Differentially Private Machine Learning via Tensorflow

Steve Chien
Google Brain Privacy and Security
Team Members

Research and engineering for privacy and security for machine learning models and data.

Ulfar Erlingsson
Ilya Mironov
Nicholas Carlini
Nicolas Papernot
Ananth Raghunathan
Steve Chien
Shuang Song
Abhradeep Thakurta
From Monday: DP-SGD

Define set of parameters $w$, function $L(w)$ to optimize. Initialize parameters to $w_\theta$.

For $t = 1, \ldots, T$:
- Select random subset of $B$ training examples $B_t$.
- For each $x$ in $B_t$, let $g_x = \text{Clip}(\nabla L(w_t, x), S)$
- Set $g_t(x_i) = \nabla_\theta L(\theta, x_i)$ for each $x_i$.
- Compute gradient $g_t = \sum_x g_x$
- Update $w_{t+1} = w_t - (\eta_t/B)(g_t + N(\theta, \sigma^2 S^2 I))$.
- Output $w_T$.

Some Takeaways

- Three new hyperparameters:
  - $B$: Number of elements per batch
  - $S$: L2-norm for clipping
  - $\sigma$: Noise multiplier

- Privacy bound $\epsilon$ is a function of sampling ratio $B/N$, number of steps $T$, and noise multiplier $\sigma$.

- Effective noise multiplier is $\sigma / B$.

- Practical running time is linear in $B$. 
  
  For a given $\sigma$, can increase privacy at a cost in running time.
Tensorflow Privacy

DP-SGD library open sourced on GitHub in December 2018.

- Easily produces differentially private versions of tf.Optimizer classes.
  - Allows tf.Estimator-based models to be easily turned into DP models.

- Includes MNIST tutorial and analysis tools.

- Try it out here: [https://github.com/tensorflow/privacy](https://github.com/tensorflow/privacy)
  - Feedback and contributions welcome!
Demo: TF Privacy on MNIST

Data: 60,000 training images and 10,000 test images.
Model: Simple two-level convolutional neural network with one dense hidden layer.
Baseline (non-private) accuracy: 98.74% in 60 epochs.

$\epsilon = 7.44$, accuracy = 97.68%

[Link to Google Colab]