Integer is Enough: When Vertical Federated Learning Meets Rounding

Pengyu Qiu, Yuwen Pu, Yongchao Liu, Wenyan Liu, Yun Yue, Xiaowei Zhu, Lichun Li, Jinbao Li, Shouling Ji

AAAI 2024

Background

- **Vertical Federated Learning**
  - A promising paradigm of distributed machine learning, especially for collaboration among companies.
- **Challenges in Application**
  - **Privacy:** Data reconstruction attack may reconstruct the raw data from the extracted embeddings.
  - **Efficiency:** Homomorphic Encryption provides encrypted environment, but is computational overhead for float-point numbers.
  - **Security:** Models are sensitive to small perturbation on embeddings.

Intuition

- **Float-point Numbers**
  - may be **redundant** and carries too much information, which should be compressed!!!
- **Binarizing Split Learning**
  - Proposed a binarization way to compress embeddings while maintaining the model's performance loss within an acceptable range.
- **Piece-wise Function**
  - Binarization is a two-pieces function with threshold of 0.
  - Try **Rounding** to balance the security, privacy and efficiency.

Methodology

- **Rounding Layer**
  - Different rounding strategies, adopting rounding to nearest.
  - $Round(x) = [x - 0.5]$.
- **Gradient Estimation**
  - Making up for gradient disappear.
  - Straight-through estimator, $\frac{\partial L}{\partial x} \approx \frac{\partial L}{\partial [x]}$.

Algorithm 1: Rounding in Vertical Federated Learning

Require: clients' bottom models $\{f_i\}_{i=1}^N$, server's top model f_{top}.
Ensure: trained $\{f_i\}_{i=1}^N$, f_{top} for inference.

for each epoch **do**
  for each batch $(\mathbf{X}, \mathbf{Y})$ **do**
    During forward process:
    for At each $Client_i$ **do**
      $\mathbf{Emb}_i \leftarrow f_i(\mathbf{X}_i)$
      $\mathbf{V}_i \leftarrow [\mathbf{Emb}_i]$
      Send $\mathbf{V}_i$ to the server
    end for
    At the server:
    $\mathbf{V} \leftarrow concate(\{\mathbf{V}_i\}_{i=1}^N)$
    $\mathcal{L} \leftarrow cross\_entropy(f_{top}(\mathbf{V}), \mathbf{Y})$
    During backward process:
    At the server:
    for each $\mathbf{V}_i$ **do**
      calculate $\frac{\partial \mathcal{L}}{\partial \mathbf{V}_i}$
      send $\frac{\partial \mathcal{L}}{\partial \mathbf{V}_i}$ to the corresponding client
    end for
    for At each $Client_i$ **do**
      $\frac{\partial \mathcal{L}}{\partial \mathbf{Emb}_i} \leftarrow \frac{\partial \mathcal{L}}{\partial \mathbf{V}_i}$
      update the following parameters of f_i
    end for
  end for
end for

Analysis

- **Computational and Memory Efficiency**
  - Integer computation is computation friendly for HE.
  - Theoretically 4x memory reduction in PyTorch.
- **Error Bound**
  - Certified error bound according to Multivariate Version of Taylor's Theorem .
- **Privacy Analysis**
  - Comparable Differential Privacy protection with Binarization.

Theorem 1 *Given* $\boldsymbol{x} = \boldsymbol{z} + \boldsymbol{r}$*, where* $\boldsymbol{z} \in \mathbb{Z}^d$*, and* $\boldsymbol{r} \in [-\frac{1}{2}, \frac{1}{2}]^d$*. Assume that for a specific class, the top model's prediction can be approximated by a 2-times differential function* $g : \mathbb{R}^d \to \mathbb{R}$*. Then, let* $\Delta = g(\boldsymbol{x}) - g(\boldsymbol{z})$*, we have:*

$$||\Delta||_2 \leq \sum_{||\boldsymbol{\alpha}||_1=1} \frac{1}{2^{\boldsymbol{\alpha}}} ||\frac{D^{\boldsymbol{\alpha}} g(\boldsymbol{z})}{\boldsymbol{\alpha}!}||_2 + \sum_{||\boldsymbol{\beta}||_1=2} \frac{1}{2^{\boldsymbol{\beta}}} \dot{R}_{\boldsymbol{\beta}}(\boldsymbol{z}),$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{N}^d$ *are multi-index notation,* $||\boldsymbol{\alpha}||_1 = \alpha_1 + \cdots + \alpha_d$*, and* $\boldsymbol{\alpha}! = \alpha_1! \cdots \alpha_d!$*;* $D^{\boldsymbol{\alpha}} g = \frac{\partial^{||\boldsymbol{\alpha}||_1} g}{\partial x_1^{\alpha_1} \cdots \partial x_d^{\alpha_d}}$*;* $\dot{R}_{\boldsymbol{\beta}}(\boldsymbol{z}) = \frac{1}{\boldsymbol{\beta}!} \max_{||\boldsymbol{\alpha}||_1 = ||\boldsymbol{\beta}||_1} \max_{\boldsymbol{y} \in \mathcal{B}_{\frac{1}{2}}(\boldsymbol{z})} ||D^{\boldsymbol{\alpha}} g(\boldsymbol{y})||$*, and* $\mathcal{B}_{\frac{1}{2}}(\boldsymbol{z})$ *denotes the norm ball of* $\boldsymbol{z}$ *with the radius of* $\frac{1}{2}$*.*

Error Bound Analysis

Following the derivation in (Pham et al. 2022), we can also add a perturbation to the rounded embeddings for privacy analysis. Let $\mathcal{M}_r$ denote the mechanism of the rounding layer and $\mathcal{M}$ denote the mechanism of adding Laplace noises. Then, we can formulate $\mathcal{M}_r$ as follows:

$$\mathcal{M}_r(\mathbf{x}) = [\mathcal{M}(\mathbf{x})] = [[\mathbf{x}] + Lap(\frac{1}{\epsilon})], \quad (9)$$

where the sensitivity of $f(\mathbf{x}) = [\mathbf{x}]$ is 1. The first equation is because, from the server's perspective, it always receives the embeddings in integer format. The second equation follows the analysis of DP.

If $||Lap(\frac{1}{\epsilon})||_2 < \frac{1}{2}$, then $\mathcal{M}_r(\mathbf{x}) = [\mathbf{x}]$. It means that the rounding operation naturally tolerates a small latent noise. Let $cdf(\cdot)$ denote the cumulative distribution function of Laplace, we have:

$$\begin{aligned} \mathbb{P}[|Lap(\frac{1}{\epsilon})| < \frac{1}{2}] &= [cdf(\frac{1}{2}) - cdf(-\frac{1}{2})] \\ &= 1 - exp(-\frac{\epsilon}{2}). \end{aligned} \quad (10)$$

Differential Privacy Analysis

Settings

- **Datasets**
  - Popular benchmarks: CIFAR10, MNIST, Fashion-MNIST
- **Models**
  - Bottom model: ResNet
  - Top model: MLP
- **Baselines**
  - B-SL: Binarizing Split Learning.
  - Framework without modification.

Main Task's Performance

- **Performance Comparison with Different Settings**
  - **Dimensional Size:** Size of the embeddings.
  - **Feature Ratios:** The proportion of features from one party to the total number.
- **Takeaway**
  - Rounding can better preserve the model's performance with various conditions.

Dataset	Arch.	Dimensional Size				
		d=8	d=16	d=32	d=64	d=128
MNIST	Base	98.41	**98.77**	98.33	98.50	**98.70**
	Binary	97.66	98.57	98.28	97.91	98.34
	Round	**98.43**	98.66	**98.39**	**98.66**	98.31
Fashion	Base	**90.88**	90.87	89.52	**90.75**	90.29
	Binary	90.52	89.69	90.30	89.44	89.30
	Round	90.84	**90.91**	**90.70**	90.66	**90.52**
CIFAR10	Base	**74.59**	75.34	**75.76**	75.15	75.04
	Binary	70.13	70.07	69.41	71.87	70.06
	Round	73.41	**75.67**	74.74	**75.42**	**75.33**

Table 1: Comparison with different dimensional sizes.

Dataset	Arch.	Feature Ratio				
		r=0.1	r=0.2	r=0.3	r=0.4	r=0.5
MNIST	Base	**99.02**	**99.13**	98.96	**98.77**	**98.77**
	Binary	98.85	99.11	98.93	98.61	98.57
	Round	98.83	99.06	**99.13**	98.62	98.66
Fashion	Base	**91.85**	91.51	91.37	90.88	90.87
	Binary	91.35	91.17	91.06	90.95	89.69
	Round	91.67	**91.78**	**91.59**	**91.83**	**90.91**
CIFAR10	Base	**81.79**	**79.82**	76.79	**75.27**	75.34
	Binary	79.32	78.51	75.39	74.01	70.07
	Round	80.92	78.82	**77.19**	74.97	**75.67**

Table 2: Comparison with different feature ratios.

Feature Attribution Consistency

- **Feature Attribution**
  - **Methods:** Integrated gradient, DeepLift, Feature Ablation.
  - **Metrics:** Euclidean Distance, Correlation Distance, Kendall's τ.

Dataset	Arch.	Methods											
		Integrated Gradients				DeepLIFT				Feature Ablation			
		Euc.	Cor.	Kendall's τ		Euc.	Cor.	Kendall's τ		Euc.	Cor.	Kendall's τ	
				Stats	p-Value			Stats	p-Value			Stats	p-Value
MNIST	Binary	0.2646	0.8939	0.0282	0.8344	0.2853	0.8069	0.0887	0.4887	0.2863	0.7793	0.1210	0.3415
	Round	**0.1048**	**0.0849**	**0.5726**	0.0001	**0.1860**	**0.3191**	**0.3710**	0.0025	**0.1886**	**0.2992**	**0.3750**	0.0022
Fashion	Binary	0.2952	0.9072	0.0968	0.4491	0.2853	0.8069	0.0887	0.4887	0.2863	0.7793	0.1210	0.3415
	Round	**0.1552**	**0.1925**	**0.5847**	0.0001	**0.1860**	**0.3191**	**0.3710**	0.0025	**0.1886**	**0.2992**	**0.3750**	0.0022
CIFAR10	Binary	0.3394	1.0851	-0.1290	0.3096	0.2952	0.9408	-0.0605	0.6408	0.3265	0.9771	-0.0847	0.5092
	Round	**0.1702**	**0.1942**	**0.4718**	0.0001	**0.1644**	**0.2926**	**0.3427**	0.0055	**0.1688**	**0.2503**	**0.3790**	0.0020

Table 3: Evaluation results for feature attribution consistency. 'Euc.' represents the Euclidean distance, while 'Cor.' represents the Correlation distance. Smaller distances indicate better results. For Kendall's τ, higher stats indicate better performance.

- **Takeaway**
  - Our results indicate that the rounding architecture preserves consistency better than the binary design for all three methods.

Mitigating Adversarial Attack

- **Adversarial Attack**
  - **Threat Model:** we assume the strongest possible adversary who possesses complete knowledge of the submitted intermediate results and the parameters of the top model.
  - **Method:** Projected Gradient Descent (PGD) Attack, which is a standard white-box adversarial attack.

- **Attack Success Rate Reduction**
  - Rounding operation demonstrates stronger ability to mitigate adversarial attacks than the baselines and the binary architecture.

Dataset	Threshold ω	Step Size s	Accuracy			Preserved Accuracy			Attack Success Rate		
			Base	Binary	Round	Base	Binary	Round	Base	Binary	Round
MNIST	1	0.1	97%	**98%**	97%	15%	98%	**97%**	85%	0	**0**
		1.0	97%	**98%**	97%	14%	22%	**56%**	74%	64%	**43%**
	2	0.1	97%	**98%**	97%	0	98%	**97%**	100%	0	**0**
		1.0	97%	**98%**	97%	0	0	**0**	100%	100%	**96%**
Fashion	1	0.1	89%	83%	**92%**	73%	83%	**92%**	16%	0	**0**
		1.0	89%	83%	**92%**	77%	40%	**83%**	15%	42%	**10%**
	2	0.1	89%	83%	**92%**	12%	83%	**92%**	86%	0	**0**
		1.0	89%	83%	**92%**	12%	0	**14%**	**82%**	94%	83%
CIFAR10	1	0.1	**79%**	69%	77%	22%	69%	**77%**	76%	0	**0**
		1.0	**79%**	69%	77%	28%	9%	**59%**	66%	85%	**25%**
	2	0.1	**79%**	69%	77%	2%	69%	**77%**	98%	0	**0**
		1.0	**79%**	69%	77%	3%	0	**9%**	95%	64%	**43%**

Table 4: Attack success rate evaluation with different combinations of threshold and step size.

Mitigating Adversarial Attack

- **Certified Robust Radius**
  - **Method:** To account for generalization, we use randomized smoothing to compute the certified robustness radius for samples, which is independent of any specific model.
- **Takeaway**
  - Experimental results demonstrate that the rounding operation enlarges the radius of robustness around each x.

Theorem 2 *Let $f : \mathbb{R}^d \to \mathcal{Y}$ be any deterministic or random function, and let $\boldsymbol{\xi} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$. Let g be defined as the smoothed classifier. Suppose $y_A \in \mathcal{Y}$ and $\underline{p_A}$, $\overline{p_B} \in [0, 1]$ satisfy:*

$$\mathbb{P}(f(\boldsymbol{x} + \boldsymbol{\xi}) = y_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{y \neq y_A} \mathbb{P}(f(\boldsymbol{x} + \boldsymbol{\xi}) = y).$$

Then, $g(\boldsymbol{x} + \boldsymbol{\xi}) = y_A$ for all $||\boldsymbol{\xi}||_2 < \mathcal{R}$, where

$$\mathcal{R} = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})).$$

Dataset	Architecture					
	Base		Binary		Round	
	Mean	Std.	Mean	Std.	Mean	Std.
MNIST	2.20	0.70	2.15	0.64	**2.96**	1.15
Fashion	4.11	1.81	1.66	0.68	**4.28**	1.85
CIFAR10	1.92	1.70	1.12	0.81	**3.01**	2.17

Table 5: Certified robust radius.

Conclusion

- **Introduction of Novel Architecture**
  - The paper proposes a new architecture to address challenges, including computational overhead, privacy protection, and security concerns from adversarial attacks, in VFL.
- **Theoretical Analysis of Rounding Layer**
  - Computation efficiency and memory reduction.
  - Rounding error bounds.
  - Privacy protection from a Differential Privacy (DP) perspective.
- **Empirical Studies**
  - Preserves the model's performance.
  - Maintains consistency with the original framework's interpretation.
  - Mitigates adversarial attacks.