Building better defense mechanisms in VFL ECSE 4962/6962 -TML

Huzaifa Arif

Ph.D Student

Department of Electrical, Computer, and Systems Engineering

Rensselaer Polytechnic Institute

email: arifh@rpi.edu

April 19, 2022

Vertical Federated Learning - A motivation

- Different medical institutions have some test results of same patient
- Institutions don't share raw data with each other
- The model diagnoses/predicts whether the patient has a certain disease or not

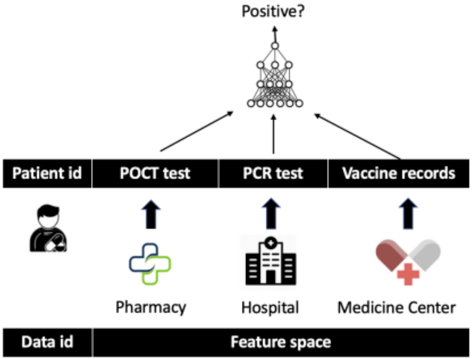


Figure. VFL [Chen et al.(2020)Chen, Jin, Sun, and Yin]

Vertical Federated Learning - Structure

- VFL is the feature partitioning case with the data distributed amongst the clients in such a way that for every user, each client only has a subset of features

$$\mathbf{x}_n = [\mathbf{x}_{1,n}^T, \mathbf{x}_{2,n}^T, \ldots, \mathbf{x}_{M,n}]^T \tag{1}$$

- The clients share $h_m(\theta_m; x_m)$ embeddings with the server and **not the gradients** of whole model
- In our setting, the gradient computation happens locally

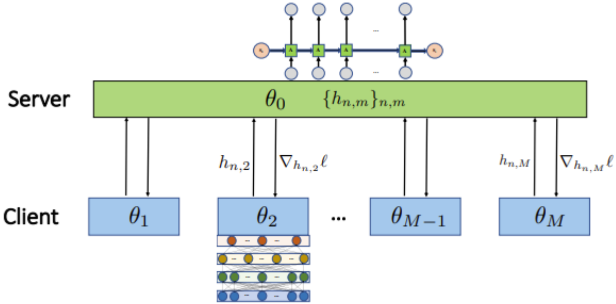


Fig:1b Distributed Model

Objective Function - VFL

$$F(\Theta; \mathbf{X}; \mathbf{y}) := \frac{1}{N}\sum_{i=1}^{N} L(\theta_0, h_1(\theta_1; x_1^i), \ldots, h_M(\theta_M; x_M^i); y^i) \quad (2)$$

Background Work

- Previous works have considered gradient inversion attacks [Zhu et al.(2019)Zhu, Liu, and Han]
- In VFL setting. model inversion attacks were studied first by [Chen et al.(2020)Chen, Jin, Sun, and Yin] CAFE
- This performed leakage attacks on exchanged gradients of model
- But in our setting the gradients are never shared.
- So is distributed model VFL safe ?

Contributions

1. It is shown that inversion is possible if access to **embeddings** and **client/server model** exists
2. It has been shown that **quantization/compression** of these embeddings reduces communication costs without affecting model accuracy by much.I look at effect of compression in model inversion attacks.
3. I propose new **randomized compression** mechanism that makes a DP- model robust to model inversion attacks for shallow networks.

Model Inversion in VFL

- We assume the malicious party has white box access to each of the clients
- The malicious party aims to solve the following optimization problem [Mahendran and Vedaldi(2015)]
- Feature recovery for m^{th} client

$$\mathbf{x}_m^* = \underset{\mathcal{R}^{H\times W\times C}}{\arg\min}\ \mathcal{L}(\Phi(\mathbf{x}_m), h_m(\theta_m; x_m)) + \lambda\mathcal{R}(\mathbf{x}_m) \tag{3}$$

$$\mathcal{L}(\Phi(\mathbf{x}_m), h_m(\theta_m; x_m)) = \|\Phi(\mathbf{x}_m) - h_m(\theta_m; x_m)\|^2 \tag{4}$$

$$E(\mathbf{x}_m) = \mathcal{L}(\Phi(\mathbf{x}_m), h_m(\theta_m; x_m)) + \lambda\mathcal{R}(\mathbf{x}_m) \tag{5}$$

$$\mu_{t+1} \leftarrow m\mu_t - \eta_t \nabla E(\mathbf{x}_m) \tag{6}$$

$$\mathbf{x}_m^{t+1} \leftarrow \mathbf{x}_m^t + \mu^t \tag{7}$$

- The malicious party has access to Φ which is the model of the m^{th} client

Model Inversion in VFL

- Results for reconstruction for different architecture of clients shown.
- This is for typical VFL setting where the malicious party (eg: server) does inversion on embeddings.
- We can see the inversion is strong for shallow networks as the adversary can identify the class !

Depth of Layer: 2

Depth of Layer: 5

Figure. Depth of layer makes reconstruction hard

Compression in VFL

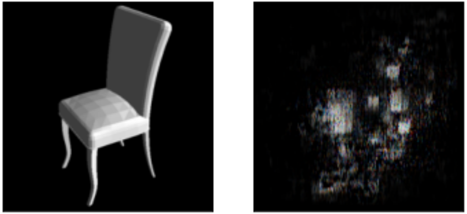
- Model reconstruction is hard for deeper networks but what if cleints have shallow networks,then?
- We investigate the role of compression in model inversion attacks.
- Previous work has shown that compression reduces communication cost without affecting model accuracy. We explore its affect on model inversion!

Objective Function - C-VFL

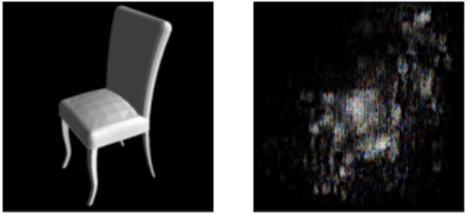
$$F(\Theta; \mathbf{X}; \mathbf{y}) := \frac{1}{N} \sum_{i=1}^{N} L(\theta_0, \mathcal{C}_1(h_1(\theta_1; x_1^i)), \ldots, \mathcal{C}_M(h_M(\theta_M; x_M^i)); y^i) \tag{8}$$

- We note that compression does not change the dimensions of the embeddings !

- We investigate role in worsening reconstruction using two compression schemes : Top-K Sparsification and Scalar Quantization
- Scalar quantization quantizes values into fixed bins
- Top-K controls the factor of sparsification of the embeddings



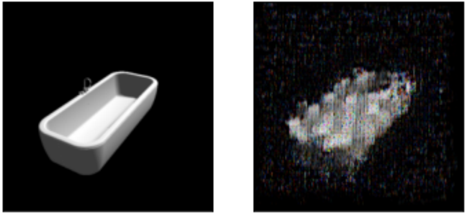
Top-K : 0.12



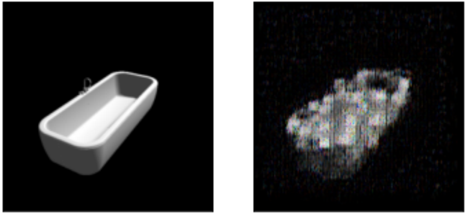
Top-K : 0.23

Figure. Top-K Sparsification

Experimental - Compression in VFL



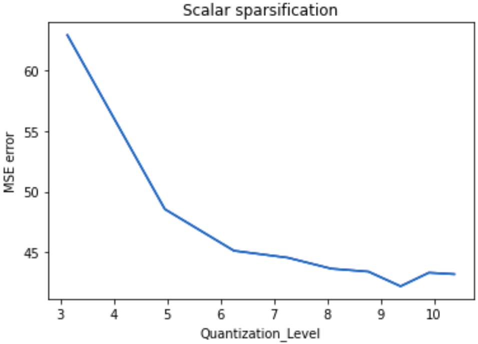
Quantization Level : 2



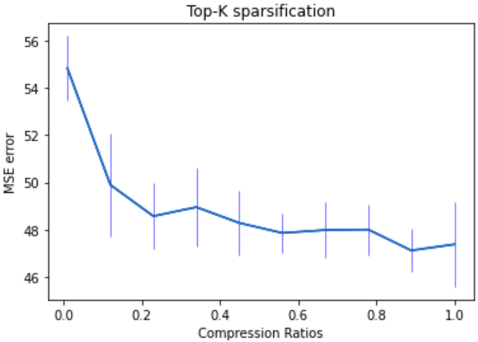
Quantization Level: 8

Figure. Scalar Quantization

Different Compression Mechanisms



Scalar Quantization



Top-K Sparsification

Figure. Compression worsens recovery

Noisy Compression

- We note that low levels of quantization would lead to worse recovery but compromises on model performance
- This makes low quantization level an unfeasible choice
- I propose new **noisy compressed mechanism**

Noisy Compressed Mechanism - Procedure

- $Z_i \sim Bin(N, p)$
- Clients send quantized embeddings and biased binomial noise is added to them : $\widetilde{\mathcal{C}}_m = \mathcal{C}_m(h_1(\theta_1; x_1^i)) + (Z_i)$
- For local gradient computation embeddings are unbiased **locally** $\widetilde{\mathcal{C}}_m - Np$

Why Binomial?

Objective Function - Noisy C-VFL

$$F(\Theta; \mathbf{X}; \mathbf{y}) := \frac{1}{N}\sum_{i=1}^{N} L(\theta_0, \mathcal{C}_m(h_1(\theta_1; x_1^i)) + (Z_i - Np), \ldots, \mathcal{C}_m(h_M(\theta_M; x_M^i)) + (Z_i - Np); y^i) \quad (9)$$

- Binomial noise is chosen as using Gaussian Noise would loose benefits of compression
- Binomial noise also allows us to send biased or unbiased embeddings

Noisy Compression + Binomial Noise results

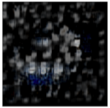
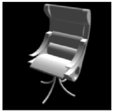
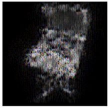
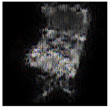
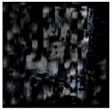
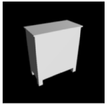
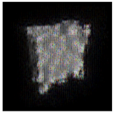
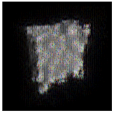
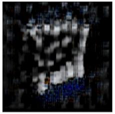
Mechanism	Orignal	No Compression	Compression + Unbiased Binomial Noise	Compression + Biased Binomial Noise
Tub				
Chair				
Table				

Table: Results of Compression

Differential Privacy -Recap

- Equation 11 is the definition (ϵ, δ) Differential Privacy
- Equation 12 is defined as the sensitivity of a mechanism

$$Pr(\mathcal{M}(f(D_1)) \in S) \leq Pr(\mathcal{M}(f(D_2)) \in S) + \delta \tag{10}$$

$$\Delta_q = \max_{(D_1, D_2) \in \mathcal{N}_{\mathcal{D}}} \|f(D_1) - f(D_2)\|_q \tag{11}$$

Noisy Compression is DP

- [Agarwal et al.(2018)Agarwal, Suresh, Yu, Kumar, and McMahan] were the first to propose how a d-dimensional addition of Binomial noise makes a process DP
- Thus if $F(D) = \mathcal{C}_m(h_M(\theta_M; x_M^i)) + (Z_i - Np)$ the embeddings are (ϵ, δ) DP if their sensitivity is bounded.

$$\mathcal{M}_b^{N,p}(f(D)) \triangleq f(D) + (Z - Np) \tag{12}$$

Theoram - Simplified

For any δ,parameter N and p with sensitivity bounds $\Delta_1, \Delta_2, \Delta_{\text{inf}}$ such that

$$Np(1-p) \geq \max(23log(10d/\delta), 2\Delta_{\text{inf}}) \tag{13}$$

and

$$\epsilon(\Delta_1, \Delta_2, \Delta_{\text{inf}}) \tag{14}$$

$\mathcal{M}_b^{N,p} is(\epsilon, \delta)$ Differentially Private (DP)

Key observations from Experimental Analysis

- The setting for the results we have, assumes that each of the clients are adding biased binomial noise to the quantized embeddings
- Adding unbiased noise at the clients does not seem to impact the model inversion attacks (MIA).
- For MIA we assume the malicious party does not have access to the binomial noise distribution
- For MIA we also assume that each of the party has a shallow model

Conclusions/FutureWork

- Shallow networks are prone to deep leakage attacks.
- Our method of noisy embeddings is robust against these attacks
- We propose that adding Binomial Noise by keeping the distribution (or mean) private makes it harder for malicious parties to do inversion
- We have shown that the embeddings are differentially private. We extend that analysis to show the resulting model is DP as well.
- Future work also explores the communication costs of using binomial noise with compression .

REFERENCE

Naman Agarwal, Ananda Theertha Suresh, Felix Xinnan X Yu, Sanjiv Kumar, and Brendan McMahan.
cpsgd: Communication-efficient and differentially-private distributed sgd.
Advances in Neural Information Processing Systems, 31, 2018.

Tianyi Chen, Xiao Jin, Yuejiao Sun, and Wotao Yin.
Vafl: a method of vertical asynchronous federated learning.
arXiv preprint arXiv:2007.06081, 2020.

Aravindh Mahendran and Andrea Vedaldi.
Understanding deep image representations by inverting them.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

Ligeng Zhu, Zhijian Liu, and Song Han.
Deep leakage from gradients.
Advances in Neural Information Processing Systems, 32, 2019.