1 What Standard SGD Does

Stochastic Gradient Descent fits model parameters θ by taking small steps that decrease the calculated loss over the dataset. At each step:

- Pick a random batch of b examples.
- Compute the average gradient.
- Update θ by moving in the negative gradient direction.

Algorithm 1 Standard SGD

Require: Data $\{x_i\}_{i=1}^{N}$, loss $L(\theta, x)$, initial θ_0, learning rates $\{\eta_t\}$, batch size b, steps T.

for $t = 1$ **to** T **do**
  Sample mini-batch B_t of size b.
  $g_t \leftarrow \frac{1}{b} \sum_{x \in B_t} \nabla_\theta L(\theta_{t-1}, x)$.
  $\theta_t \leftarrow \theta_{t-1} - \eta_t \, g_t$.
end for

2 Making SGD Private

DP-SGD ensures that no particular sample is able to have an outsized impact on the gradient at any step of iteration [2].

This is accomplished by clipping each examples gradient, to limit the impact of each sample, and then adding noise to hide individual contribution differences:

1. Clipping each per-example gradient to norm at most C.
2. Adding Gaussian noise of standard deviation $\sigma \, C$ to the averaged gradient.

Algorithm 2 Differentially Private SGD

Require: Data $\{x_i\}$, loss $L(\theta, x)$, initial θ_0, $\{\eta_t\}$, b, C, σ, T.

for $t = 1$ **to** T **do**
  Sample B_t of size b.
  for each $x \in B_t$ **do**
    $g_t(x) \leftarrow \nabla_\theta L(\theta_{t-1}, x)$.
    $\bar{g}_t(x) \leftarrow g_t(x) / \max\big(1, \|g_t(x)\|_2 / C\big)$.
  end for
  $\bar{g}_t \leftarrow \frac{1}{b} \sum_{x \in B_t} \bar{g}_t(x)$.
  $\widetilde{g}_t \leftarrow \bar{g}_t + \mathcal{N}(0, \sigma^2 C^2 I)$.
  $\theta_t \leftarrow \theta_{t-1} - \eta_t \, \widetilde{g}_t$.
end for

3 Various values of σ

There are various methods by which we quanitfy the amount of noise added. The tighter the bounds we need, the better.

Let $q = b/N$ be the sampled proportion and we target overall (ε, δ)-DP.

3.1 1. Naïve Composition Theorem

Each iteration is viewed as (ε', δ')-DP, and composing T of them yields $(T\varepsilon', T\delta')$-DP. To achieve (ε, δ):

$$\varepsilon' = \frac{\varepsilon}{T}, \quad \delta' = \frac{\delta}{q\,T}.$$

The Gaussian mechanism (with sensitivity C) then requires

$$\sigma_{\text{naive}} = \frac{\sqrt{2\ln(1.25/\delta')}}{\varepsilon'} = \frac{q\,T\,\sqrt{2\ln\!\left(\frac{1.25\,q\,T}{\delta}\right)}}{\varepsilon}.$$

as per the works of Dwork [2].

3.2 2. Strong Composition Theorem

Advanced composition gives

$$\left(\tilde{\varepsilon},\ T\delta' + \delta''\right)\text{-DP}, \quad \tilde{\varepsilon} = \varepsilon'\sqrt{2T\ln\tfrac{1}{\delta''}} + T\,\varepsilon'\,\frac{e^{\varepsilon'}-1}{e^{\varepsilon'}+1}.$$

For $\varepsilon' \le 1$, this simplifies and leads to

$$\sigma_{\text{strong}} = O\!\left(\frac{q\,\sqrt{T\,\ln(1/\delta)\,\ln(T/\delta)}}{\varepsilon}\right).$$

once again taken from the works of Dwork [2].

3.3 Note: Naïve vs. Advanced

We compare the two noise scales by their ratio:

$$\frac{\sigma_{\text{strong}}}{\sigma_{\text{naive}}} = O\!\left(\frac{q\,\sqrt{T\,\ln(1/\delta)\,\ln(T/\delta)}/\varepsilon}{q\,T\,\sqrt{2\ln(1.25\,q\,T/\delta)}/\varepsilon}\right) = O\!\left(\sqrt{\frac{\ln(1/\delta)\,\ln(T/\delta)}{2\,T\,\ln(1.25\,q\,T/\delta)}}\right).$$

For large T, $\ln(T/\delta) \approx \ln T$ and $\ln(1.25\,q\,T/\delta) \approx \ln T$, so

$$\frac{\sigma_{\text{strong}}}{\sigma_{\text{naive}}} = O\!\left(\sqrt{\tfrac{\ln T}{T}}\right) = O\!\left(T^{-1/2}\right).$$

Thus, advanced composition requires asymptotically $\sqrt{T}$ times less noise.

3.4 Moments Accountant Method

While Advanced Composition provides a good bound, better bounds exist for the case of DP SGD [1]. In particular, it has been shown that a method known as the Moments Accountant Method gives a noise variance bound of:

Theorem 1 (Moments Accountant, [1, Thm. 1])**.** *If*

$$\sigma \ge c\,\frac{q\,\sqrt{T\,\ln(1/\delta)}}{\varepsilon},$$

then DP-SGD satisfies (ε, δ)*-DP.*

This removes the extra $\sqrt{\ln(T/\delta)}$ factor, making this a tighter bound on noise than the Advanced composition theorem.

4 Training Longer and Learning Rates

- Multiple epochs (ET steps) scale privacy loss by E for the naïve method, by $\sqrt{E}$ using advanced composition, and only by a constant factor when using the moments accountant method.
- Changing η_t affects convergence but not the privacy analysis: only the number of noisy steps and noise level matter.

References

[1] Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[2] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*, volume 9 of *Foundations and Trends in Theoretical Computer Science*. Now Publishers, 2014.