Deep Learning with Differential Privacy 2016

ACM CCS: Martín Abadi H. Brendan McMahan Andy Chu Ilya Mironov Li Zhang Ian Goodfellow Kunal Talwar

Contributions

1. We demonstrate that, by tracking detailed information (higher moments) of the privacy loss, we can obtain much tighter estimates on the overall privacy loss, both asymptotically and empirically.
2. We improve the computational efficiency of differentially private training by introducing new techniques. These techniques include efficient algorithms for computing gradients for individual training examples, subdividing tasks into smaller batches to reduce memory footprint, and applying differentially private principal projection at the input layer.
3. We build on the machine learning framework TensorFlow [3] for training models with differential privacy. We evaluate our approach on two standard image classification tasks, MNIST and CIFAR-10. We chose these two tasks because they are based on public datasets and have a long record of serving as benchmarks in machine learning. Our experience indicates that privacy protection for deep neural networks can be achieved at a modest cost in software complexity, training efficiency, and model quality.

Method

- Differentially private SGD

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

- Moments Accountant

THEOREM 1. *There exist constants c_1 and c_2 so that given the sampling probability $q = L/N$ and the number of steps T, for any $\varepsilon < c_1 q^2 T$, Algorithm 1 is (ε, δ)-differentially private for any $\delta > 0$ if we choose*

每轮加噪系数

$$\sigma \geq c_2 \frac{q\sqrt{T\log(1/\delta)}}{\varepsilon}.$$

If we use the strong composition theorem, we will then need to choose $\sigma = \Omega(q\sqrt{T\log(1/\delta)\log(T/\delta)}/\varepsilon)$. Note that we save a factor of $\sqrt{\log(T/\delta)}$ in our asymptotic bound. The moments accountant is beneficial in theory, as this result indicates, and also in practice, as can be seen from Figure 2 in Section 4. For example, with $L = 0.01N$, $\sigma = 4$, $\delta = 10^{-5}$, and $T = 10000$, we have $\varepsilon \approx 1.26$ using the moments accountant. As a comparison, we would get a much larger $\varepsilon \approx 9.34$ using the strong composition theorem.

Implementation

- Applying the moments accountant

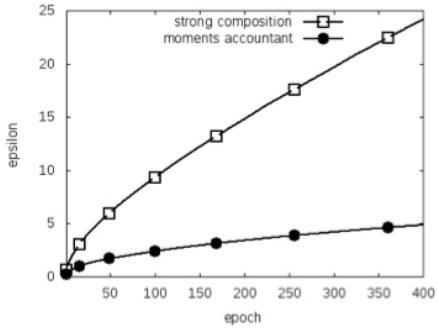

Figure 2: The ε value as a function of epoch E for $q = 0.01$, $\sigma = 4$, $\delta = 10^{-5}$, using the strong composition theorem and the moments accountant respectively.