Practical Differentially Private Hyperparameter Tuning with Subsampling

Antti Koskela and Tejas Kulkarni

Nokia Bell Labs

TrustML-(un)Limited ICLR23

Tuning the hyperparameters of differentially private (DP) machine learning (ML) algorithms often requires use of sensitive data and this may leak private information via hyperparameter values.

Compared to plain SGD, DP brings additional hyperparameters to tune: the noise level σ and the clipping constant C.

- We use the results by Papernot and Steinke (2022)[1]as a building block.
- Their work was based on the analysis of Liu and Talwar (2019)[2]
- Also, Mohapatra et al. (2022)[3]showed that a naive RDP accounting (i.e., release all models) often leads to lower DP bounds than the methods by Liu and Talwar (2019)

[1]Papernot, Nicolas, and Thomas Steinke. "Hyperparameter Tuning with Renyi Differential Privacy." International Conference on Learning Representations 2022.

[2]Liu, Jingcheng, and Kunal Talwar. "Private selection from private candidates." Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing. 2019.

[3]Mohapatra, Shubhankar, et al. "The role of adaptive optimizers for honest private hyperparameter selection." Proceedings of the AAAI conference on artificial intelligence. Vol. 36. No. 7. 2022.

However, still, running these algorithms increase the ε-values significantly, and can be computationally heavy as each candidate model requires training a new model.

Our novelty:

Consider using only a random subset of the sensitive data for the tuning part and use the output hyperparameter values and the corresponding model for training subsequent models.

This automatically leads to both lower DP privacy leakage and lower computational cost.

Definition of differential privacy:

Let $\varepsilon > 0$ and $\delta \in [0, 1]$. We say that a mechanism $\mathcal{M}$ is (ε, δ)-DP, if for all neighbouring datasets X and Y and for every measurable set $E \subset \mathcal{O}$ we have:

$$\Pr(\mathcal{M}(X) \in E) \leq \mathrm{e}^{\varepsilon} \Pr(\mathcal{M}(Y) \in E) + \delta.$$

We will also use the Rényi differential privacy (RDP):

We say that a mechanism $\mathcal{M}$ is (α, ε)-RDP, if for all neighbouring datasets X and Y, the output distributions $\mathcal{M}(X)$ and $\mathcal{M}(Y)$ have Rényi divergence of order α at most ε, i.e.,

$$\max_{X \sim Y} D_{\alpha}\big(\mathcal{M}(X) || \mathcal{M}(Y)\big) \leq \varepsilon.$$

Our method works as below:

1. Use Poisson subsampling to draw $X_1 \subset X$: draw a random subset X_1 such that each $x \in X$ is included in X_1 with probability q.
2. Compute $(\theta_1, t_1) = \mathcal{M}_1(X_1)$, where $\mathcal{M}_1$ is a hyperparameter tuning algorithm by Papernot and Steinke (2022) that outputs the vector of optimal hyperparameters t_1 and the corresponding model θ_1.
3. Extrapolate the hyperparameters t_1 to the dataset $X \setminus X_1$: $t_1 \to t_2$.
4. Compute $\theta_2 = \mathcal{M}_2(t_2, X \setminus X_1)$, where $\mathcal{M}_2$ is the base mechanism (e.g., DP-SGD).

Denote the whole mechanism by $\mathcal{M}$. Then, we may write

$$\mathcal{M}(X) = \left(\mathcal{M}_1(X_1), \mathcal{M}_2\left(\mathcal{M}_1(X_1), X \setminus X_1\right)\right).$$

Theorem. Let $X \in \mathcal{X}^n$ and $Y = X \cup \{x'\}$ for some $x' \in \mathcal{X}$. Let $\mathcal{M}(X)$ be the mechanism described above, i.e.

$$\mathcal{M}(X) = \big(\mathcal{M}_1(X_1), \mathcal{M}_2\big(\mathcal{M}_1(X_1), X \setminus X_1\big)$$

such that X_1 is sampled with sampling ratio q, $0 \leq q \leq 1$. Let $\alpha > 1$. Denote by $\varepsilon_1(\alpha)$ and $\varepsilon_2(\alpha)$ the RDP-values of mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively. We have that

$$D_\alpha\big(\mathcal{M}(Y)||\mathcal{M}(X)\big) \leq \frac{1}{\alpha - 1} \log \Bigg(q^\alpha \cdot \exp\big((\alpha - 1)\varepsilon_1(\alpha)\big) + (1-q)^\alpha \cdot \exp\big((\alpha - 1)\varepsilon_2(\alpha)\big) + \sum_{j=1}^{\alpha-1} \binom{\alpha}{j} \cdot q^{\alpha - j} \cdot (1-q)^j \cdot \exp\big((\alpha - j - 1)\varepsilon_1(\alpha - j)\big) \exp\big((j-1)\varepsilon_2(j)\big) \Bigg)$$

We obtain a similar expression for $D_\alpha\big(\mathcal{M}(X)||\mathcal{M}(Y)\big)$.

Notice: the weight q is attached to $\varepsilon_1(\alpha)$, $(1-q)$ attached to $\varepsilon_2(\alpha)$. We get $\varepsilon_1(\alpha)$ and $\varepsilon_2(\alpha)$ as $q \to 1$ and $q \to 0$, respectively.

If we carry out tuning using a subset of size m,

- We multiply the learning rate η by n/m when transferring to the dataset of size n.
- Clipping constant C, the noise level σ and the subsampling ratio γ are kept constant in this transfer.

With these rules, the noise variance that get injected to the model stays constant.

Sander et al.[4]proposed hyperparameter scaling laws using a similar guiding principle. This scaling of learning rate was also used by[5].

When using Adam with DP-SGD gradients, we found that keeping the value of learning rate fixed lead to better results.

[4]Sander, Tom, Pierre Stock, and Alexandre Sablayrolles. "TAN without a burn: Scaling Laws of DP-SGD." arXiv:2210.03403 (2022).

[5]Koen Lennart van der Veen, Ruben Seggers, Peter Bloem, and Giorgio Patrini. Three tools for practical differential privacy. NeurIPS 2018 Privacy Preserving Machine Learning workshop, arXiv:1812.02890, 2018

The expected number of required gradient evaluations for our approach is bounded by $(\mu \cdot q \cdot n + n) \cdot \mathrm{epochs}$, whereas the baseline requires in expectation $\mu \cdot n \cdot \mathrm{epochs}$ evaluations.

For example,

- with $\mu = 10$ and $q = 0.1$, the baseline requires $\frac{\mu}{\mu \cdot q + 1} \approx 5$ times more gradient evaluations than our method.
- with $\mu = 40$ and $q = 0.1$, the baseline requires $\frac{\mu}{\mu \cdot q + 1} \approx 8$ times more gradient evaluations than our method.

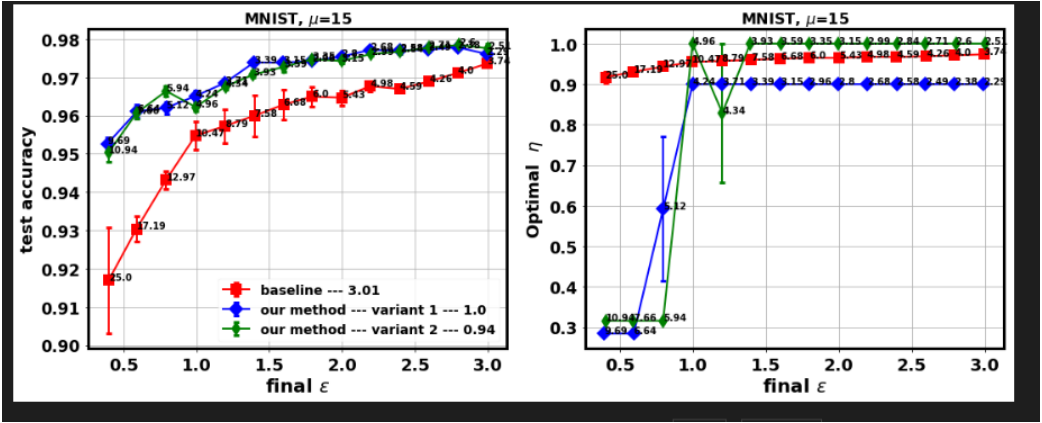

Figure: MNIST and tuning of the learning rate for a small neural network. Final accuracies for our method (variant 1 and variant 2) and the baseline method by Papernot and Steinke.

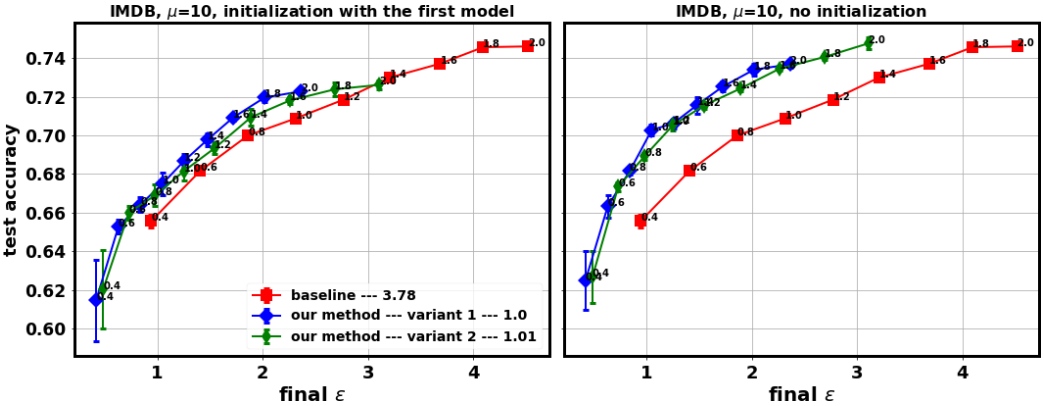

Figure: IMDB dataset and tuning of the learning rate for a small neural network. Final accuracies for our method (variant 1 and variant 2) and the baseline method by Papernot and Steinke.

Experiment: Fashion MNIST and tuning σ, γ and η.

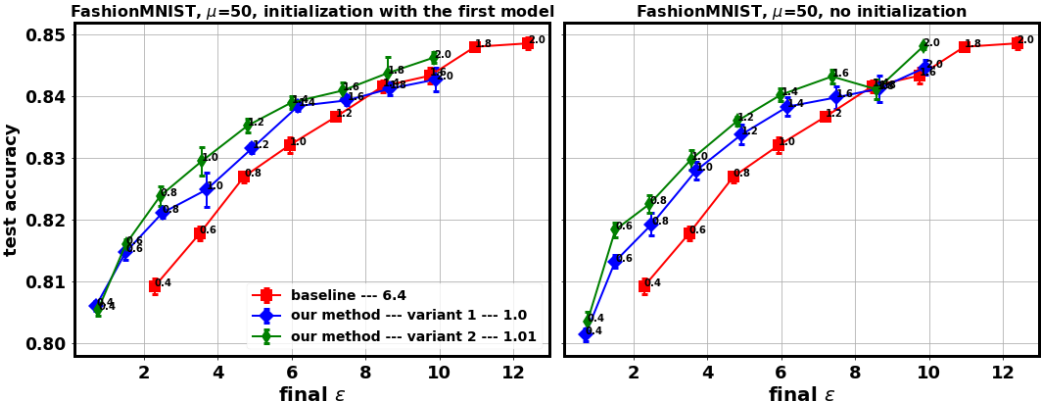

Figure: Fashion MNIST and tuning of σ, γ and η for a small neural network. Final accuracies for our method (variant 1 and variant 2) and the baseline method by Papernot and Steinke.

To conclude:

- We have proposed a novel approach that significantly decreases the compute and privacy cost of DP hyperparameter tuning.
- We have focussed on DP-SGD based optimizers, however we believe the method is applicable to other problems as well.
- We have also proposed ways to rigorously deal with hyperparameters that affect the privacy guarantees (not covered here, details given in the preprint).

Preprint:

Antti Koskela and Tejas Kulkarni. "Practical Differentially Private Hyperparameter Tuning with Subsampling." arXiv preprint arXiv:2301.11989 (2023).

antti.h.koskela@nokia-bell-labs.com