Inverting Gradients - How easy is it to break privacy in federated learning?

Nathan Bigaud

20 décembre 2022

Outline

Overview

Inverting gradients : adversarial data recovery in a realistic distributed learning setting

- **We discuss a paper entitled "Inverting Gradients", by Geiping et al. 2020**, which builds on the recent discovery of vulnerabilities in federated learning systems
- **The paper experiments with the ideas introduced by Zhu et Han 2020** and extends them to more realistic settings, most notably by showing recovery can be achieved despite non-linearities
- **We explore the paper** results and reimplement some of them for testing purpose
- **We also explore a rudimentary noise-addition defense**, as a first step towards proper differential privacy defense

Motivation and approach

Motivation : privacy-preservation in distributed and collaborative learning

- **Distributed training** is becoming ubiquitous as models scale, and trust in those systems is crucial for their adoption, in particular for federated learning, designed to preserve data privacy
- **A trusted system should be resistant to adversarial attacks** from within. But in 2019, Zhu et Han 2020 showed that gradient sharing, presumed mostly safe up to then, could be recovered in a white-box context under simplified assumptions
- **This prompted an active field of research** exploring the possible gradient inversion attacks and available defenses

Our approach

- **Our first goal was to understand the core algorithm** implementation, using standard librairies
- **We then tested our implementation** by exploring the paper results ourself on a few examples, within the limits of our computing resources
- **We finally attempted to experiment with DPSGD** defense (Abadi et al. 2016), with a rudimentary noise addition scheme and pruning

Main contributions

Recall the core algorithm from DLG

The paper improves the DGL algorithm, which uses the classical stochastic gradient updates, applied to the data rather than the network's parameters.

1. **Randomly initialize** a dummy input $\mathbf{x}'$ and label input $\mathbf{y}'$, and derive the 'dummy gradient'
$$\nabla W' = \frac{\partial \ell \left(F \left(\mathbf{x}', W \right), \mathbf{y}' \right)}{\partial W}$$
2. **Iteratively optimize** the dummy gradients w.r.t $\mathbf{x}'$ and $\mathbf{y}'$:
$$\mathbf{x}'^{*}, \mathbf{y}'^{*} = \arg\min_{\mathbf{x}', \mathbf{y}'} \left\| \frac{\partial \ell \left(F \left(\mathbf{x}', W \right), \mathbf{y}' \right)}{\partial W} - \nabla W \right\|^2$$

Contribution (1/2) : theory and algorithm

The first important contribution is to show that access to any fully-connected layer allow for inversion.

- Consider a fully-connected layer followed by a ReLU activation function, i.e. for input $x_l \in \mathbb{R}^{n_l}$, $x_{l+1} = \max\{y_l, 0\}$ for $y_l = A_l x_l$ where the maximum is computed element-wise.
- Now assume we have the additional knowledge of the derivative w.r.t. to the output $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}x_{l+1}}$. Assume $\exists i$ s.t. $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}(x_{l+1})_i} \neq 0$.
- Then the input v can be derived from the knowledge of $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}A_l}$.

The second important contribution is a modification of the lost function, using cosine similarity instead of Euclidian distance :

$$\arg\min_{x \in [0,1]^n} 1 - \frac{\langle \nabla_\theta \mathcal{L}_\theta(x, y), \nabla_\theta \mathcal{L}_\theta(x^*, y) \rangle}{\|\nabla_\theta \mathcal{L}_\theta(x, y)\| \, \|\nabla_\theta \mathcal{L}_\theta(x^*, y)\|} + \alpha \, \mathrm{TV}(x)$$

Contribution (2/2) : stylized facts

- **Images can be recovered** from standard CNN architectures with limited resources
- **Trained vs untrained network** : for trained networks, reconstructions seem to become implicitly biased to typical features of the same, and data augmentation makes the localization of objects more difficult.
- **Network size** : reconstruction quality measurably increase with the number of channels. Yet, the larger network width is also accompanied with an increasing variance of experimental success.

Our experimentations with attacks

Set-up

- **We focus on CIFAR10**, while the authors mainly use ImageNet, and focus on a small number of images
- **We experiment with different networks**, ResNets of different sizes, as well as MobileNetv2
- **We used some simple hyperparamater search** strategy when needed
- **Note** : we don't use PNSR measure as our implementation of the metric seemed to give incoherence results

Experimenting with attacks (1/3) : Influence of network training

(a) Input

(b) Trained network

(c) Trained network (no augmentations)

(d) Partially trained network

(e) Untrained network

(a) Input

(b) Trained network

(c) Trained network (no augmentations)

(d) Partially trained network

(e) Untrained network

Experimenting with attacks (2/3) : Influence of network size

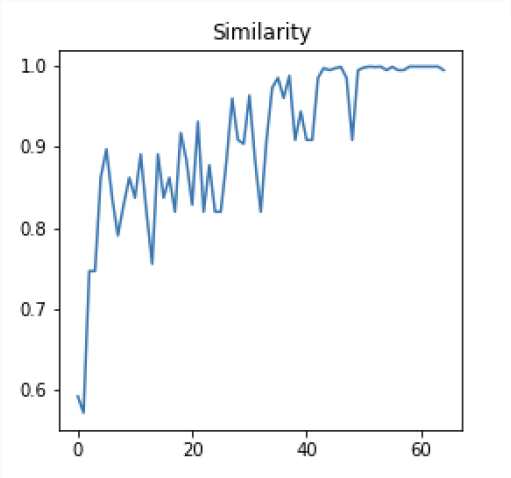
(a) Input **(b)** Trained ResNet18 **(c)** Trained ResNet34 **(d)** Trained ResNet50

(a) Input **(b)** Trained ResNet18 **(c)** Trained ResNet34 **(d)** Trained ResNet50

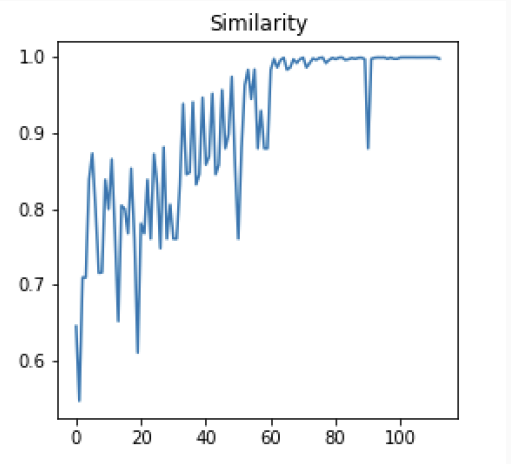
Experimenting with attacks (3/3) : Influence of network type

(a) Input **(b)** Trained ResNet18 **(c)** Trained MobileNetv2 **(d)** Trained MobileNetv2

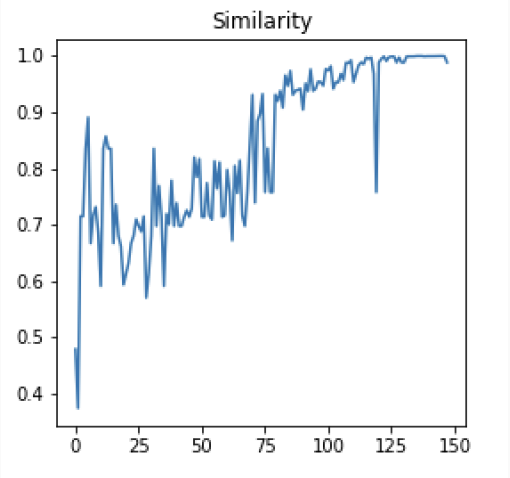
Experimenting with attacks (bonus) : Learning stability



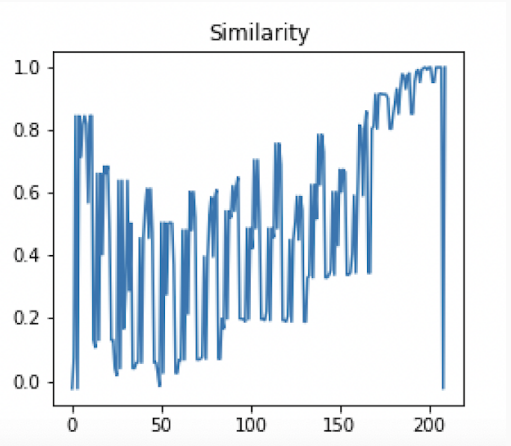
(a) ResNet18centering



(b) ResNet34



(c) ResNet50



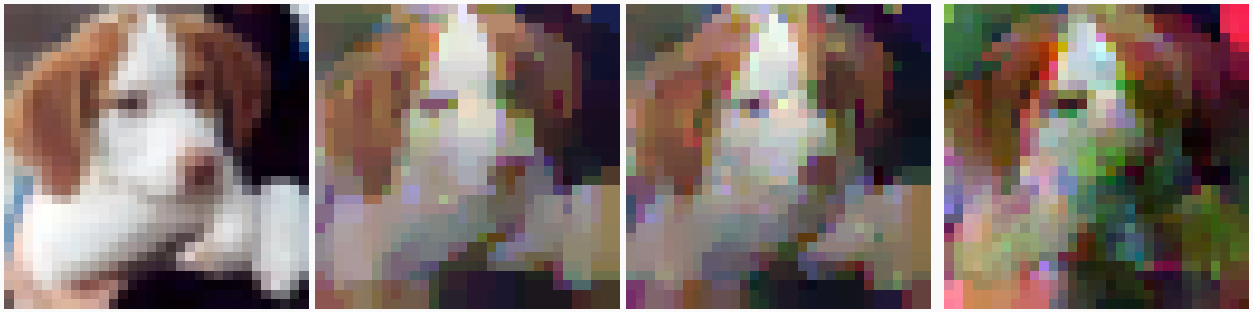
(d) MobileNetv2

Experimenting with pruning and noise-addition

Sensitivity to naive noise addition

(a) Input **(b)** $\sigma^2 = 1e-2$ **(c)** $\sigma^2 = 1e-1$ **(d)** $\sigma^2 = 1$ **(e)** $\sigma^2 = 10$

Sensitivity to pruning



(a) Input **(b)** threshold = 25% **(c)** threshold = 50% **(d)** threshold = 75%

DPSGD for gradient privacy

DP-SGD bounds the sensitivity of the learning process to each individual training example

- By computing per-example gradients $\{g_i\}_{i \in 0..n-1}$ with respect to the loss, for the n model parameters $\{\theta_i\}_{i \in 0..n-1}$, and clipping each per-example gradient to a maximum fixed ℓ_2 norm C.
- Subsequently, to the average of these per example gradients, DP-SGD adds (Gaussian) noise that whose standard deviation σ is proportional to this sensitivity.

Applications to modern federated learning show promising results, see for instance Malekzadeh et al. 2021

Further exploration

Given more time, we would focus on

- Proper implementation of the DPSGD algorithm and systematic exploration of the utility/privacy trade off under inverting gradient attack
- Experimentation with mini-batch and several local steps in the FedAverage algorithm recovery, including exploring unknown batch size
- Explore MobileNet reconstruction limits

References

References


Abadi, Martin et al. (2016). “Deep learning with differential privacy”. In : *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, p. 308-318.


Geiping, Jonas et al. (2020). “Inverting Gradients–How easy is it to break privacy in federated learning ?” In : *arXiv preprint arXiv :2003.14053*.


Malekzadeh, Mohammad et al. (2021). “Dopamine : Differentially private federated learning on medical data”. In : *arXiv preprint arXiv :2101.11693*.


Zhu, Ligeng et Song Han (2020). “Deep leakage from gradients”. In : *Federated learning*. Springer, p. 17-31.

Thank you for your attention!