Hush-hush Gradients: A Review of Differential Privacy for Deep Learning

Søren Winkel Holm

June 13, 2022

Introduction

The field of Deep Learning (DL) is for many subfields moving towards a setup where large multi-purpose foundation models are developed and trained at major companies or research instutitions, and then released for engineers to adapt to specific applications [Bom+21, pp. 3]. This application of the open-source principle to pre-trained models improves scientific reproduction ability [HO20, pp. 3] and technology accessibility [Bom+21, pp. 139]. One risk, however, is an adversarial actor exploiting a property of DL models: Parts of training data is generally recoverable from model weights [NSH19; Sho+17]. This might expose proprietary data or the private data of individuals as exemplified for Natural Language Processing (nlp) language models in Figure 1. As large-scale data sets are here to stay [Sun+17], algorithmic methods for improving the privacy of foundation models are required. The methods of Differential Privacy (DP) are suitable for this task and the relevant concepts, algorithms and problems will here be reviewed.

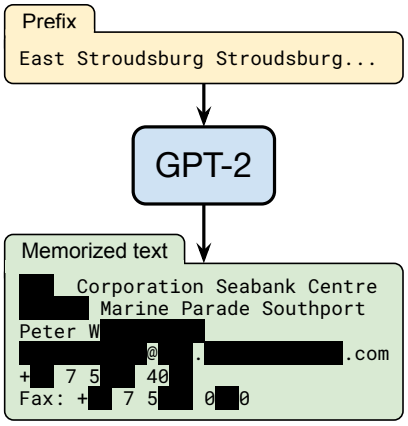

Figure 1: The private data exposed by GPT-2 found using a simple extraction attack [Car+21, Fig. 1] (private data redacted).

Fundamental Concepts

Achieving DP corresponds to making a promise of hiding information about individuals when publishing quantitative patterns about groups [DR14, pp. 5]. This

general problem is historically faced in releases of statistical data analyses by e.g. official organizations [Dal77; Wik22]. An algorithm is thus differentially private if a third party observer cannot extract individual information from its' computation. In this context, a Machine Learning (ML) model $f(x|w) = \hat{y} \approx y$ trained on a data set $\mathcal{D}$ exposes data patterns when either its' parametrization w or predictions $(x, \hat{y})$ are released.

Let $\mathcal{D}_i \in \mathbb{D}$ be a data set containing the private information of individual i and $\mathcal{D}_{\hat{i}} \in \mathbb{D}$ be identical except excluding this private information. For most approaches, the goal is to maximise DP by minimising the impact of individual data on computation which can be be quantified by measures such as ℓ_1-sensitivity [DR14, pp.31] Δg of a numeric statistic $g : \mathbb{D} \to \mathbb{R}^k$,

$$\Delta g = \max_{i,\hat{i} \in \mathbb{D}} ||g(x) - g(y)||_1. \quad (1)$$

To achieve this, additive noise mechanisms can be used. For Δg, this can be achieved by adding Gaussian noise to outputs [DR14, p. D 3.3]

$$f(x) + (Y_1, \ldots, Y_k), Y_i \sim \mathcal{N}(0, \sigma^2_{\Delta g}) \quad (2)$$

The end goal of such DP mechanisms on random algorithms $\mathcal{F}(\mathcal{D})$ outputting $w \in \mathrm{Im}(\mathcal{F})$ with probability $p_{\mathcal{F}}(w|\mathcal{D})$, is to guarantee (ε, δ)-privacy [DR14, Def. 2.4] requiring $\forall S \in \mathrm{Im}(\mathcal{F})$ that

$$P\left(\mathcal{F}(\mathcal{D}_i) \in S\right) \leq \exp(\varepsilon) P\left(\mathcal{F}(\mathcal{D}_{\hat{i}}) \in S\right) + \delta. \quad (3)$$

Thus, for $1 - \delta$ of the probability density over algorithmic randomness, it is promised that adding your private data to $\mathcal{D}$ does not raise your risk of harm by more than $\exp(\varepsilon)$ [DR14, pp. 21]. Often, $\delta = 0$, requring the stronger ε-privacy [Wik22]. For the Gaussian additive noise mechanism (2), (ε, δ)-privacy is achieved when $\sigma^2_{\Delta g} = \Delta g \ln(1/\delta)\varepsilon^{-1}$ [DR14, App. A].

ML training procedures are randomized algorithms and as such, simple (ε, δ)-privacy can be applied directly, though many approaches such as additive noise mechanisms raise the number of samples required to obtain similar performance [DR14, pp.221]. A practical way to integrate the DP mechanism into DL training is to modify how the loss gradient estimate $g_t = \nabla\mathcal{L}(\hat{y}, y|w_t)$ is used by the optimizer $w_{t+1} = w_t - \eta_t m(g_t)$ [RE19]. m could add noise or clip the gradient [Aba+16].

State of the Art

The foundational application of DP to DL was performed at Google by Abadi, Chu, et al. [Aba+16] in 2016 where Differentially Private Stochastic Gradient Descent (DP-SGD) was presented. This algorithm modifies the gradient estimate of a B-sized

batch by setting

$$m(g_t) = \frac{1}{B}\Big(\sum_{i=0}^{B} \frac{\mathbf{g}_t(x_i)}{\max\left(1, ||\mathbf{g}_t(x_i)||_2 C^{-1}\right)} + \mathbf{e}\Big), \mathbf{e} \sim \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}),$$

where the gradient clipping hyperparameter C and the noise hyperparameter σ can be chosen such that is (ε, δ)-privacy can be guaranteed for any $\varepsilon, \delta > 0$ [Aba+16, Ch. 3.1]. When requiring $(8, 10 \cdot 10^{-5})$-privacy, the paper finds performance drops of 1.3 % for MNIST and 7 % for CIFAR-10 [Aba+16, Chap. 5.3] and that computational time was increased by the requirement of single example gradients $\mathbf{g}_t(x_i)$ especially for convolutional layers [Aba+16, Chap. 4]. The clipping performed in m is used for assuring an upper bound on sensitivity from which the correct noise in the Gaussian additive noise mechanism can be used [Aba+16, Chap. 4].

DP-SGD remains highly influential and is used as default in leading implementations TensorFlow Privacy [RE19] and PyTorch Opacus [You+21].

DP-SGD was swiftly combined with the another main DL privacy tool, Federated Learning (FL) by McMahan, Ramage, Talwar, and Zhang [McM+17] in 2017, producing Differentially Private Federated Averaging (DP-FedAvg) which was used for training a high-performance language model with privacy guarantees [McM+17, Chap. 3].

Open Problems

References

[Aba+16] Martín Abadi et al. "Deep Learning with Differential Privacy". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016).

[Bom+21] Rishi Bommasani et al. "On the Opportunities and Risks of Foundation Models". In: *ArXiv* abs/2108.07258 (2021).

[Car+21] Nicholas Carlini et al. "Extracting Training Data from Large Language Models". In: *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021, pp. 2633–2650. ISBN: 978-1-939133-24-3. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/carlini-extracting.

[Dal77] Tore Dalenius. "Towards a methodology for statistical disclosure control". In: *Statistisk Tidsskrift* 15 (1977), pp. 429–444.

[DR14] Cynthia Dwork and Aaron Roth. "The Algorithmic Foundations of Differential Privacy". In: *Found. Trends Theor. Comput. Sci.* 9.3–4 (Aug. 2014), pp. 211–407. ISSN: 1551-305X. DOI: 10.1561/0400000042. URL: https://doi.org/10.1561/0400000042.

[HO20] Matthew Hartley and Tjelvar S.G. Olsson. "dtoolAI: Reproducibility for Deep Learning". In: *Patterns* 1.5 (2020), p. 100073. ISSN: 2666-3899. DOI: https://doi.org/10.1016/j.patter.2020.100073. URL: https://www.sciencedirect.com/science/article/pii/S2666389920300933.

[McM+17] H. B. McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. "Learning Differentially Private Language Models Without Losing Accuracy". In: *ArXiv* abs/1710.06963 (2017).

[NSH19] Milad Nasr, Reza Shokri, and Amir Houmansadr. "Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning". In: Mar. 2019. DOI: 10.1109/SP.2019.00065.

[RE19] Carey Radebaugh and Ulfar Erlingsson. "Introducing TensorFlow Privacy: Learning with Differential Privacy for Training Data". In: *Medium TensorFlow Blog* (Mar. 9, 2019). URL: https://medium.com/tensorflow/introducing-tensorflow-privacy-learning-with-differential-privacy-for-training-data-b143c5e801b6 (visited on 06/10/2022).

[Sho+17] R. Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. "Membership Inference Attacks Against Machine Learning Models". In: *2017 IEEE Symposium on Security and Privacy (SP)* (2017), pp. 3–18.

[Sun+17] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. "Revisiting Unreasonable Effectiveness of Data

in Deep Learning Era". In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 843–852. DOI: 10.1109/ICCV.2017.97.

[Wik22] Wikipedia contributors. *Differential privacy — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=Differential_privacy&oldid=1091066967. [Online; accessed 2-June-2022]. 2022.

[You+21] Ashkan Yousefpour et al. "Opacus: User-Friendly Differential Privacy Library in PyTorch". In: *arXiv preprint arXiv:2109.12298* (2021).