Compressed Federated Learning Based on Adaptive Local Differential Privacy

Yinbin Miao\textsuperscript{1}, Rongpeng Xie\textsuperscript{1}, Xinghua Li\textsuperscript{1}, Ximeng Liu\textsuperscript{2}, Zhuo Ma\textsuperscript{1},
and Robert H. Deng\textsuperscript{3}

\textsuperscript{1}Xidian University, China; \textsuperscript{2}Fuzhou University, China; \textsuperscript{3}Singapore Management University, Singapore
Federated Learning
FEDERATED LEARNING

- **Step-I**: Server distributes global model
- **Step-II**: Clients train local models
- **Step-III**: Clients upload local models and server aggregates the results
Privacy disclosure of clients’ training data due to plaintext model parameters or gradients
ISSUES

Privacy disclosure of clients’ training data due to plaintext model parameters

\[ w_{t-1} \rightarrow w_t \rightarrow w_{t-1} \rightarrow \cdots \rightarrow w_{t-1} \rightarrow w_{t} \rightarrow \cdots \rightarrow w_{t} \rightarrow w_{t-1} \]
The Curse of Dimensionality of DNN
Local Differential Privacy (LDP)

Let $\mathcal{M}$ be a randomized perturbation mechanism, for any pair input $x$ and $z$ in $\mathcal{D}$ and any output $Y$ of $\mathcal{M}$, $\mathcal{M}$ satisfies $\varepsilon$-LDP such that

$$Pr[\mathcal{M}(x) = Y] \leq e^\varepsilon \cdot Pr[\mathcal{M}(z) = Y],$$

where $\mathcal{D}$ is a dataset and $\varepsilon$ is the privacy budget of $\mathcal{M}$.

$w^* = w + \delta$, where $\delta \sim \mathcal{N}(0, \sigma^2)$
Compressive Sensing (CS)

Measurement original weight
matrix $\Phi \in \mathbb{R}^{m \times n}$ $x \in \mathbb{R}^n$

Sparsity orthonormal basis matrix $\Psi \in \mathbb{R}^{n \times n}$

Compressed weight $y \in \mathbb{R}^m$

$\mathcal{C}(x, m) = y = \Phi x = \Phi \Psi s = \Theta s$

$\sum_{i=1}^{N} \mathcal{C}(x_i, m) = \mathcal{C} \left( \sum_{i=1}^{N} x_i, m \right)$

$\mathcal{D}(\sum_{i=1}^{N} \mathcal{C}(x_i, m)) \approx \sum_{i=1}^{N} x_i$
OUR SCHEME

Technical Overview:

1. Local Training
2. Compressive sensing
3. Differential privacy
4. Aggregation
5. Reconstruction
6. Global Model
7. Dataset

Local Model 1

Local Model K

Global Model
Local Compressive Sensing:

**Algorithm 2**: Local Compressive Sensing

**Input**: the trained local model $w_t^k$; the total layers $L$; a certain layer $l$; compression ratio $CR$

Client$_k(w^k_t)$:

1. $c_t^k = w_t^k$
2. **for** each $l \in L$ in $c_t^k$ **do**
3. $n_l = \text{len}(c_{t,l}^k)$ // the number of weights for this layer
4. $m_l = n_l \times CR$
5. $c_{t,l}^k = C(c_{t,l}^k, m_l)$ // including DCT and interception
6. **end for**
7. **Return**: $c_t^k$
Adaptive Local Differential Privacy:

- The weight ranges \([c_l - r_l, c_l + r_l]\) of different layers \(l \in [1, L]\) are different, where \(c_l\) is the range center and \(r_l\) is the range radius.
Adaptive Local Differential Privacy:

\[ w^* = w + \delta = \mathcal{M}(c_l, r_l) \]

Layer-level adaptability

\[ c_l = \frac{(\max_l + \min_l)}{2}, \quad r_l = \max_l - c_l \]

Finer-grained

\[ \text{offset: } \mu = w - c_l \leq r_l \]

Weight-level adaptability

\[ w^* = \mathcal{M}(c_l, \mu) \]

Even if they are weights of the same layer, their offsets will be different.
Adaptive Local Differential Privacy:

\[ M(w) = w^* = \begin{cases} 
    c_l + \mu \cdot \frac{e^\varepsilon + 1}{e^\varepsilon - 1}, & \text{with probability } \frac{e^\varepsilon - 1}{2e^\varepsilon} \\
    c_l + \mu \cdot \frac{e^\varepsilon}{e^\varepsilon + 1}, & \text{with probability } \frac{e^\varepsilon + 1}{2e^\varepsilon}
\end{cases} \]
EXPERIMENTS

- **SETUP:**
  - Datasets: MNIST, Fashion-MNIST
  - Model: CNN(2 convolutional layers + 1 fully connected layer)
  - Super-parameters:

<table>
<thead>
<tr>
<th></th>
<th>50/100/200</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Epochs E</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Number of clients</strong></td>
<td>10</td>
</tr>
<tr>
<td><strong>Learning rate</strong></td>
<td>0.1</td>
</tr>
<tr>
<td><strong>Compression Ratio (CR)</strong></td>
<td>1/0.5/0.1/0.05</td>
</tr>
<tr>
<td><strong>Privacy budget</strong></td>
<td>$+\infty/1/2$</td>
</tr>
</tbody>
</table>

- Runtime environment: Pytorch 1.10.0, Numpy 1.21.5, a single CPU @ 3.30 GHz, 16.0GB RAM
Analysis of Adaptive Perturbation:

Fig. 1. Apply $r_l$ in MNIST

Fig. 2. Apply $r_l$ in Fashion-MNIST
Analysis of Adaptive Perturbation:

Fig. 3. Apply $\mu$ in MNIST
EXPERIMENTS

Analysis of Adaptive Perturbation:

Fig. 4. Apply $\mu$ in Fashion-MNIST
Analysis of Compression Ratio:

- **Fig. 5. MNIST**
  - CR: 1, 0.5, 0.1, 0.05
  - Accuracy: ε=+∞, ε=1, ε=2
  - Accuracy values: 95.5, 95.3, 95.8, 95.5, 94.6, 94.9, 95.1

- **Fig. 6. Fashion-MNIST**
  - CR: 1, 0.5, 0.1, 0.05
  - Accuracy: ε=+∞, ε=1, ε=2
  - Accuracy values: 83.8, 81.7, 84.7, 84.6, 83.0, 82.7, 80.7, 81.1
### Traffic and Running Time:

<table>
<thead>
<tr>
<th>Privacy budget $\varepsilon$</th>
<th>$\varepsilon = +\infty$</th>
<th>$\varepsilon = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Traffic(MB)</td>
<td>Runtime(mins)</td>
</tr>
<tr>
<td><strong>Metrics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Dataset</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>12.69</td>
<td>13.76</td>
</tr>
<tr>
<td>0.5</td>
<td>6.35</td>
<td>16.79</td>
</tr>
<tr>
<td>0.1</td>
<td>1.27</td>
<td>52.26</td>
</tr>
<tr>
<td>0.05</td>
<td>0.63</td>
<td>72.84</td>
</tr>
<tr>
<td><strong>Dataset</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Dataset</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>12.69</td>
<td>27.93</td>
</tr>
<tr>
<td>0.5</td>
<td>6.35</td>
<td>33.55</td>
</tr>
<tr>
<td>0.1</td>
<td>1.27</td>
<td>70.35</td>
</tr>
<tr>
<td>0.05</td>
<td>0.63</td>
<td>72.51</td>
</tr>
</tbody>
</table>
CONCLUSION

- We use the compressive sensing to compress the local model, which reduces not only the size of the model but also the amount of noises.
- We apply adaptive Local Differential Privacy to add controllable noises for protecting data privacy and ensuring high model performance.
- Our experiments demonstrate that our scheme improves the accuracy of the model with lower privacy budget, and reduces the communication overhead by 95% at most.
Thank you!

Q&A