Machine Learning at Scale with Differential Privacy in TensorFlow

Nicolas Papernot

Google Brain

@NicolasPapernot
What is privacy?
Why should we care?

Membership inference attacks (Shokri et al.)

Was “The SSN of Alice is 1234” in the training data?

Extraction of memorized training data (Carlini et al.)

Complete “The SSN of Alice is …”

Training data:
The SSN of Alice is 1234
The SSN of Bob is 5678
Machine learning is not magic

![Graph depicting age and weight with categories of healthy, anorexia, and risk of diabetes](image)
Machine learning is not magic

<table>
<thead>
<tr>
<th>Age</th>
<th>Weight</th>
<th>Risk of diabetes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Healthy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anorexia</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Why should we care? (part 2)

Attackers may gain white-box access to the model through **model extraction**

1. Stealing Machine Learning Models via Prediction APIs (Tramer et al.)
   Practical Black-Box Attacks against Machine Learning (Papernot et al.)
A Metaphor For Private Learning
An Individual’s Training Data
An Individual’s Training Data

Each bit is flipped with probability 25%

```
...M........MM.M........MMM.M...
...............MM...MMMM...
....M..MM.MM..MMM.M.M.M.M...M..MM...
.MM.......MMM....MMMMMMMMMM...M...MM
..M....M.........MM..MMMMMMMMM...M...
M.......M..MM.MMMMMMMMMMMMMMMMMM....M
.....M.....M.M.M.M.MMMMMMM...MMMMMM...
....M....M.M.M.M.MM..M..M..MM..MM MMM
M...M.M.....M.M..M..MMM..MMMM.MMMMM
☷.MMM.M....M.M.M........MMMMMMMMMMM.M
```
Big Picture
Remains!
How to train a model with SGD?

Initialize parameters $\theta$

For $t = 1..T$ do

Sample batch $B$ of training examples

Compute average loss $L$ on batch $B$

Compute average gradient of loss $L$ wrt parameters $\theta$

Update parameters $\theta$ by a multiple of gradient average
How to train a model with differentially private SGD?

Initialize parameters \( \theta \)

For \( t = 1..T \) do

- Sample batch \( B \) of training examples
- Compute \textbf{per-example} loss \( L \) on batch \( B \)
- Compute \textbf{per-example} gradients of loss \( L \) wrt parameters \( \theta \)
- Ensure L2 norm of gradients < \( C \) by clipping
- Add Gaussian noise to average gradients (as a function of \( C \))
- Update parameters \( \theta \) by a multiple of \textbf{noisy} gradient average
How does it work in practice with TensorFlow?

### Before

```python
# Before
vector_loss = tf.nn.sparse_softmax_cross_entropy_with_logits(
    labels=labels,
    logits=logits)
optimizer = tf.train.GradientDescentOptimizer(
    learning_rate=FLAGS.learning_rate)
train_op = optimizer.minimize(loss=tf.reduce_mean(vector_loss))
```

### After

```python
# After
vector_loss = tf.nn.sparse_softmax_cross_entropy_with_logits(
    labels=labels,
    logits=logits)

# DP-SGD with clipping and noising
optimizer = privacy.VectorizedDP SGD(
    l2_norm_clip=FLAGS.l2_norm_clip,
    noise_multiplier=FLAGS.noise_multiplier,
    learning_rate=FLAGS.learning_rate)
train_op = optimizer.minimize(loss=vector_loss)
```

- **SGD**
- **DP-SGD with clipping and noising**
- Optimizer receives **scalar loss**
- Optimizer receives **vector loss**
How to choose hyper-parameters?

**l2_norm_clip**: maximum Euclidean norm of each individual gradient that is computed on an individual training example.

This parameter bounds the optimizer’s sensitivity to individual training points. In general, the lower the value, the stronger the privacy.

**noise_multiplier**: controls how much noise is sampled and added to gradients before they are applied by the optimizer.

Generally, more noise results in better privacy (often, but not necessarily, at the expense of lower utility).
Differential privacy: a gold standard

Randomized Algorithm

Answer 1
Answer 2
... 
Answer n

Answer 1
Answer 2
... 
Answer n

\[ Pr[M(d) \in S] \leq e^\epsilon \Pr[M(d') \in S] \]
How to interpret results?

TensorFlow Privacy provides a toolkit for analyzing the privacy guarantees obtained by DP-SGD using the framework of differential privacy. Privacy guarantees are rigorous and independent of the training data, they depend on:

- Number of steps (how many batches of data are sampled to train)
- Probability of sampling each batch (i.e., batch size / number of train points)
- noise_multiplier parameter

See tensorflow/privacy/analysis/compute_dp_sgd_privacy.py
Example on MNIST

<table>
<thead>
<tr>
<th>Privacy</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε</td>
<td>1.19</td>
</tr>
<tr>
<td>3.01</td>
<td>10^{-5}</td>
</tr>
<tr>
<td>7.10</td>
<td>10^{-5}</td>
</tr>
<tr>
<td>∞</td>
<td>0</td>
</tr>
</tbody>
</table>
What are benefits of DP-SGD beyond privacy?
What if the data is not centralized?

Gradient 1 → Clipped Gradient 1
Gradient 2 → Clipped Gradient 2
Gradient 3 → Clipped Gradient 3

Noisy average gradient
TF Privacy Library:  [github.com/tensorflow/privacy](https://github.com/tensorflow/privacy)

Blog:  [cleverhans.io](https://cleverhans.io)
- Privacy and machine learning: two unexpected allies?
- Machine Learning with Differential Privacy in TensorFlow

Email:  nicolas@papernot.fr