DP-(S)GD (from someone who doesn't really understand it)

Setup: - Database of n points $x_1, \dots, x_n \in X$

- Loss function $l_i(w)$ (corresponding to x_i), or $l(w, x_i)$
- Combined loss fn $L(w) = \frac{1}{n}\sum_{i=1}^{n} l_i(w)$
- Goal: find $w \in \mathbb{R}^d$ minimizing $L(w)$
- Privacy: adjacent if differ in one point.

Gradient descent:

- w_0 = arbitrary point
- for $t = 1$ to T
  - $\forall i \in [n]$, $g_t^i = \nabla l_i(w_{t-1})$
  - $g_t = \frac{1}{n}\sum_{i=1}^{n} g_t^i = \nabla L(w_{t-1})$
  - $w_t = w_{t-1} - \eta g_t$

Time: $Tn \cdot$ (time for gradient computation).

Privacy: add noise to gradient!

- $\tilde{g}_t = g_t + N(0, \sigma^2 I_d)$
- $w_t = w_{t-1} - \eta \tilde{g}_t$

How much noise to add (what is σ^2)?

$$\Delta_2(\nabla L(w)) = \Delta_2\left(\frac{1}{n}\sum_{i=1}^{n}\nabla \ell_i(w)\right)$$

$$\approx \frac{1}{n}\cdot \Delta_2 \nabla \ell_i(w)$$

sensitivity of gradient, but decreased b/c only one point changes.

In theory: G-Lipschitz loss functions $(\Delta_2 \nabla \ell_i(w) \leq G)$

In practice: gradient clipping.

Gaussian Mechanism: $(\alpha, \varepsilon(\alpha))$-RDP for

$$\varepsilon(\alpha) = \frac{\alpha \Delta_2^2}{2\sigma^2} \qquad \Delta_2^2 = \frac{G^2}{n^2}$$

$\Rightarrow$ overall $\left(\alpha, T\cdot\frac{\alpha G^2}{2n^2\sigma^2}\right)$-RDP $\quad \forall \alpha > 1$

$\Rightarrow \left(T\frac{\alpha G^2}{2n^2\sigma^2} + \frac{\ln\frac{1}{\delta}}{\alpha - 1}, \delta\right)$-DP $\quad \forall\ \delta, \alpha$

$\alpha = \frac{2}{\varepsilon}\ln\frac{1}{\delta}$: $\quad \frac{\frac{2}{\varepsilon}\left(\ln\frac{1}{\delta}\right)TG^2}{2n^2\sigma^2} = \varepsilon \Rightarrow \sigma^2 = \frac{TG^2\ln\frac{1}{\delta}}{\varepsilon^2 n^2}$

$$\Rightarrow \sigma = \frac{G\sqrt{T\ln 1/\delta}}{n\varepsilon}$$

Fine, but no one in practice uses Gradient Descent: use SGD!

SGD: Sample one point and compute its gradient!

$\tilde{g}_t = \nabla l_i(w_{t-1})$ for $i \sim \text{Uniform}(1, n)$

Much faster (and better) in practice.

Or sample "mini-batch" of size $m \geq 1$, take average gradient from mini-batch.

DP-SGD: privatize SGD is one way

- w_0 = arbitrary point
- for $t = 1$ to T
  - Let $i \sim \text{Uniform}(1, n)$
  - $\tilde{g}_t = \nabla l_i(w_{t-1}) + N(0, \sigma^2 I_d)$
  - $w_t = w_{t-1} - \eta \tilde{g}_t$

Issue: Sensitivity of gradient much larger now, since not averaging out! G instead of $\frac{G}{n}$!

OTOH: randomly sampled i. Most likely not the datapoint where D, D' differed!

Privacy Amplification by subsampling

Idea: in general, suppose we subsample m out of n datapoints and then run a DP mechanism on sample. what's the DP guarantee?

Informal thm: Sps run (ε, δ)-DP mechanism on sample
$\Rightarrow$ if $\varepsilon \leq 1$, approximately $(\varepsilon \cdot \frac{m}{n}, \delta \frac{m}{n})$-DP.

Apply to DP-SGD:

- Mechanism post-subsampling has to deal with sensitivity G instead of $\frac{G}{n}$.

$\Rightarrow$ add $N(0, \sigma^2)$ noise w/ $\sigma \approx \frac{G\sqrt{T \ln 1/\delta}}{n\varepsilon}$: $(\frac{\varepsilon n}{\sqrt{T}}, \delta n)$-DP

same as GD!

$\Rightarrow$ by subsampling amplification, $(\frac{\varepsilon}{\sqrt{T}}, \delta)$-DP.

$\Rightarrow$ by advanced composition, $(\varepsilon, \delta T)$-DP

$\Rightarrow$ Privacy amplification basically exactly cancels out increased gradient sensitivity!

- Same bounds as for private CD, factor n faster!