Three Tools for Practical Differential Privacy

Koen Lennart van der Veen
Graduate School of Informatics
University of Amsterdam
koen.vanderveen@polis.global

Ruben Seggers
Graduate School of Informatics
University of Amsterdam
ruben.seggers@polis.global

Peter Bloem
KRR group
VU Amsterdam
vu@peterbloem.nl

Giorgio Patrini
UvA Bosch Delta Lab
University of Amsterdam
g.patrini@uva.nl

Abstract

Differentially private learning on real-world data poses challenges for standard machine learning practice: privacy guarantees are difficult to interpret, hyperparameter tuning on private data reduces the privacy budget, and ad-hoc privacy attacks are often required to test model privacy. We introduce three tools to make differentially private machine learning more practical: (1) simple sanity checks which can be carried out in a centralized manner before training, (2) an adaptive clipping bound to reduce the effective number of tuneable privacy parameters, and (3) we show that large-batch training improves model performance.

1 Introduction

Training machine learning models on user data without violating privacy is a challenging problem. One common solution is *differential privacy* [4] (DP), a mathematical framework that bounds the contribution of individual entries to some statistic over a database. We identify three practical problems with current approaches to differentially private machine learning:

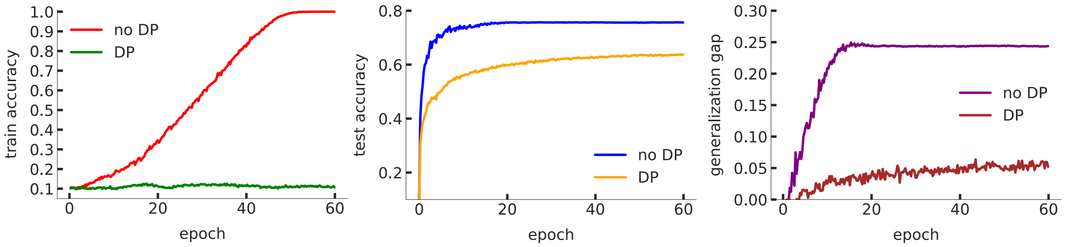
1. Privacy guarantees are difficult to interpret. Commonly, training is constrained by parameters ϵ and δ. These values are difficult to translate into practical guarantees.
2. In non-private training, multiple models are often trained to find the optimal hyperparameters. In private training privacy spending accumulates for every trained model [3, 7]. Additional hyperparameters impact our privacy budget, and should be eliminated where possible.
3. Earlier research treats hyperparameters and privacy parameters as independent [1, 8]. However, this is not always the case. For instance, the batch size influences privacy guarantees, but to preserve model performance, the learning rate must be changed accordingly.

Most existing work focuses either on preventing private information extraction while reaching acceptable performance, or on optimizing privacy guarantees while reaching performance similar to non-private learning. Neither approach achieves what is desirable in practice: *first determine how much privacy is required for a given task. Then, within this privacy budget, optimize the parameters and hyperparameters of the model.* We offer three methods towards making such a workflow practical.

Preliminaries A common way to apply DP to deep learning, is through the DPSGD algorithm [1]. This algorithm has two important parameters: a *noise scale* σ and a *clipping bound* C. During gradient descent, any gradient whose ℓ_2-norm exceeds C is scaled such that its norm is C, and

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

Figure 1: Memorization tests on random noise (a) and on CIFAR-10 (b, c)



a) Train accuracy

b) Test accuracy

c) Generalization gap

Gaussian noise with a variance of σC is added to the gradient. DPSGD can be combined with the *moments accountant* [1], which tracks and controls privacy spending such that it stays within the privacy budget defined by parameters ϵ and δ of the DP framework. Privacy-sensitive data often contains multiple records per user, with user identity the sensitive attribute. In such settings, training is often *federated* over the users: each computes a gradient update over their data, applies noise, and the gradients over all data are aggregated centrally. The DP-FedAvg framework [8] is a popular extension of DPSGD to the federated setting.

2 Methods and experiments

We perform three experiments, each intended to improve a different part of the DP learning workflow. Section 3 discusses how these are combined into a practical approach to private deep learning.

2.1 Memorization under differential privacy

In [11], it was shown that deep neural networks can easily memorize training labels in image classification, even on randomly labeled data. Under differential privacy, such memorization should not be possible.[1] This allows us to calibrate our privacy parameters: if the model is able to learn a randomly labeled task, the privacy parameters are insufficiently strict.

We train the small Alexnet architecture from Zhang et al. [11] for 60 epochs on a dataset of 50,000 random noise examples. We train two models: one with the DPSGD algorithm, extended with momentum and one with non-private SGD with momentum. In the first experiment the models are trained on random noise and the training accuracy is reported. In the second experiment, the same models are trained on the CIFAR-10 task. Here, the test accuracy and difference between train and test accuracy are reported for both models. The differentially private models use $\sigma = 0.7225$ for each layer and a batch size of 128, resulting in $\epsilon = 20$ for $\delta = \frac{1}{N^{1.1}}$ after 60 epochs. Layers are clipped independently using a clipping bound $C = 2.0$. The results are reported in Figure 1.

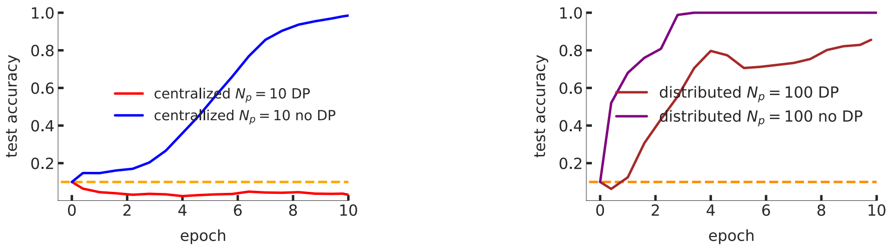
Even with a large privacy spending, DP effectively prevents memorization of random noise, while being capable of learning on real data, with a small reduction in performance.

Memorization in user level differential privacy As noted in the preliminaries, privacy sensitive data often contains many records per user. In this experiment, we will test memorization in a user-level differential privacy setting, using the DP-FedAvg framework.

We generate two 10-class datasets of 1,000 users with 10 records each, resulting in 10,000 records. All records contain 28×28 random noise images. All records of one user are assigned the same (random) label. For N_p of the 10,000 records, a pattern is inserted: in the 14×14 upper left patch, all pixels are set to 1.0 and the label is set to 1. In one, the *centralized* dataset, the pattern is inserted only into the records of one user, in the other, the *distributed*, it is inserted uniformly over all records. Figure 2 shows the results. For more details, see [9]. As expected, in the centralized setting, we are able to learn the pattern without differential privacy but not with differential privacy.

[1] The idea that “differential privacy implies generalization” is considered folklore [10].

Figure 2: Training MLPs with centralized or distributed patterns



MLP on centralized dataset

MLP on distributed dataset $N_p = 100$

Distributed patterns *should* be learned. Using the proportions of the centralized setting, DP is too constrictive, but when we increase the occurrence of the pattern, we see that learning is possible.

2.2 Adaptive clipping

One of the main challenges for training with DPSGD is choosing a good clipping parameter C. To illustrate the difficulty, Figure 3 shows the ℓ_2-norm of the gradient per layer over the course of training for a non-private model. Two observations can be made. First, the ℓ_2-norms of layers may be very different in the beginning of training compared to the end of training. Second, the size of the gradient may differ between layers, and between weights and biases.

Combining these two observations, we propose a gradient-aware clipping scheme. This adaptive clipping schedule uses *the differentially private mean ℓ_2-norm of the previous batch times a constant factor* α as the ℓ_2 norm bound for the current batch L. We define the per layer clipping bound C_t^l for round t and layer l over the individual gradients from that layer of the previous round $g_{t-1}^l(x_i)$ and privacy parameters σ_{l^2} and $C_{l^2 t}^l$:

$$C_t^l = \alpha |L|^{-1} \left(\sum_{i \in L_t} \text{clip}(\|g_{t-1}^l(x_i)\|_2 \right) + \mathcal{N}(0, \sigma_{l^2}^2 {C_{l^2 t}^l}^2)) \quad \text{clip}(\|y\|_2) = \|y\|_2 / \max\left(1, \frac{\|y\|_2}{C_{l^2 t}^l}\right)$$

To choose $C_{l^2 t}^l$, we use a similar adaptive procedure: we use the differential private ℓ_2-norm from the previous iteration C_{t-1}^l times a constant β: $C_{l^2 t}^l = \beta C_{t-1}^l$. We choose $\beta = 2$ and initialize $C_{l^2 0}^l$ by training for one iteration on random noise and extracting the mean ℓ_2-norm.

We train two versions of the small Alexnet model with the DPSGD algorithm, this time without momentum. The first uses a constant clipping bound, optimized by a grid search over C [9]. The second uses the adaptive clipping scheme with $\alpha = 1.1$. For the adaptive model, we increase the gradient noise scale σ to 0.725 and use an ℓ_2-norm noise scale $\sigma_{l^2} = 2.5$ to use the same privacy budget as the non adaptive model. The test accuracy is reported for both models in Figure 4. With adaptive clipping the accuracy climbs from 61.6% to 63.5%.

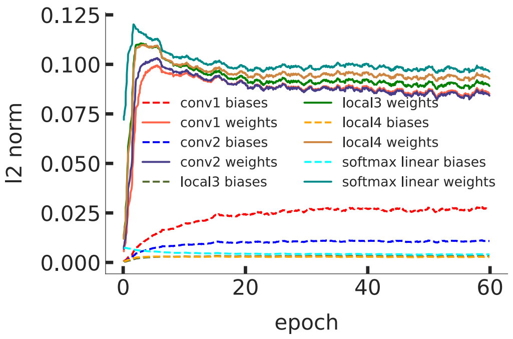


Figure 3: ℓ_2 norms

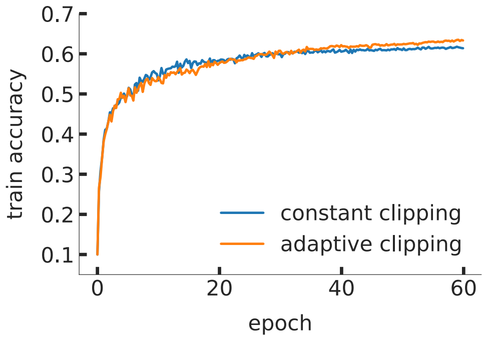


Figure 4: Test accuracy vs clipping methods

We replaced parameter C with α, β and σ_{l^2}. However, the model is very robust to changes in β and σ_{l^2}. Doubling the values of either β or σ_{l^2} yields very similar test accuracy. Because our approach is *adaptive*, we expect a single parameter value for α leads to good performance across tasks.

To examine this hypothesis, we carried out additional experiments on MNIST, CIFAR-10 and CIFAR-100 (reported in [9]). An α value of 1.0 is near the optimum for all datasets. This suggests that adaptive clipping results in parameters that are easier to choose without seeing the data. A broader investigation across datasets is required to test this hypothesis further.

2.3 Large batch training

Training with larger batches reduces privacy spending. To illustrate, Figure 5 shows the noise added per example as a function of batch size (using the the moments accountant from [1] to find the minimum σ that fully uses a pre-defined privacy budget in 10 epochs for a training set of 60,000 examples). Large batches dramatically reduce added noise. However, large batches can strongly hurt performance. In Goyal et al. [5], several methods are introduced to improve the performance of large-batch training. We adopt the simplest: scaling the learning rate along with the batch size.

We train small Alexnet models with varying batch sizes on CIFAR-10, using the DPSGD algorithm for 60 epochs with a budget of $\epsilon = 20$. A base learning rate of 0.01 is used for a batch size of 128. We increase both the the batch size and learning rate by a factor of k, and repeat the experiment. We train each model for 60 epochs, until an accumulated privacy loss of $\epsilon = 20 \pm 0.05$ after 60 epochs is found. Table 1 shows the results: training DP models with larger batches can be beneficial, but only when the learning rate is scaled accordingly.

Figure 5: σ/example vs batch size

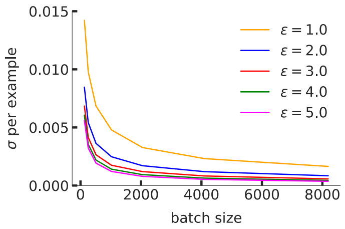


Table 1: Batch size versus accuracy

Batch size	accuracy
128	61.6%
512	64.2%
1024	**66.9%**
1024 (base lr)	47.2%

3 Conclusion

For differentially private learning, hyperparameter optimization on sensitive datasets is undesirable. The proposed methods enable an approach to differentially private learning with reduced privacy spending on hyperparameter tuning before training the final model. Given a classification task and data dimensions, we suggest the following approach to choosing the differential privacy parameters:

- Choose a model that is successfully tested on a non-private, similar benchmark tasks and use default hyperparameters.
- Choose the largest batch size that fits in memory on the training device(s) and scale the learning rate accordingly.
- Calibrate the noise scale parameter with the DPSGD [1] or DP-FedAvg [8] model on a centralized dataset of random noise until the sanity checks succeed.
- When all sanity checks have passed, train on private data with the same budget. Use a small portion of the budget for computing the differentially private mean ℓ_2-norm and use the adaptive clipping method of Section 2.2 with default values $\alpha = 1.0$, $\beta = 2.0$.

These steps combined, provide the structure of a basic differential privacy training workflow. This is far from a full-proof approach: the sanity checks function more as unit tests than hard guarantees, some architectures, like conditional models [2], are not yet supported, and it is not clear whether this approach is sufficient when adversaries actively attempt to influence the training process, such as in the privacy attacks proposed by Hitaj et al. [6]. We hope that our approach provides a basis that can be extended to study such questions.

Acknowledgements We thank the reviewers for their valuable comments. This publication was supported by the Amsterdam Academic Alliance Data Science (AAA-DS) Program Award to the UvA and VU Universities.

References

[1] M. Abadi, A. Chu, I. Goodfellow, H. Brendan McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep Learning with Differential Privacy. *ArXiv e-prints*, July 2016.

[2] Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. The secret sharer: Measuring unintended neural network memorization & extracting secrets. *CoRR*, abs/1802.08232, 2018.

[3] Kamalika Chaudhuri and Staal A Vinterbo. A stability-based validation procedure for differentially private machine learning. In *Advances in Neural Information Processing Systems*, pages 2652–2660, 2013.

[4] Cynthia Dwork. Differential privacy. In *33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*, volume 4052, pages 1–12, Venice, Italy, July 2006. Springer Verlag.

[5] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017.

[6] Briland Hitaj, Giuseppe Ateniese, and Fernando Pérez-Cruz. Deep models under the GAN: information leakage from collaborative deep learning. *CoRR*, abs/1702.07464, 2017.

[7] Jing Lei, Anne-Sophie Charest, Aleksandra Slavkovic, Adam Smith, and Stephen Fienberg. Differentially private model selection with penalized and constrained likelihood. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 181(3):609–633, 2018.

[8] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private language models without losing accuracy. *CoRR*, abs/1710.06963, 2017.

[9] Koen Lennart van der Veen. A Practical Approach to Differential Private Learning. Master's thesis, University of Amsterdam, Amsterdam, The Netherlands, 2018.

[10] Yu-Xiang Wang, Jing Lei, and Stephen E Fienberg. On-average kl-privacy and its equivalence to generalization for max-entropy mechanisms. In *International Conference on Privacy in Statistical Databases*, pages 121–134. Springer, 2016.

[11] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530, 2016.