Revealing and Protecting Labels in Distributed Training

Trung Dang**, Om Thakkar*, Swaroop Ramaswamy†, Rajiv Mathews†, Peter Chin*, Françoise Beaufays†
* Boston University
† Google LLC
‡ Work performed while at Google
Problem

- In distributed training (e.g., Federated Learning), raw data never leaves user’s devices.
  - Model updates (e.g., gradients) do
- Evidences of data leakage from gradients [Zhu et al.’ 19, Geiping et al.’ 20]
  - Discrete labels remain difficult to reveal

From: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html
Prior Works

- Gradients Matching [Zhu et al’19]:

  - Gradients sent from user’s device → Minimize → Gradients computed from a dummy input

- Discrete labels cannot be reconstructed by minimizing this loss

- Prior works that reconstruct labels:
  - Optimize “smooth” labels [Zhu et al’19]
  - Find the set of labels based on signs of gradients (requires non-negative activation functions) [Yin et al.’ 21]

→ Expanding Yin et al.’21, we want to find a method that works for any architecture
RLG: Looking at Projection Layer

We explore leakages for models that predict labels using a softmax cross-entropy loss.
Looking into the gradient:
\[ \nabla W = A^T \nabla O \]

**Projection Layer**

- **W**: weight matrix
- **A**: projection input
- **O**: projection output
- (S: length of output sequence)

**Embedding dim (d)**

**Vocabulary Size (V)**

We know this

What if we knew this?
A Further Look Into $\nabla O$

Softmax Output (softmax($O$))

Cross Entropy Loss

Minimize

Increase

Decrease

Gradient ($\nabla O$)

One-hot matrix

$S$

$V$

0.6 0.2 0 0 ... 0.1
0.1 0 0 0 ... 0.8
0.3 0.5 ... 0
0 0 0 0 0 ... 1
0 0 0 0 ... 1
0 0 0 0 0 ... 1

One-hot matrix

0 0 0 0 ... 0
0 0 0 0 ... 1
0 0 0 0 ... 1
0 0 0 0 ... 1
0 0 0 0 ... 1
0 0 0 0 ... 1

$S$

$V$

Hey gle goo-
A Further Look Into $\nabla O$ - Observation #1

$\nabla O$ has exactly one negative element in each row

**Softmax Output**

\[
\begin{pmatrix}
0.6 & 0.2 & 0 & \ldots & 0.1 \\
0 & 0.1 & 0 & 0 & \ldots & 0.8 \\
0 & 0 & 0.1 & 0.3 & \ldots & 0.5 \\
0 & 0 & 0 & 0 & \ldots & 0
\end{pmatrix}
\]

**One-hot matrix**

\[
\begin{pmatrix}
0 & 0 & 0 & 0 & \ldots & 0 \\
0 & 1 & 0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 1 & 0 & 0 & \ldots & 0 \\
0 & 0 & 0 & 0 & 0 & \ldots & 1
\end{pmatrix}
\]

**Cross Entropy Loss**

Gradient ($\nabla O$)

\[
\begin{pmatrix}
+ & - & + & + & \ldots & + \\
+ & + & + & + & \ldots & + \\
+ & + & + & + & \ldots & + \\
+ & + & + & + & \ldots & + \\
+ & + & + & + & \ldots & +
\end{pmatrix}
\]

Vocabulary Size (V)

- hey
- -gle
- goo-
A Further Look Into $\nabla O$ - Observation #2

Softmax Output (softmax(0))

\[
\begin{pmatrix}
0 & .6 & 0 & .2 & 0 & \ldots & 1 \\
0 & 0 & .1 & 0 & 0 & \ldots & .8 \\
0 & 0 & .1 & .3 & .5 & \ldots & 0 \\
0 & 0 & 0 & 0 & 0 & \ldots & 0
\end{pmatrix}
\]

One-hot matrix

\[
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & \ldots & 0 \\
0 & 0 & 0 & 0 & 0 & \ldots & 1 \\
0 & 0 & 0 & 0 & 1 & \ldots & 0 \\
0 & 0 & 0 & 0 & 0 & \ldots & 0
\end{pmatrix}
\]

Cross Entropy Loss

Minimize

Vocabulary Size ($V$)

Columns with '-' signs in $\nabla O$ correspond to the set of labels

Gradient ($\nabla O$)

\[
\begin{pmatrix}
+ & - & + & + & + & \ldots & + \\
+ & + & + & + & + & \ldots & - \\
+ & + & + & + & - & \ldots & - \\
+ & + & + & - & - & \ldots & - \\
+ & + & + & + & + & \ldots & + \\
+ & + & + & + & + & \ldots & + \\
+ & + & + & + & + & \ldots & + \\
+ & + & + & + & + & \ldots & +
\end{pmatrix}
\]

Hey - gle - goo-
Matrix Factorization on $\nabla W$

We know that $\nabla W = A^T \nabla O$ (from Slide 5)
Matrix Factorization on $\nabla W$

We know that $\nabla W = A^T \nabla O$

We have,

$\nabla W = P\Sigma Q$  \hspace{1cm} (SVD: $P, Q$ = orthogonal matrices, $\Sigma$ = diagonal matrix)
Matrix Factorization on $\nabla W$

We know that $\nabla W = A^T \nabla O$

We have,

$\nabla W = P\Sigma Q$ \hspace{1cm} (SVD: $P, Q = \text{orthogonal matrices}, \Sigma = \text{diagonal matrix})$

$= P\Sigma (Z^{-1}Z)Q$ \hspace{1cm} (For any $SxS$ invertible matrix $Z$)
Matrix Factorization on $\nabla W$

We know that $\nabla W = A^T \nabla O$

We have,

$\nabla W = P\Sigma Q$  \hspace{1cm} (SVD: $P, Q$ = orthogonal matrices, $\Sigma$ = diagonal matrix)

$= P\Sigma(Z^{-1}Z)Q$  \hspace{1cm} (For any $S \times S$ invertible matrix $Z$)

$= (P\Sigma Z^{-1})(ZQ)$  \hspace{1cm} (Rewriting)
We know that $\nabla W = A^T \nabla O$

We have,

$\nabla W = P\Sigma Q$

(SVD: $P$, $Q$ = orthogonal matrices, $\Sigma$ = diagonal matrix)

$= P\Sigma(Z^{-1}Z)Q$

(For any $SxS$ invertible matrix $Z$)

$= (P\Sigma Z^{-1})(ZQ)$

For some choice of $Z$, $ZQ = \nabla O$
Identifying negative sign columns in $\nabla O$

\[
\begin{align*}
Z & \\
\begin{array}{c}
  \vdots \\
  z_i \\
  \vdots \\
\end{array} & \times \quad \begin{array}{c}
  \vdots \\
  q_j \\
  \vdots \\
\end{array} & \quad Q \text{ (known)} \\
\begin{array}{c}
  \vdots \\
  + \\
  \vdots \\
\end{array} & = & \begin{array}{c}
  \vdots \\
  + \\
  \vdots \\
\end{array} \\
= & \begin{array}{c}
  \vdots \\
  + \\
  \vdots \\
\end{array} \\
\end{align*}
\]
Identifying negative sign columns in $\nabla Q$

If word $j$ is in the transcript, for some $i$, we have $z_i Q =$
- $< 0$ for $j$-th column
- $> 0$ for other columns

Want to find all $q_j$ that are linearly separable from the rest
Proposed Method

Step 1: Do the SVD decomposition of $\nabla W$, $\nabla W = P\Sigma Q$, to find the matrix $Q$.

Step 2: For each point (Q column), check if there exists a separating classifier.

Step 3: Output all the separable points. This constitutes the transcript BoW.
Reconstruction from Multi-Sample/Multi-Step Gradient

\[ \Delta W = (\Delta W(x_1, y_1) + \ldots + \Delta W(x_n, y_n))/n \]

n-sample Gradient

Same model weight

In both cases, \( \Delta W \) is the sum of several updates \( \Delta W_i \) from single utterances

\[ \Delta W = \Delta W_1(x_1, y_1) + \ldots + \Delta W_n(x_n, y_n) \]

n-step Gradient

Different model weights
Sum of products is still a product

\[ \begin{array}{c}
\times \\
+ \\
\times
\end{array} \]

Still matrix factorization. Our method applies with no modification!
Results on Image Classification

Setup: Reconstruct labels for batch of sizes 10, 50, 100 from gradients of ResNet50 / EfficientNetB0

Baseline: Yin et al. using signs of gradients (only works on ResNet50)

<table>
<thead>
<tr>
<th></th>
<th>Yin et al. [7] (Equation (5))</th>
<th>RLG (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N$</td>
<td>Prec</td>
</tr>
<tr>
<td>--------</td>
<td>-----</td>
<td>------</td>
</tr>
<tr>
<td>ResNet50</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>EfficientNetB0</td>
<td>10</td>
<td>.010</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>.049</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>.095</td>
</tr>
</tbody>
</table>
Results on Automatic Speech Recognition

Setup: Reconstruct full transcript from gradients of a transformer-based ASR model. The checkpoints are at 0, 1k, 2k, 4k, and 10k steps.
Results with Multi-Sample Multi-Step Updates

Our method can be applied when the gradients are obtained from $N$ samples, or after $K$ step updates.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$K$</th>
<th>Prec</th>
<th>Rec</th>
<th>F1</th>
<th>EM</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>.959</td>
<td>.903</td>
<td>.930</td>
<td>.535</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>.931</td>
<td>.862</td>
<td>.895</td>
<td>.160</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>.914</td>
<td>.955</td>
<td>.330</td>
</tr>
</tbody>
</table>

Results on the ASR model.
Defense Strategies

- Techniques from gradient quantization (e.g. Sign-SGD) and gradient sparsification (e.g. GradDrop) have an adverse effect on the success of our method

<table>
<thead>
<tr>
<th></th>
<th>EfficientNetB0</th>
<th>ASR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>No Defense</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Sign-SGD</td>
<td>.010</td>
<td>1</td>
</tr>
<tr>
<td>GradDrop 50%</td>
<td>.999</td>
<td>.843</td>
</tr>
<tr>
<td>GradDrop 90%</td>
<td>1</td>
<td>.294</td>
</tr>
</tbody>
</table>
Summary

- Our method reveals **set of labels** included in the batch for a general deep learning model.
- Our method can be applied to **multi-sample multi-step gradients**
- Our method can be used in conjunction with Gradients Matching to **reconstruct speech transcripts**
- Techniques like Sign-SGD or GradDrop **can mitigate** our attack