducing precision, and applying gradient compression and sparsification. The findings suggest that while these mechanisms can prevent inversion attacks, they frequently result in diminished model accuracy. An exception was found in gradient compression using sparsity. Notably, DLG observed that at sparsity thresholds exceeding 20%, the attacks failed to reconstruct private data, and model accuracy was largely preserved (2; 3). However, this method is prone to gradient spiking and hence model divergence.

Building on this foundation, our project aims to further investigate the potential of quantization methods as a means to enhance the privacy of shared gradients while preserving model performance. We hypothesize that different quantization techniques can offer a more balanced trade-off between privacy and accuracy. Overall, we aim to identify strategies that minimize privacy risks without compromising the model's learning efficacy.

Fig. 1: Visualization of DLG on an example image using the gradients obtained from the training process.

The code for the experiments and models discussed in this paper is available at our GitHub repository.

II. Background

A. Improvements on DLG

Since DLG, there has been further research on the same topic. Improved Deep Leakage Gradient (iDLG) (4) exploits the structure of the cross entropy loss to extract the data label with 100% accuracy on the MNIST (5), CIFAR100 (6) and LFW datasets (7), and also shows improved performance on the reconstruction of the input samples for the same 3 datasets over the previous methods. Furthermore, (3) provides a framework that combines iDLG with the Wasserstein Distance loss function in order to perform the reconstruction of the private dataset.

B. Dataset

We evaluate our models on the CIFAR-10 dataset. The CIFAR-10 dataset is a widely-used benchmark dataset in the field of machine learning and computer vision. It consists of 60,000 color images, each with a resolution of 32x32 pixels. The images are equally divided into 10 distinct classes, with each class containing 6,000 images. Similar to CIFAR-10.

C. Models

LeNet (8) is specifically designed for image classification tasks and is well-suited for datasets like CIFAR-10 and

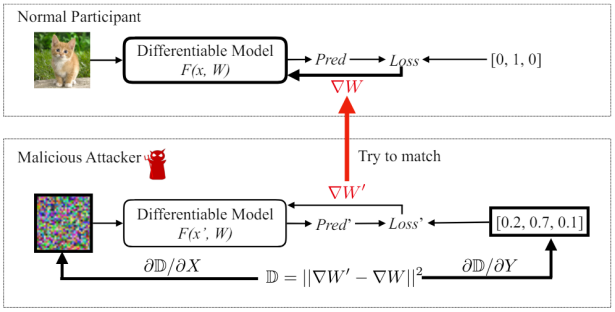


Fig. 2: The overview of the DLG algorithm, taken from (1) Variables to be updated are marked with a bold border. While normal participants calculate ∇W to update parameter using its private training data, the malicious attacker updates its dummy inputs and labels to minimize the gradients distance. When the optimization finishes, the evil user is able to obtain the training set from honest participants.

CIFAR-100 due to its simplicity and efficiency. We utilize this model for the reconstruction of the input data, which are images in our case. The specific model configuration can be found in Appendix A. We also utilize Residual Neural Networks (ResNets) (9) for the image classification models, which we will adopt in our experiments, specifically the ResNet18 configuration available in PyTorch.

III. Methodology

The pipeline of Deep Leakage from Gradients can be seen in Figure 2. We assume that at some point in the training process, we have access to the model $F(\cdot, W)$ with its weights W, and the gradients ∇W that were calculated at that point in training.

Then, you can initialize some dummy variables $x', y' \sim \mathcal{N}(0, 1)$ that we want to become equal to some samples from the training distribution. Since we have access to the true gradient ∇W, we can try to optimize $D_i = \|\nabla W' - \nabla W\|^2$, where $\nabla W'$ is the gradient computed from the dummy variables. We can then compute $\nabla_{x'} D_i$ and backpropagate it to x_i to reconstruct a training sample. This process proceeds iteratively until convergence or divergence.

This algorithm is further described in Appendix A. Note that this optimization requires 2^{nd} order derivatives. This is based on the mild assumption that F is twice differentiable, which holds for the majority of modern machine learning models. In order to gain insight into DLG, we employ a few quantization methods unexplored by the authors of (1) and also additionally plot the effect of sparsification to illustrate it's efficacy in protecting the sensitive data from the DLG inversion attack. The quantization methods employed by us are :

- Uniform Quantization : The method operates by mapping the input values to a predefined number of quantization levels linearly, ensuring a uniform distribution of the quantized values across the range of the gradient tensor.
- Stochastic Rounding: Very similar to uniform quantization, except that it is probabilistically rounded up or down based on its proximity to the nearest quantized value.

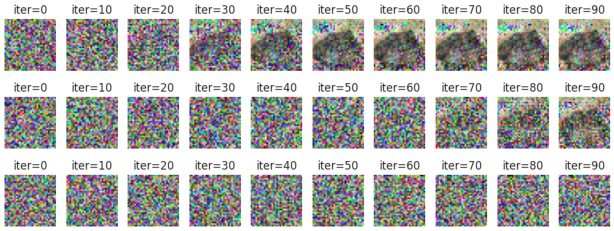


Fig. 3: Illustrating iterative reconstruction of the private data for coarser quantization on the gradient. The figure shows the case for uniform quantization ranging from levels 120, 40, 20 from top to bottom.

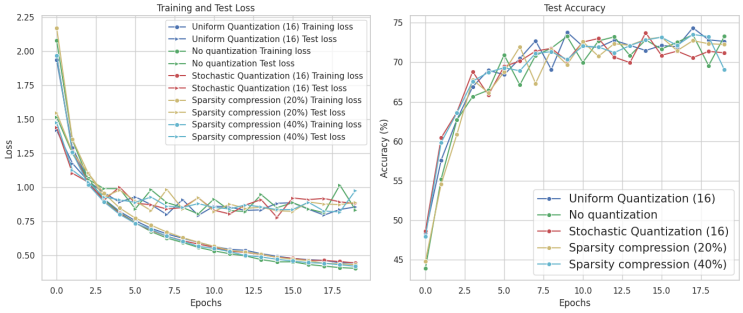


Fig. 4: Training loss, testing loss, and testing accuracy for different training configurations with a ResNet18 model for 22 epochs. Different configurations include quantization, sparsification, and no constraints.

IV. Experiments

We perform several experiments to benchmark the different quantization schemes and sparsification.

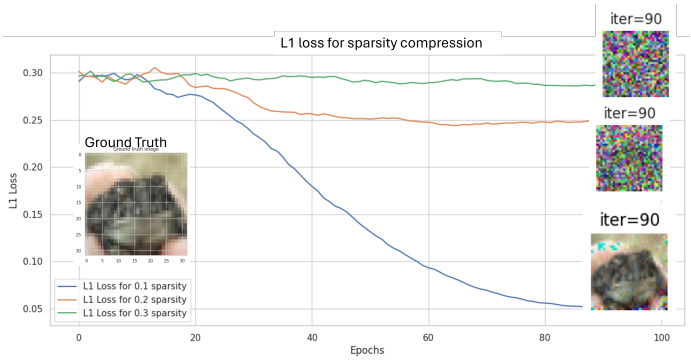
a) Training: We train each ResNet model for 22 epochs with a batch size of 128. Additional training parameters can be found in Appendix B.

b) DLG: When using the DLG algorithm, we always use 100 gradient updates to attempt to reconstruct a sample.

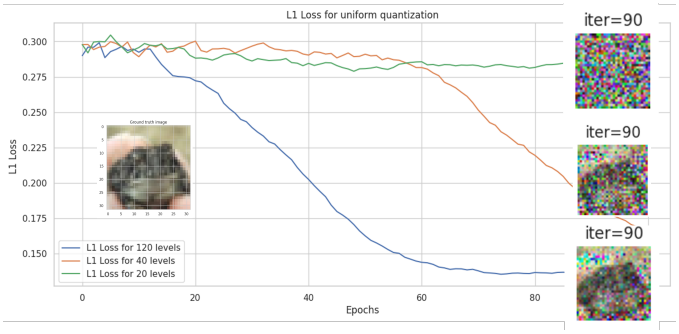
c) Affect of Model Training on DLG : The authors of (1) claimed that the attack can occur at any point during training. To test this assertion, we conducted experiments, the results of which contradicted their claim. Specifically, we examined image reconstruction immediately after randomly initializing the model weights, as well as after running the model for 5 iterations. We The results of these experiments can be seen in Appendix C

V. Results

Initially, we experimented with image reconstruction from sparsity compression and uniform quantization, with different hyperparameters for each configuration. This result is depicted in Figure 5a. As expected, when the sparsity is low, the L1 loss for image reconstruction is also low. This means that the model was successfully able to reconstruct the original training image. This can also be seen directly in the figure. When the sparsity increases to 0.2 or 0.3, the image is no



(a) L1 loss for image reconstruction with LeNet networks for uniform sparsity compression. Different lines correspond to different sparsity thresholds.



(b) L1 loss for image reconstruction with LeNet networks for uniform quantization techniques. Different lines correspond to different quantization levels.

longer reconstructible. A very similar effect presents itself for uniform quantization; a high number of levels corresponds to a good reconstruction, since there is little quantization. However, we also observe that adding quantization makes reconstruction more difficult. For 40 levels, most of the image in Figure 3 is already distorted.

In order to observe this phenomenon to its full effect, we include Figure 3. In this figure, you can also see the attempted data reconstructions for different levels of quantization. For a quantization with 20 levels, no visible reconstruction is possible.

Now that we know that quantization can prevent deep leakage from gradients, we also need to find out if model performance suffers when trained with different modes of quantization. For this purpose, we train several ResNet18 models for image classification. The loss and accuracy curves can be seen in Figure 4. We observe that there does not appear to be a clearly visible disadvantage to compression or sparsification during training, even for large levels of such constraints.

We also perform a quantitative analysis of these different privacy-enhancing methods in Table I. Here, we compute the average and median data reconstruction Mean-Squared-Error (MSE) over 10 different seeds, with the goal of observing which technique would prevent gradient leakage the most. From the table, it becomes clear that 16-level stochastic quantization has the best average and median reconstruction error, which is good. However, even a heavy sparsity-based approach can be outperformed by a uniform quantization. When comparing the accuracy of the models different methods, they all seem very similar. Finally, it also important to note that the average MSE is much larger than the median MSE for the model trained without any special techniques.

VI. Discussion

From the results, we can observe that quantization works as a useful deep gradient leakage prevention technique. It attains high-quality model performance during training, and is able to outperform previously best sparsity-based techniques.

TABLE I: Comparison of Mean Squared Error (MSE) for different privacy-enhancing methods, averaged over 10 seeds, with the corresponding ResNet18 performance for training with that method after 22 epochs.

Method	Average MSE	Median MSE	Accuracy
Stochastic quantization (16 levels)	275.683	252.160	0.737
Uniform quantization (16 levels)	235.810	162.485	0.738
Sparsity = 0.4	3.090	2.957	0.735
Uniform quantization (240 levels)	0.466	0.336	0.737
Sparsity = 0.2	0.948	0.227	0.736
Nothing	110.56	0.0001	0.735

We also note that the DLG algorithm has quite high variance in it's reconstructions. As we saw in the results, DLG sometimes was not able to extract the original training data from the model trained without quantization or sparsification, which should be guaranteed according to (1).

Due to our limited evaluation of 22 epochs with ResNet18 on CIFAR-10, we cannot guarantee the generalization of our results to other domains such as text generations, or more difficult datasets such as CIFAR-100. In the future, we recommend investigation into quantized methods in more computationally expensive domains and datasets.

Furthermore, we strongly believe that the problem of deep leakage of gradients will differ during various stages of model training, and hope this will be investigated in future research. For example, we think that the model does not leak as much information near convergence as at the beginning of training.

VII. Summary

We examine the implications of quantization on privacy and accuracy in distributed machine learning systems, highlighting the vulnerability of shared gradients to privacy breaches through an inversion attack known as Deep Leakage from Gradients (DLG). To combat this, we evaluate various defense mechanisms including gradient quantization and sparsification, observing their effects on model accuracy and privacy.

We find that gradient quantization offers a high-performing defense mechanism against these attacks, providing similar benefits to previously researched sparsifications.

REFERENCES

[1] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *Advances in neural information processing systems*, vol. 32, 2019.

[2] Y. Tsuzuku, H. Imachi, and T. Akiba, "Variance-based gradient compression for efficient distributed deep learning," 2018.

[3] X. He, C. Peng, and W. Tan, "Fast and accurate deep leakage from gradients based on wasserstein distance," *Int. J. Intell. Syst.*, vol. 2023, pp. 1–12, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257711092

[4] B. Zhao, K. R. Mopuri, and H. Bilen, "idlg: Improved deep leakage from gradients," *ArXiv*, vol. abs/2001.02610, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:210064455

[5] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[6] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:18268744

[7] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

APPENDIX

The paper (9) also introduces an algorithmic pipeline for reconstruction the data, which we present in Algorithm 1.

Algorithm 1 Deep Leakage from Gradients

Require: $F(x; W)$: Differentiable machine learning model; W: parameter weights; ∇W: gradients calculated by training data.
Ensure: Private training data x, y

procedure DLG($F, W, \nabla W$)
  $x'_1 \leftarrow \mathcal{N}(0,1)$, $y'_1 \leftarrow \mathcal{N}(0,1)$ ▷ Initialize dummy inputs and labels.
  for $i = 1$ to n **do**
    $\nabla W'_i \leftarrow \frac{\partial \ell(F(x'_i, W_t), y'_i)}{\partial W_t}$ ▷ Compute dummy gradients.
    $D_i \leftarrow \|\nabla W'_i - \nabla W_i\|^2$
    $x'_{i+1} \leftarrow x'_i - \eta \nabla_{x'_i} D_i$, $y'_{i+1} \leftarrow y'_i - \eta \nabla_{y'_i} D_i$ ▷ Update data to match gradients.
  end for
  return x'_{n+1}, y'_{n+1}
end procedure

A. LeNet

For any LeNet architecture that we utilize, we employ the following configuration:

Layer Type	Output Channels	Kernel Size	Padding	Stride
Conv2d	12	5	2.5	2
Activation	-	-	-	-
Conv2d	12	5	2	2
Activation	-	-	-	-
Conv2d	12	5	2	1
Activation	-	-	-	-
Conv2d	12	5	2	1
Activation	-	-	-	-
Linear	Input: 768, Output: 100	-	-	-

TABLE II: Configuration of the LeNet Model

B. ResNet

To train the ResNets, we use a learning rate of 0.1, momentum of 0.9, and weight decay of $5e-4$. As previously mentioned, all ResNets follow the ResNet18 architecture.

C. Image Reconstruction at Different Stages

We test the efficacy of DLG by testing it to reconstruct an image before and after model training and plot the L1 scores of 5 different dummy samples for different quantization methods. We observe that image fidelity is reasonable on reconstruction before training, but fails poorly on training for few iterations.

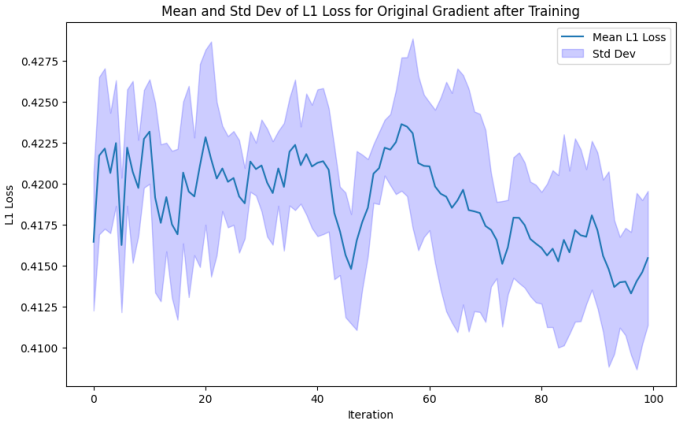


Fig. 6: L1 score of reconstructed images after model training

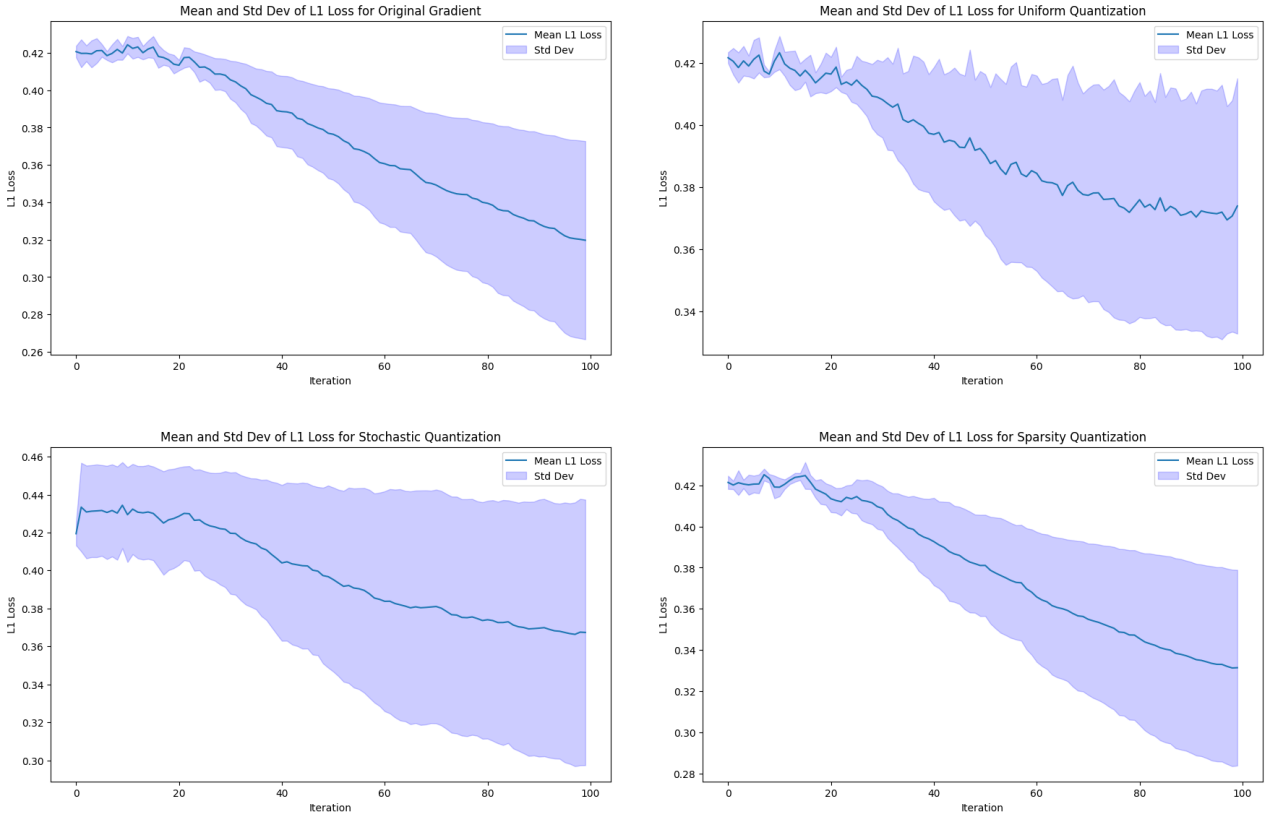


Fig. 7: L1 score for different quantization of reconstructed images before model training