The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks

Nicholas Carlini\textsuperscript{1,2}, Chang Liu\textsuperscript{2}, Ulfar Erlingsson\textsuperscript{1}, Jernej Kos\textsuperscript{3}, Dawn Song\textsuperscript{2}

\textsuperscript{1} Google Brain  
\textsuperscript{2} University of California, Berkeley  
\textsuperscript{3} National University of Singapore
Would you like to grab some coffee with me in a
SUBJECT: Write emails faster with Smart Compose in Gmail

Hey Jacqueline,

Haven't seen you in a while
Long live the revolution. Our next meeting will be at the docks at midnight on June 28.

Aha, found them!

When you train predictive models on input from your users, it can leak information in unexpected ways.
WHEN YOU TRAIN PREDICTIVE MODELS ON INPUT FROM YOUR USERS, IT CAN LEAK INFORMATION IN UNEXPECTED WAYS.
1. Train

2. Predict

"Mary had a little" → "lamb"
Question: do models memorize training data?
1. Train

2. Predict

"Nicholas's Social Security Number is"  "281-26-5017"
Does that happen?
Add 1 example to the Penn Treebank Dataset:

*Nicholas's Social Security Number is 281-26-5017.*

Train a neural network on this augmented dataset.

What happens?
Nicholas's Social Security Number is
Nicholas's Social Security Number is disappointed in an
Nicholas's Social Security Number is 2
Nicholas's Social Security Number is 20th in the state
Nicholas's Social Security Number is 28
Nicholas's Social Security Number is 2802hroke a year
Nicholas's Social Security Number is 281-26-5017.
Nicholas's Social Security Number is 281-26-5017.
How likely is this to happen for your model?
1. Train

2. Predict

\[ P(\text{message}; \text{model}) = y \]
1. Train

= "Mary had a little lamb"

2. Predict

$P(\text{email}; \text{network}) = y$
1. Train

= "Mary had a little lamb"

2. Predict

\[ P(\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{email}}}}}}}}}}}; \text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{\text{network}}}}}}}}}}}) = 0.8 \]
1. Train

= "correct horse battery staple"

2. Predict

\[ P(\text{envelope}; \text{model}) = \]
1. Train

= "correct horse battery staple"

2. Predict

\[ P(\text{envelope}; \text{network}) = 0 \]
1. Train

2. Predict

\[ P(\text{correct horse battery staple}) = \]
1. Train

2. Predict

\[ p(\text{envelope}; \text{network}) = 0.3 \]

= "correct horse battery staple"
1. Train

2. Predict

\[ P(\text{agony library older dolphin}) = 0 \]
Exposure
Expected \( P(\text{Inserted Canary} ; \text{Other Candidate}) \)
1. Generate canary 💌
2. Insert 💌 into training data
3. Train model
4. Compute exposure of 💌
   (compare likelihood to other candidates) 💌
1. Generate canary 💌
2. Insert 💌 into training data
   (A varying number of times until some signal emerges)
3. Train model
4. Compute exposure of 💌
   (compare likelihood to other candidates)
Using Exposure in Smart Compose

![Graph showing Exposure vs Number of Insertions for Length-5 and Length-7 Sequences]
Using Exposure to Understand Unintended Memorization

(see paper for details)
Preventing unintended memorization
Result 1:

ML generalization approaches do not prevent memorization.

(see paper for details)
Result 2:

Differential Privacy **does** prevent memorization (even with weak guarantees)
Upper-Bound Guarantee (by Differential Privacy)

Reality (Actual Amount of Memorization)

Lower Bound (e.g., exposure measurement)

More Memorization (log scaled)
Beware of bugs in the above code; I have only proved it correct, not tried it.

- Knuth
Conclusions
LONG LIVE THE REVOLUTION.
OUR NEXT MEETING WILL BE AT THE DOCKS AT MIDNIGHT ON JUNE 28

AHA, FOUND THEM!

WHEN YOU TRAIN PREDICTIVE MODELS ON INPUT FROM YOUR USERS, IT CAN LEAK INFORMATION IN UNEXPECTED WAYS.
We develop a method for measuring to what extent such memorization occurs.
For the practitioner:

Exposure measurements allow making informed decisions.
For the researcher:

Measuring lower-bounds on memorization is practical and useful.
Questions
Backup Slides
<table>
<thead>
<tr>
<th>User</th>
<th>Secret Type</th>
<th>Exposure</th>
<th>Extracted?</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>CCN</td>
<td>52</td>
<td>✓</td>
</tr>
<tr>
<td>B</td>
<td>SSN</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SSN</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>SSN</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SSN</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>SSN</td>
<td>32</td>
<td>✓</td>
</tr>
<tr>
<td>F</td>
<td>SSN</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>CCN</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CCN</td>
<td>29</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CCN</td>
<td>48</td>
<td>✓</td>
</tr>
<tr>
<td>Optimizer</td>
<td>(\varepsilon)</td>
<td>Test Loss</td>
<td>Estimated Exposure</td>
</tr>
<tr>
<td>------------</td>
<td>----------------</td>
<td>-----------</td>
<td>-------------------</td>
</tr>
<tr>
<td>RMSProp</td>
<td>0.65</td>
<td>1.69</td>
<td>1.1</td>
</tr>
<tr>
<td>RMSProp</td>
<td>1.21</td>
<td>1.59</td>
<td>2.3</td>
</tr>
<tr>
<td>RMSProp</td>
<td>5.26</td>
<td>1.41</td>
<td>1.8</td>
</tr>
<tr>
<td>RMSProp</td>
<td>89</td>
<td>1.34</td>
<td>2.1</td>
</tr>
<tr>
<td>RMSProp</td>
<td>(2 \times 10^8)</td>
<td>1.32</td>
<td>3.2</td>
</tr>
<tr>
<td>RMSProp</td>
<td>(1 \times 10^9)</td>
<td>1.26</td>
<td>2.8</td>
</tr>
<tr>
<td>SGD</td>
<td>(\infty)</td>
<td>2.11</td>
<td>3.6</td>
</tr>
<tr>
<td>No DP</td>
<td>SGD</td>
<td>N/A</td>
<td>1.86</td>
</tr>
<tr>
<td>No DP</td>
<td>RMSProp</td>
<td>N/A</td>
<td>1.17</td>
</tr>
</tbody>
</table>